THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS” WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE. IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOSS PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: https://www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2018 Cisco Systems, Inc. All rights reserved.
CONTENTS

CHAPTER 1

IP Multicast Technology Overview  1

Information About IP Multicast Technology  1
Role of IP Multicast in Information Delivery  1
Multicast Group Transmission Scheme  1
IP Multicast Routing Protocols  3
IP Multicast Group Addressing  3
  IP Class D Addresses  4
IP Multicast Address Scoping  4
Layer 2 Multicast Addresses  5
IP Multicast Delivery Modes  6
  Any Source Multicast  6
  Source Specific Multicast  6
Protocol Independent Multicast  6
  PIM Dense Mode  7
  PIM Sparse Mode  7
  Sparse-Dense Mode  8
Multicast Group Modes  8
  Sparse Mode  9
  Dense Mode  9
Rendezvous Points  9
  Auto-RP  9
  Sparse-Dense Mode for Auto-RP  10
Multicast Forwarding  11
  Multicast Distribution Source Tree  11
  Multicast Distribution Shared Tree  12
  Source Tree Advantage  13
Restrictions for Source Specific Multicast 35
Information About Source Specific Multicast 36
SSM Overview 36
SSM Components 36
How SSM Differs from Internet Standard Multicast 37
SSM Operations 37
IGMPv3 Host Signaling 38
Benefits of Source Specific Multicast 38
IGMP v3lite Host Signalling 39
How to Configure Source Specific Multicast 40
  Configuring SSM 40
  Monitoring SSM 41
Configuration Examples of Source Specific Multicast 41
  SSM with IGMPv3 Example 41

CHAPTER 4

SSM Mapping 43
  Finding Feature Information 43
  Prerequisites for SSM Mapping 43
  Restrictions for SSM Mapping 44
  Information About SSM Mapping 44
     SSM Components 44
     Benefits of Source Specific Multicast 44
  SSM Transition Solutions 45
  SSM Mapping Overview 46
     Static SSM Mapping 46
     DNS-Based SSM Mapping 47
  SSM Mapping Benefits 48
How to Configure SSM Mapping 48
  Configuring Static SSM Mapping 48
  Configuring DNS-Based SSM Mapping 50
  Configuring Static Traffic Forwarding with SSM Mapping 51
  Verifying SSM Mapping Configuration and Operation 52
Configuration Examples for SSM Mapping 55
  SSM Mapping Example 55
### Chapter 7: Configuring Multicast VPN Extranet Support

**Finding Feature Information** 101

**Prerequisites for Configuring Multicast VPN Extranet Support** 101

**Restrictions for Configuring Multicast VPN Extranet Support** 102

**Information About Multicast VPN Extranet Support** 102

- **Overview of MVPN Extranet Support** 102
- **Benefits of MVPN Extranet Support** 103
- **Components of an Extranet MVPN** 103
- **Solution for MVPN Extranet Support** 104

**Configuration Guidelines for MVPN Extranet Support** 104

- **MVPN Extranet Support Configuration Guidelines for Option 1** 104
- **MVPN Extranet Support Configuration Guidelines for Option 2** 105
- **RPF for MVPN Extranet Support Using Imported Routes** 106
- **RPF for MVPN Extranet Support Using Static Mroutes** 106

**Multicast VPN Extranet VRF Select** 107

**How to Configure Multicast VPN Extranet Support** 107

**Configuring MVPN Support** 107

- **Configuring the Receiver MVRF on the Source PE - Option 1 (SSC)** 107
- **Configuring the Source MVRF on the Receiver PE - Option 2 (RSC)** 110

**Configuring RPF for MVPN Extranet Support Using Static Mroutes** 112

**Configuring Group-Based VRF Selection Policies with MVPN** 113

**Configuration Examples for Multicast VPN Extranet Support** 115

- **Example: Configuring the Receiver VRF on the Source PE Router - Option 1 (SSC)** 115
- **Example: Configuring the Source VRF on the Receiver PE - Option 2 (RSC)** 122
- **Example: Displaying Statistics for MVPN Extranet Support** 130
- **Example: Configuring RPF for MVPN Extranet Support Using Static Mroutes** 132
- **Example: Configuring Group-Based VRF Selection Policies with MVPN Extranet Support** 133

### Chapter 8: IPv6 Multicast PIM

**IPv6 Multicast PIM** 135
Prerequisites for IPv6 Multicast 135
Restrictions for IPv6 Multicast 135
Information About IPv6 Multicast 136
  IPv6 Multicast Routing Implementation 136
  Protocol Independent Multicast 137
  PIM-Sparse Mode 137
PIM IPv6 Stub Routing 141
MRIB 142
MFIB 142
  MFIB 143
IPv6 Multicast Process Switching and Fast Switching 143
Enabling IPv6 Multicast Routing 144
IPv6 Multicast: PIM Sparse Mode 144
IPv6 PIM Passive Mode 144
IPv6 Multicast: PIM Source-Specific Multicast 144
IPv6 Source Specific Multicast Mapping 144
How to Configure IPv6 Multicast 145
  Enabling IPv6 Multicast Routing 145
  Configuring PIM Options 145
  Configuring PIM-SM and Displaying PIM-SM Information for a Group Range 147
Reseting the PIM Traffic Counters 149
Clearing the PIM Topology Table to Reset the MRIB Connection 149
Turning Off IPv6 PIM on a Specified Interface 151
Disabling Embedded RP Support in IPv6 PIM 152
Configuring IPv6 SSM 153
Configuring PIM Passive/MLD State Limit 154
Configuring a BSR 156
  Configuring a BSR and Verifying BSR Information 156
  Sending PIM RP Advertisements to the BSR 157
Using MFIB in IPv6 Multicast 157
  Verifying MFIB Operation in IPv6 Multicast 157
  Resetting MFIB Traffic Counters 158
Configuration Examples for IPv6 Multicast 159
  Sample Output from the show ipv6 rpf Command 159
Example: Enabling IPv6 Multicast Routing 159
Example: Configuring PIM Options 159
Example: Configuring PIM 159
Example: Displaying PIM-SM Information for a Group Range 159
Example: Displaying IPv6 PIM Topology Information 160
Example: Displaying Information About PIM Traffic 161
Example: Disabling Embedded RP Support in IPv6 PIM 161
Example: IPv6 SSM Mapping 162

CHAPTER 9
IPv6 Multicast Listener Discovery Protocol 163
Finding Feature Information 163
Restrictions for IPv6 Multicast Listener Discovery Protocol 163
Information About IPv6 Multicast Listener Discovery Protocol 164
IPv6 Multicast Overview 164
IPv6 Multicast Routing Implementation 164
Multicast Listener Discovery Protocol for IPv6 165
MLD Access Group 166
How to Configure IPv6 Multicast Listener Discovery Protocol 166
Enabling IPv6 Multicast Routing 166
Customizing MLD on an Interface 167
Disabling MLD Device-Side Processing 170
Resetting the MLD Traffic Counters 171
Clearing the MLD Interface Counters 171
Clearing the MLD Groups 172
Verifying IPv6 Multicast Listener Discovery Protocol 173

CHAPTER 10
Using MSDP to Interconnect Multiple PIM-SM Domains 177
Finding Feature Information 177
178
Information About Using MSDP to Interconnect Multiple PIM-SM Domains 178
Benefits of Using MSDP to Interconnect Multiple PIM-SM Domains 178
178
MSDP Message Types 180
SA Messages 180
Controlling SA Messages Originated by an RP for Local Sources 199
Controlling the Forwarding of SA Messages to MSDP Peers Using Outgoing Filter Lists 200
Controlling the Receipt of SA Messages from MSDP Peers Using Incoming Filter Lists 201
Using TTL Thresholds to Limit the Multicast Data Sent in SA Messages 202
Requesting Source Information from MSDP Peers 203
Controlling the Response to Outgoing SA Request Messages from MSDP Peers Using SA Request Filters 204
Including a Bordering PIM Dense Mode Region in MSDP 205
Configuring an Originating Address Other Than the RP Address 206
Monitoring MSDP 207
Clearing MSDP Connections Statistics and SA Cache Entries 209
Enabling SNMP Monitoring of MSDP 210
Troubleshooting Tips 211
Configuration Examples for Using MSDP to Interconnect Multiple PIM-SM Domains 212
Example: Configuring an MSDP Peer 212
Example: Configuring MSDP MD5 Password Authentication 212
Example: Configuring a Default MSDP Peer 213
Example: Configuring MSDP Mesh Groups 214
Additional References 214
Feature Information for Multicast Source Discovery Protocol 215

CHAPTER 11
MPLS Point-to-Multipoint Traffic Engineering 217
Information About MPLS Point-to-Multipoint Traffic Engineering 217
MPLS Point-to-Multipoint Traffic Engineering Overview 217
How P2MP TE Sub-LSPs Are Signaled 219
How P2MP TE Traffic Is Forwarded 220
Computing the IGP Path Using Dynamic Paths or Explicit Paths 221
Remerge Events 222
Crossover Events 222
Benefits of MPLS Point-to-Multipoint Traffic Engineering 222
MPLS Point-to-Multipoint Traffic Engineering—Re-optimizing Traffic 222
P2P TE Tunnels Coexist with P2MP TE Tunnels 223
Using FRR to Protect P2MP TE Links 224
FRR Failure Detection Mechanisms 226
CHAPTER 14

MVPN mLDP Partitioned MDT 287
Finding Feature Information 287
Prerequisites for MVPN mLDP Partitioned MDT 287
Restrictions for MVPN mLDP Partitioned MDT 287
Information About MVPN mLDP Partitioned MDT 288
Overview of MVPN mLDP Partitioned MDT 288
How to Configure MVPN mLDP Partitioned MDT 289
Configuring MVPN mLDP Partitioned MDT 289
Configuration Examples for MVPN mLDP Partitioned MDT 291
Example: VPN mLDP Partitioned MDT  291

CHAPTER 15  MLDP In-Band Signaling/Transit Mode  295
  Finding Feature Information  295
  Restrictions for MLDP In-Band Signaling  295
  Information About MLDP In-Band Signaling/Transit Mode  295
    MLDP In-Band Signaling/Transit Mode  295
  How to Configure MLDP In-Band Signaling/Transit Mode  296
    Enabling In-Band Signaling on a PE Device  296
  Configuration Examples for MLDP In-Band Signaling/Transit Mode  297
    Example: In-Band Signaling  297
IP Multicast Technology Overview

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

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

- Information About IP Multicast Technology, on page 1

Information About IP Multicast Technology

Role of IP Multicast in Information Delivery

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

Multicast Group Transmission Scheme

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

Multicast is based on this group concept. A multicast group is an arbitrary number of receivers that join a group in order to receive a particular data stream. This multicast group has no physical or geographical boundaries—the hosts can be located anywhere on the Internet or on any private internetwork. Hosts that are interested in receiving data from a source to a particular group must join that group. Joining a group is accomplished by a host receiver by way of the Internet Group Management Protocol (IGMP).
In a multicast environment, any host, regardless of whether it is a member of a group, can send to a group. However, only the members of a group can receive packets sent to that group. Multicast packets are delivered to a group using best-effort reliability, just like IP unicast packets.

**Unicast transmission**—One host sends and the other receives.

**Broadcast transmission**—One sender to all receivers.

**Multicast transmission**—One sender to a group of receivers.

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

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

- IGMP is used between hosts on a LAN and the routers on that LAN to track the multicast groups of which hosts are members.

- Protocol Independent Multicast (PIM) is used between routers so that they can track which multicast packets to forward to each other and to their directly connected LANs.

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

Figure 1: IP Multicast Routing Protocols

IP Multicast Group Addressing

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

IP Class D Addresses

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

Note

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

IP Multicast Address Scoping

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

Table 1: Multicast Address Range Assignments

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

Reserved Link-Local Addresses

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

Within this range, reserved link-local addresses provide network protocol functions for which they are reserved. Network protocols use these addresses for automatic router discovery and to communicate important routing information. For example, Open Shortest Path First (OSPF) uses the IP addresses 224.0.0.5 and 224.0.0.6 to exchange link-state information.
IANA assigns single multicast address requests for network protocols or network applications out of the 224.0.1.xxx address range. Multicast routers forward these multicast addresses.

---

**Note**

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

---

**Globally Scoped Addresses**

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

**Source Specific Multicast Addresses**

Addresses in the range 232.0.0.0/8 are reserved for Source Specific Multicast (SSM) by IANA. In Cisco IOS software, you can use the `ip pim ssm` command to configure SSM for arbitrary IP multicast addresses also. SSM is an extension of Protocol Independent Multicast (PIM) that allows for an efficient data delivery mechanism in one-to-many communications. SSM is described in the IP Multicast Delivery Modes, on page 6 section.

**GLOP Addresses**

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

**Limited Scope Addresses**

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

---

**Note**

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

---

**Layer 2 Multicast Addresses**

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

**IP Multicast Delivery Modes**

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

**Any Source Multicast**

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

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

**Source Specific Multicast**

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

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

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

**Protocol Independent Multicast**

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

Although PIM is called a multicast routing protocol, it actually uses the unicast routing table to perform the reverse path forwarding (RPF) check function instead of building up a completely independent multicast routing table. Unlike other routing protocols, PIM does not send and receive routing updates between routers.
PIM can operate in dense mode or sparse mode. The router can also handle both sparse groups and dense groups at the same time. The mode determines how the router populates its multicast routing table and how the router forwards multicast packets it receives from its directly connected LANs.

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

**PIM Dense Mode**

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

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

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

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

---

**Note**

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

---

**PIM Sparse Mode**

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

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

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

In sparse mode, a router assumes that other routers do not want to forward multicast packets for a group, unless there is an explicit request for the traffic. When hosts join a multicast group, the directly connected routers send PIM Join messages toward the RP. The RP keeps track of multicast groups. Hosts that send multicast packets are registered with the RP by the first hop router of that host. At this point, packets are forwarded on a shared distribution tree. If the multicast traffic from a specific source is sufficient, the first hop router of the host may send Join messages toward the source to build a source-based distribution tree.
Sources register with the RP and then data is forwarded down the shared tree to the receivers. The edge router sends PIM (S,G) Join messages toward that source. Each router along the reverse path compares the unicast routing metric of the RP address to the metric of the source address. If the metric for the source address is better, it will forward a PIM (S,G) Join message toward the source. If the metric for the RP is the same or better, then the PIM (S,G) Join message will be sent in the same direction as the RP. In this case, the shared tree and the source tree would be considered congruent.

If the shared tree is not an optimal path between the source and the receiver, the routers dynamically create a source tree and stop traffic from flowing down the shared tree. This behavior is the default behavior in software. Network administrators can force traffic to stay on the shared tree by using the `ip pim spt-threshold infinity` command. The default value of `ip pim spt-threshold infinity` command is 0.

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

**Sparse-Dense Mode**

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

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

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

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

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

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

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

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

**Multicast Group Modes**

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

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

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

**Sparse Mode**

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

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

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

**Dense Mode**

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

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

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

**Rendezvous Points**

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

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

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

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

**Auto-RP**

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

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

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

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

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

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

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

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

**Sparse-Dense Mode for Auto-RP**

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

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

Multicast Forwarding

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

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

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

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

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

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

Multicast Distribution Source Tree

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

The figure shows an example of an SPT for group 224.1.1.1 rooted at the source, Host A, and connecting two receivers, Hosts B and C.
Using standard notation, the SPT for the example shown in the figure would be (192.168.1.1, 224.1.1.1).

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

**Multicast Distribution Shared Tree**

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

The following figure shows a shared tree for the group 224.2.2.2 with the root located at Router D. This shared tree is unidirectional. Source traffic is sent towards the RP on a source tree. The traffic is then forwarded down the shared tree from the RP to reach all of the receivers (unless the receiver is located between the source and the RP, in which case it will be serviced directly).
In this example, multicast traffic from the sources, Hosts A and D, travels to the root (Router D) and then down the shared tree to the two receivers, Hosts B and C. Because all sources in the multicast group use a common shared tree, a wildcard notation written as (*, G), pronounced "star comma G", represents the tree. In this case, * means all sources, and G represents the multicast group. Therefore, the shared tree shown in the figure would be written as (*, 224.2.2.2).

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

### Source Tree Advantage

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

### Shared Tree Advantage

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

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

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

**Reverse Path Forwarding**

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

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

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

**RPF Check**

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

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

1. The router looks up the source address in the unicast routing table to determine if the packet has arrived on the interface that is on the reverse path back to the source.

2. If the packet has arrived on the interface leading back to the source, the RPF check succeeds and the packet is forwarded out the interfaces present in the outgoing interface list of a multicast routing table entry.
3. If the RPF check in Step 2 fails, the packet is dropped.

The figure shows an example of an unsuccessful RPF check.

*Figure 3: RPF Check Fails*

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

The figure shows an example of a successful RPF check.

*Figure 4: RPF Check Succeeds*

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

**Guidelines for Choosing a PIM Mode**

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

Basic guidelines include the following:

- In general, if the application is one-to-many or many-to-many in nature, then PIM-SM can be used successfully.
- For optimal one-to-many application performance, SSM is appropriate but requires IGMP version 3 support.
CHAPTER 2

Configuring Basic IP Multicast

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

Note

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

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.
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 5: 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.

**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)
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 v3 lite 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.
- 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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td><code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><code>Example:</code></td>
<td></td>
</tr>
<tr>
<td><code>Device&gt; enable</code></td>
<td></td>
</tr>
</tbody>
</table>

- Enter your password if prompted.
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Device# configure terminal</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Enables IP multicast routing.</td>
</tr>
<tr>
<td><code>ip multicast-routing distributed</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config)# ip multicast-routing</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>--</td>
</tr>
<tr>
<td>Either perform Steps 5 through 7 or perform Steps 6 and 8.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>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.</td>
</tr>
<tr>
<td><code>ip pim autorp listener</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config)# ip pim autorp listener</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>Selects an interface that is connected to hosts on which PIM can be enabled.</td>
</tr>
<tr>
<td><code>interface type number</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config)# interface Gigabitethernet 1/0/0</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>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.</td>
</tr>
<tr>
<td><code>ip pim sparse-mode</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-if)# ip pim sparse-mode</td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>Enables PIM sparse-dense mode on an interface.</td>
</tr>
<tr>
<td><code>ip pim sparse-dense-mode</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-if)# ip pim sparse-dense-mode</td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td>Exits interface configuration mode and returns to global configuration mode.</td>
</tr>
<tr>
<td><code>exit</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-if)# exit</td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td>Repeat Steps 1 through 9 on all PIM interfaces.</td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td>Sends RP announcements out all PIM-enabled interfaces.</td>
</tr>
<tr>
<td>`ip pim send-rp-announce {interface-type interface-number</td>
<td>ip-address} scope ttl-value [group-list access-list] [interval seconds] [bidir]`</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| Device(config)# ip pim send-rp-announce loopback0 scope 31 group-list 5 | • 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. |

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| Step 12 ip pim send-rp-discovery [interface-type interface-number] scope ttl-value [interval seconds] | Configures the device to be an RP mapping agent.  
• Perform this step on RP mapping agent devices or on combined RP/RP mapping agent devices.  

Note  
Auto-RP allows the RP function to run separately on one device and the RP mapping agent to run on one or multiple devices. It is possible to deploy the RP and the RP mapping agent on a combined RP/RP mapping agent device.  

• Use the optional `interface-type` and `interface-number` arguments to define which IP address is to be used as the source address of the RP mapping agent.  

• Use the `scope` keyword and `ttl-value` argument to specify the Time-to-Live (TTL) value in the IP header of Auto-RP discovery messages.  

• Use the optional `interval` keyword and `seconds` argument to specify the interval at which Auto-RP discovery messages are sent. |
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Purpose</strong></td>
<td>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).</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>The example shows limiting the Auto-RP discovery messages to 31 hops on loopback interface 1.</td>
</tr>
<tr>
<td><strong>Step 13</strong></td>
<td>Filters incoming RP announcement messages sent from candidate RPs (C-RPs) to the RP mapping agent.</td>
</tr>
<tr>
<td><strong>ip pim rp-announce-filter rp-list access-list group-list</strong></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td>Device(config)# ip pim rp-announce-filter rp-list 1 group-list 2</td>
<td></td>
</tr>
<tr>
<td><strong>Step 14</strong></td>
<td>(Optional) Prevents PIM dense mode fallback.</td>
</tr>
<tr>
<td><strong>no ip pim dm-fallback</strong></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td>Device(config)# no ip pim dm-fallback</td>
<td></td>
</tr>
<tr>
<td><strong>Step 15</strong></td>
<td>Selects an interface that is connected to hosts on which PIM can be enabled.</td>
</tr>
<tr>
<td><strong>interface type number</strong></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td>Device(config)# interface gigabitethernet 1/0/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 16</strong></td>
<td>Configures an administratively scoped boundary.</td>
</tr>
<tr>
<td><strong>ip multicast boundary access-list [filter-autorp]</strong></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td>Device(config-if)# ip multicast boundary 10 filter-autorp</td>
<td></td>
</tr>
<tr>
<td><strong>Step 17</strong></td>
<td>Returns to global configuration mode.</td>
</tr>
<tr>
<td><strong>end</strong></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td>Device(config-if)# end</td>
<td></td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
</tr>
<tr>
<td>------</td>
<td>------------------</td>
</tr>
<tr>
<td>18</td>
<td><code>show ip pim autorp</code>&lt;br&gt;<code>Example:</code>&lt;br&gt;Device# show ip pim autorp</td>
</tr>
<tr>
<td>19</td>
<td><code>show ip pim rp [mapping] [rp-address]</code>&lt;br&gt;<code>Example:</code>&lt;br&gt;Device# show ip pim rp mapping</td>
</tr>
<tr>
<td>20</td>
<td>`show ip igmp groups [group-name</td>
</tr>
<tr>
<td>21</td>
<td>`show ip mroute [group-address</td>
</tr>
</tbody>
</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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>- Enter your password if prompted.</td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td>Step 2 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 3 ip multicast-routing distributed</td>
<td>Enables IP multicast routing.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# ip multicast-routing</td>
<td></td>
</tr>
<tr>
<td>Step 4 interface type number</td>
<td>Selects an interface that is connected to hosts on which PIM can be enabled.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# interface gigabitethernet 1/0/0</td>
<td></td>
</tr>
<tr>
<td>Step 5 ip pim sparse-mode</td>
<td>Enables PIM on an interface. You must use sparse mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-if)# ip pim sparse-mode</td>
<td></td>
</tr>
<tr>
<td>Step 6 Repeat Steps 1 through 5 on every interface that uses IP multicast.</td>
<td>--</td>
</tr>
<tr>
<td>Step 7 exit</td>
<td>Returns to global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Router(config-if)# exit</code></td>
<td>Ends the current configuration session and returns to EXEC mode.</td>
</tr>
</tbody>
</table>
| **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.  
- 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.  
- If no access list is defined, the RP will map to all multicast groups, 224/4.  
  
  **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` | (Optional) Displays RPs known in the network and shows how the router learned about each RP. |
| **Step 10** `show ip pim rp [mapping] [rp-address]` **Example:** `Router# show ip pim rp mapping` | (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 11** `show ip igmp groups [group-name | group-address| interface-type interface-number] [detail]` **Example:** `Router# show ip igmp groups` | (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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td></td>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td><code>ip multicast-routing distributed</code></td>
<td>Enables IP multicast routing.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config)# ip multicast-routing</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>`ip pim ssm {default</td>
<td>range access-list}`</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
<td></td>
</tr>
</tbody>
</table>
| Device(config)# ip pim ssm default | • 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:**  
Device# show ip igmp groups

(Optional) Displays the multicast groups having receivers that are directly connected to the device and that were learned through IGMP.  
• A receiver must be active on the network at the time that this command is issued in order for receiver information to be present on the resulting display.

**Step 12**  
**show ip mroute**  
**Example:**  
Device# show ip mroute

(Optional) Displays the contents of the IP mroutetable.  
• 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
```

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 gigabitethernet 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
```
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.

```
description ethernet connected to hosts
ip igmp version 3
!
ip pim ssm default

SSM Filtering Example
```

```
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
```
Configuring Source Specific Multicast

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

- Restrictions for Source Specific Multicast, on page 35
- Information About Source Specific Multicast, on page 36
- How to Configure Source Specific Multicast, on page 40
- Configuration Examples of Source Specific Multicast, on page 41

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. Therefore, enabling SSM in a network may cause problems for existing applications if they use addresses within the designated SSM range.

IGMP v3lite Requires a Cisco Last Hop Router

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

Note

This limitation does not apply to an application using the 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. 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 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 IGMP v3lite is used with hosts where the operating system is not upgraded for IGMPv3, because IGMP v3lite relies only on IGMPv1 or IGMPv2 membership reports.

**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.

---

**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 | rangeaccess-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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Defines the SSM range of IP multicast addresses.</td>
</tr>
<tr>
<td>`ip pim ssm [default</td>
<td>rangeaccess-list ]`</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Selects an interface that is connected to hosts on which IGMPv3, IGMP v3lite, and URD can be enabled.</td>
</tr>
<tr>
<td><code>interface type number</code></td>
<td>Emulates the interface number. Example:</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Enables PIM on an interface. You must use either sparse mode or sparse-dense mode.</td>
</tr>
<tr>
<td>`ip pim {sparse-mode</td>
<td>sparse-dense-mode}`</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Enables IGMPv3 on this interface. The default version of IGMP is set to Version 2.</td>
</tr>
<tr>
<td>Do one of the following:</td>
<td>Enables IGMPv3 on this interface. The default version of IGMP is set to Version 2. or</td>
</tr>
<tr>
<td>- <code>ip igmp version 3</code></td>
<td>Enables IGMPv3 on this interface. The default version of IGMP is set to Version 2. or</td>
</tr>
<tr>
<td>- <code>ip igmp v3lite</code></td>
<td>Enables IGMPv3 on this interface. The default version of IGMP is set to Version 2. or</td>
</tr>
</tbody>
</table>
### Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-if)# ip igmp version 3 or Router(config-if)# ip igmp v3lite</td>
<td>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.</td>
</tr>
</tbody>
</table>

### Monitoring SSM

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router# <code>show ip igmp groups detail</code></td>
<td>Displays the (S, G) channel subscription through IGMPv3 or IGMP v3lite.</td>
</tr>
<tr>
<td>Router# <code>show ip mroute</code></td>
<td>Displays whether a multicast group supports SSM service or whether a source-specific host report was received.</td>
</tr>
</tbody>
</table>

### Configuration Examples of Source Specific Multicast

**tbd**

### SSM with IGMPv3 Example

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

```plaintext
ip multicast-routing
!
interface GigabitEthernet3/1/0
 ip address 172.21.200.203 255.255.255.0
description backbone interface
 ip pim sparse-mode
!
interface GigabitEthernet3/2/0
 ip address 131.108.1.2 255.255.255.0
 ip pim sparse-mode
description ethernet connected to hosts
 ip igmp version 3
!
 ip pim ssm default
```
SSM with IGMPv3 Example

Configuring Source Specific Multicast
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, on page 43
• Prerequisites for SSM Mapping, on page 43
• Restrictions for SSM Mapping, on page 44
• Information About SSM Mapping, on page 44
• How to Configure SSM Mapping, on page 48
• Configuration Examples for SSM Mapping, on page 55

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, IGMPv3lite, 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 6: 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:

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IN A source-address-2</td>
</tr>
<tr>
<td></td>
<td>IN A source-address-n</td>
</tr>
</tbody>
</table>

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 RR.

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.

- 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`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables privileged EXEC mode. Enter your password if prompted.</td>
</tr>
<tr>
<td>enable</td>
<td>Device&gt; enable</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>configure terminal</td>
<td>Device# configure terminal</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Enables SSM mapping for groups in the configured SSM range. By default, this command enables DNS-based SSM mapping.</td>
</tr>
<tr>
<td>ip igmp ssm-map enable</td>
<td>Device(config)# ip igmp ssm-map enable</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>(Optional) Disables DNS-based SSM mapping. Disable DNS-based SSM mapping if you only want to rely on static SSM mapping. By default, the <code>ip igmp ssm-map</code> command enables DNS-based SSM mapping.</td>
</tr>
<tr>
<td>no ip igmp ssm-map query dns</td>
<td>Device(config)# no ip igmp ssm-map query dns</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Configures static SSM mapping. The ACL supplied for the <code>access-list</code> argument defines the groups to be mapped to the source IP address entered for the <code>source-address</code> argument.</td>
</tr>
<tr>
<td>ip igmp ssm-map static access-list source-address</td>
<td>Device(config)# ip igmp ssm-map static 11 172.16.8.11</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>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 switchcontrollerdevice determines the source addresses associated with the group by walking each configured <code>ip igmp ssm-map static</code> command. The switchcontrollerdevice associates up to 20 sources per group.</td>
</tr>
</tbody>
</table>
### Configuring DNS-Based SSM Mapping

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.
- 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. `ip name-server server-address1 [server-address2...server-address6]`
7. Repeat the Steps to configure additional DNS servers for redundancy, if required.
8. `end`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>enable</strong></td>
</tr>
<tr>
<td>Example:</td>
<td>Enter your password if prompted.</td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>configure terminal</strong></td>
</tr>
<tr>
<td>Example:</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>ip igmp ssm-map enable</strong></td>
</tr>
<tr>
<td>Example:</td>
<td>Enables SSM mapping for groups in a configured SSM range.</td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 4</td>
<td><code>ip igmp ssm-map query dns</code></td>
<td>(Optional) Enables DNS-based SSM mapping.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Device(config)# ip igmp ssm-map query dns</code></td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td><code>ip domain multicast domain-prefix</code></td>
<td>(Optional) Changes the domain prefix used for DNS-based SSM mapping.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Device(config)# ip domain multicast ssm-map.cisco.com</code></td>
<td></td>
</tr>
<tr>
<td>Step 6</td>
<td><code>ip name-server server-address1 [server-address2...server-address6]</code></td>
<td>Specifies the address of one or more name servers to use for name and address resolution.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Device(config)# ip name-server 10.48.81.21</code></td>
<td></td>
</tr>
<tr>
<td>Step 7</td>
<td>Repeat the Steps to configure additional DNS servers for redundancy, if required.</td>
<td>--</td>
</tr>
<tr>
<td>Step 8</td>
<td><code>end</code></td>
<td>Returns to privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Device(config-if)# end</code></td>
<td></td>
</tr>
</tbody>
</table>

### 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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> interface type number</td>
<td>Selects an interface on which to statically forward traffic for a multicast group using SSM mapping and enters interface configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config)# interface gigabitethernet 1/0/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> ip igmp static-group group-address source ssm-map</td>
<td>Configures SSM mapping to be used to statically forward a (S, G) channel out of the interface.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config-if)# ip igmp static-group 232.1.2.1 source ssm-map</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> end</td>
<td>Returns to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config-if)# end</td>
<td></td>
</tr>
</tbody>
</table>

### 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
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
172.16.8.4
```

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
```
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.

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

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>no ip domain lookup</strong></td>
<td>Disables IP DNS-based hostname-to-address translation. <strong>Note</strong> The <code>no ip domain-list</code> 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.</td>
</tr>
<tr>
<td><strong>ip domain multicast</strong></td>
<td>Specifies ssm-map.cisco.com as the domain prefix for SSM mapping.</td>
</tr>
<tr>
<td><strong>ssm-map.cisco.com</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ip name-server 10.48.81.21</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>ip multicast-routing</strong></td>
<td>Enables IP multicast routing.</td>
</tr>
<tr>
<td><strong>ip igmp ssm-map enable</strong></td>
<td>Enables SSM mapping.</td>
</tr>
</tbody>
</table>
| **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. |
**Command** | **Description**  
--- | ---  
ip igmp explicit-tracking | Minimizes the leave latency for IGMPv3 host leaving a multicast channel.  
*Note* | This command is not required for configuring SSM mapping.  
ip igmp limit 2 | Limits the number of IGMP states resulting from IGMP membership states on a per-interface basis.  
*Note* | This command is not required for configuring SSM mapping.  
ip igmp v3lite | Enables the acceptance and processing of IGMP v3lite membership reports on this interface.  
*Note* | This command is shown in the configuration only to demonstrate that IGMP v3lite can be configured simultaneously with SSM mapping; however, it is not required.  
ip urd | Enables interception of TCP packets sent to the reserved URD port 465 on an interface and processing of URD channel subscription reports.  
*Note* | This command is shown in the configuration only to demonstrate that URD can be configured simultaneously with SSM mapping; however, it is not required.  
ip pim ssm default | Configures SSM service.  
The *default* keyword defines the SSM range access list as 232/8.  
access-list 10 permit 232.1.2.10 0.0.0.255  
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.
CHAPTER 5

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, on page 59
- Prerequisites for Configuring Multicast Admission Control, on page 59
- Information About Configuring Multicast Admission Control, on page 59
- How to Configure Multicast Admission Control, on page 66
- Configuration Examples for Configuring Multicast Admission Control, on page 74

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

Per VRF mroutestate 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 mroutestate 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 mroutestate limiters are configured using the `ip multicastroute-limit` command in global configuration mode. The syntax of the `ip multicastroute-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 mroutestate limiter. The optional `vrf` keyword and `vrf-name` arguments are used with the `ip multicast limit` command to configure per MVRF mroutestate limiters.

**Note**

When configuring global and per VRF mroutestate 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 mroutestate limiters, the optional `threshold` argument is available to set mroutethreshold limits.

Mechanics of Global and Per MVRF Mroute State Limiters

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

- Each time the state for an mroutestate is created on a router, the Cisco IOS software checks to see if the limit for the global mroutestate limiter (if the mroutestate is associated with the global table) or the limit for the per MVRF mroutestate limiter (if the mroutestate is associated with the MVRF table) has been reached.

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

IGMP State Limit

The IGMP State Limit feature allows for the configuration of IGMP state limiters, which impose limits on
mroutestatesresultingfromIGMPmembershipreports(IGMPjoins)onaglobalorperinterfacebasis.
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 amulticastCACmechanisminnetworkenvironmentswherelll
themulticastflowsroughlyutilizesamemeanumbervfbandwidth.

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>
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.

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 ACL which will prevent the ACL from being accounted. If an mroute matches a deny statement, the search immediately continues to the next configured mroute state limiter. Configuring an explicit deny statement in an ACL can be more efficient than forcing the mroute to fall through an ACL by using an implicit deny-any statement at the end of the ACL.
How to Configure 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 a 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

IP Multicast: Multicast Configuration Guide, Cisco IOS XE Release 3S (Cisco ASR 900 Series)
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
  **Example:**  
  Router> enable |
| **Step 2** configure terminal | Enters global configuration mode.  
  **Example:**  
  Router# configure terminal |
| **Step 3** ip multicast route-limit `limit [threshold]` | Limits the number of mroutes that can be added to the global table.  
  **Example:**  
  Router(config)# ip multicast route-limit 1500 1460  
  • 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 | Ends the current configuration session and returns to privileged EXEC mode.  
  **Example:**  
  Router(config)# end |
| **Step 5** show ip mroutecount | (Optional) Displays mroute data and packet count statistics.  
  **Example:**  
  Router# show ip mroutecount  
  • 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 67 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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router&gt; enable</code></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>ip multicast vrf vrf-name route-limit limit [threshold]</code></td>
<td>Limits the number of mroutes that can be added to a particular MVRF table.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>• For the <code>vrf</code> keyword and <code>vrf-name</code> argument, specify the MVRF for which to apply the limit.</td>
</tr>
<tr>
<td><code>Router(config)# ip multicast vrf red route-limit 1500 1460</code></td>
<td>• For the required <code>limit</code> 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.</td>
</tr>
<tr>
<td></td>
<td>• Use the optional <code>threshold</code> argument to set an mroute threshold limit. The range is from 1 to 2147483647</td>
</tr>
<tr>
<td></td>
<td>• Maximum number of mroute state limits supported on MVRF level is 1000.</td>
</tr>
<tr>
<td><strong>Step 4</strong> Repeat Step 3 to configure additional per VRF mroute state limiters for other VRFs on an MVPN PE router.</td>
<td>--</td>
</tr>
<tr>
<td><strong>Step 5</strong> <code>end</code></td>
<td>Ends the current configuration session and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config)# end</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> <code>show ip mroute vrf vrf-name count</code></td>
<td>(Optional) Displays mroute data and packet count statistics related to the specified MVRF.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>• Use this command to verify the number of mroutes in a particular MVRF table.</td>
</tr>
</tbody>
</table>
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**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| Step 1 | enable | Enables privileged EXEC mode.  
*Example:* Device> enable |
| Step 2 | configure terminal | Enters global configuration mode.  
*Example:* Device# configure terminal |
| Step 3 | ip igmp limit  *number* | Configures a global limit on the number of mroute states resulting from IGMP membership reports (IGMP joins).  
*Example:* Device(config)# ip igmp limit 150 |
### Purpose

**Command or Action**

<table>
<thead>
<tr>
<th>Step 4</th>
<th>end</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Purpose</strong></td>
<td>Ends the current configuration session and returns to privileged EXEC mode.</td>
</tr>
</tbody>
</table>

**Example:**

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

<table>
<thead>
<tr>
<th>Step 5</th>
<th>show ip igmp groups</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Purpose</strong></td>
<td>(Optional) Displays the multicast groups with receivers that are directly connected to the device and that were learned through IGMP.</td>
</tr>
</tbody>
</table>

**Example:**

```plaintext
Device# show ip igmp groups
```

### What to Do Next

Proceed to the Configuring Per Interface IGMP State Limiters, on page 70 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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>enable</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
</tbody>
</table>

| **Step 2** | configure terminal  |
| **Example:** | Enters global configuration mode. |
| Device# configure terminal |

| **Step 3** | interface type number |
| **Example:** | Enters interface configuration mode. |
| Device type number |
### Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>• Specify an interface that is connected to hosts.</td>
</tr>
<tr>
<td><code>Device(config)# interface GigabitEthernet0/0</code></td>
<td></td>
</tr>
</tbody>
</table>

**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 mroute states created as a result of IGMP membership reports (IGMP joins).

**Step 5**

Do one of the following:  
• *exit*  
• *end*  
**Example:**  
`Device(config-if)# exit`  
`Device(config-if)# end`

• (Optional) Ends the current configuration session and returns to global configuration mode. Repeat steps 3 and 4 to configure a per interface limiter on another interface.  
• Ends the current configuration session and returns to privileged EXEC mode.

**Step 6**

**show ip igmp interface** [*type number*]  
**Example:**  
`Device# show ip igmp interface`

(Optional) Displays information about the status and configuration of IGMP and multicast routing on interfaces.

**Step 7**

**show ip igmp groups**  
**Example:**  
`Device# show ip igmp groups`

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

### Configuring Per Interface Mroute State Limiters

Perform this task to prevent DoS attacks or to provide a multicast CAC mechanism for controlling 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 4 to configure per interface mroute state limiters on additional interfaces.
### Detailed Steps

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>interface type number</td>
<td>Enters interface configuration mode for the specified interface type and number.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config)# interface GigabitEthernet0/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>ip multicast limit [connected</td>
<td>configures per interface mroute state limiters.</td>
</tr>
<tr>
<td></td>
<td>out</td>
<td></td>
</tr>
<tr>
<td></td>
<td>rpf] access-list max-entries</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-if)# ip multicast limit 15 100</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Repeat Step 4 to configure additional per interface mroute state limiters on this interface.</td>
<td>--</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>Repeat Steps 3 and Step 4 to configure per interface mroute state limiters on additional interfaces.</td>
<td>--</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>end</td>
<td>Returns to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-if)# end</td>
<td></td>
</tr>
</tbody>
</table>

### 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.

### 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.
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.

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.

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:

```plaintext
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 8: 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 mroutestate limit of 75 for the SD channels that match acl-basic.
- An mroutestate limit of 25 for the SD channels that match acl-premium.
- An mroutestate 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:

```plaintext
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
```
CHAPTER 6

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, on page 79
- Restrictions for Configuring Multicast VPN, on page 79
- Information About Configuring Multicast VPN, on page 81
- How to Configure Multicast VPN, on page 86
- Configuration Examples for Multicast VPN, on page 96

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.

• Generic Routing Encapsulation (GRE) based Multicast VPN (MVPN) is supported on RSP2 for IPv4 from Cisco IOS XE Release 3.17.0S onwards. MVPN GRE is supported only in video template.

Effective Cisco IOS XE Everest 16.6.1, the following restrictions are applicable on the Cisco ASR 900 RSP3 Module:

• MVPN bidirectional PIM is not supported.

• The following GREs are not supported:
  • Routing field
  • Sequence number
  • Secret source number
  • Recursion Control Field
  • Checksum

• MVPN with core interface pseudowire is not supported.

• A maximum of 20 OIFs towards core are supported.

• IPv6 is not supported.

• Route leaking is not supported.

• Only one EFP per BDI per physical port is supported.

• MVPN GREs and mLDPs are not supported on the same VRF.

• NETCONF/YANG is not supported.

The following table shows the scaling numbers for MVPN-GRE:

<table>
<thead>
<tr>
<th>Scale Scenario</th>
<th>RSP1A</th>
<th>RSP1B</th>
<th>RSP2</th>
<th>RSP3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of supported data MDT per VRF</td>
<td>255</td>
<td>255</td>
<td>255</td>
<td>255</td>
</tr>
<tr>
<td>Number of supported data MDT overall</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>Overall 4000. No restriction for data MDT.</td>
</tr>
</tbody>
</table>
### 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.

### Table: Scale Scenario Comparison

<table>
<thead>
<tr>
<th>Scale Scenario</th>
<th>RSP1A</th>
<th>RSP1B</th>
<th>RSP2</th>
<th>RSP3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number of supported mroutes</td>
<td>Number of VRF supported</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of supported mroutes</td>
<td>SM-500, SSM-1000</td>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SM-2000, SSM-4000</td>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SM-1000, SSM-2000</td>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4000</td>
<td>20</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

*IP Multicast: Multicast Configuration Guide, Cisco IOS XE Release 3S (Cisco ASR 900 Series)*
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 9: 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 10: 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.

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.

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 utmost 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 85 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.
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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: <code>Device&gt; enable</code></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: <code>Device# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> ip multicast-routing distributed</td>
<td>Enables multicast routing.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
</tbody>
</table>
## Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device(config)# ip multicast-routing distributed</td>
<td>Supports the MVPN VRF instance.</td>
</tr>
</tbody>
</table>

### Step 4

**ip multicast-routing vrf vrf-name distributed**  
**Example:**  
Device(config)# ip multicast-routing vrf vrf1 distributed  

### Step 5

**ip vrf vrf-name**  
**Example:**  
Device(config)# ip vrf vrf1  

### Step 6

**mdt default group-address**  
**Example:**  
Device(config-vrf)# mdt default 232.0.0.1  

## 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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>enable</td>
</tr>
<tr>
<td>Example:</td>
<td>Device&gt; enable</td>
</tr>
</tbody>
</table>
| | Enables privileged EXEC mode.  
| | • Enter your password if prompted. |
| **Step 2** | configure terminal |
| Example: | Router# configure terminal |
| | Enters global configuration mode. |
| **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: | | Exits address family configuration mode and returns to router configuration mode. |
Configuring Multicast VPN

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device(config-router-af)# exit</td>
<td>Enters address family configuration mode to create a VPNv4 address family session.</td>
</tr>
<tr>
<td><strong>Step 8</strong> address-family vpnv4</td>
<td>Enables the VPNv4 address family for this neighbor.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(config-router)# address-family vpnv4</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong> neighbor neighbor-address activate</td>
<td>Enables community and (or) extended community exchange with the specified neighbor.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(config-router-af)# neighbor 192.168.1.1 activate</td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong> neighbor neighbor-address send-community [both</td>
<td>extended</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(config-router-af)# neighbor 192.168.1.1 send-community extended</td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong> end</td>
<td>Exits address family configuration mode and enters privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(config-router-af)# end</td>
<td></td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><strong>enable</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;Device&gt; enable</td>
<td>Enables privileged EXEC mode.&lt;br&gt;• Enter your password if prompted.</td>
</tr>
<tr>
<td>Step 2</td>
<td><strong>configure terminal</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;Device# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Step 3</td>
<td><strong>ip vrf vrf-name</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;Device(config)# ip vrf vrf1</td>
<td>Enters VRF configuration mode and defines the VPN routing instance by assigning a VRF name.&lt;br&gt;See the Example: Configuring MVPN and SSM, on page 96 section for an alternate command.</td>
</tr>
<tr>
<td>Step 4</td>
<td><strong>mdt data group-address-range wildcard-bits [threshold kbps] [list access-list]</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;Device(config-vrf)# mdt data 239.192.20.32/0.0.0.15 threshold 1</td>
<td>Specifies a range of addresses to be used in the data MDT pool.&lt;br&gt;• For the <code>group-address-range</code> and <code>wildcard-bits</code> arguments, specify a multicast group address range. The range is from 224.0.0.1 to 239.255.255.255.&lt;br&gt;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 <code>wildcard-bits</code> argument.&lt;br&gt;• The threshold is in kbps. The range is from 1 through 4294967.&lt;br&gt;• Use the optional <code>list</code> keyword and <code>access-list</code> 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 <code>access-list</code> argument</td>
</tr>
<tr>
<td>Step 5</td>
<td><strong>mdt log-reuse</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;Device(config-vrf)# mdt log-reuse</td>
<td>(Optional) Enables the recording of data MDT reuse and generates a syslog message when a data MDT has been reused.</td>
</tr>
<tr>
<td>Step 6</td>
<td><strong>end</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;Device(config-vrf)# end</td>
<td>Returns to privileged EXEC mode.</td>
</tr>
</tbody>
</table>
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**

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

<table>
<thead>
<tr>
<th>MDT (Route Distinguisher + IPv4)</th>
<th>Router ID</th>
<th>Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>2:200:50.0.0.4</td>
<td>0.0.0.0</td>
<td>0.0.0.0</td>
</tr>
<tr>
<td>MDT group 239.1.1.1</td>
<td>50.0.0.4</td>
<td>50.0.0.4</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>MDT-data send history for VRF - vrf1 for the past 20 minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDT-data group</td>
</tr>
<tr>
<td>--------------------</td>
</tr>
<tr>
<td>10.9.9.8</td>
</tr>
<tr>
<td>10.9.9.9</td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1**
  show ip mroute   | Displays the contents of the IP multicast routing table in the provider's core. |
  **Example:**     |         |

### Purpose

**Command or Action**

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

<table>
<thead>
<tr>
<th>(*, 239.1.1.1), 05:06:08/stopped, RP 0.0.0.0, flags: DCZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incoming interface: Null, RPF nbr 0.0.0.0</td>
</tr>
<tr>
<td>Outgoing interface list:</td>
</tr>
<tr>
<td>TenGigabitEthernet0/2/0, Forward/Sparse-Dense, 01:08:04/stopped</td>
</tr>
<tr>
<td>MVRF VPN_A, Forward/Sparse-Dense, 05:06:08/stopped</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(50.0.0.4, 239.1.1.1), 01:04:27/00:01:31, flags: TZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incoming interface: TenGigabitEthernet0/2/0, RPF nbr 60.0.0.3</td>
</tr>
<tr>
<td>Outgoing interface list:</td>
</tr>
<tr>
<td>MVRF VPN_A, Forward/Sparse-Dense, 01:04:27/stopped</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(50.0.0.2, 239.1.1.1), 05:06:07/00:02:42, flags: T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incoming interface: Loopback50, RPF nbr 0.0.0.0</td>
</tr>
<tr>
<td>Outgoing interface list:</td>
</tr>
<tr>
<td>TenGigabitEthernet0/2/0, Forward/Sparse-Dense, 01:08:04/stopped</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(*, 238.2.2.0), 00:52:26/stopped, flags: DCZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incoming interface: Null, RPF nbr 0.0.0.0</td>
</tr>
<tr>
<td>Outgoing interface list:</td>
</tr>
<tr>
<td>TenGigabitEthernet0/2/0, Forward/Sparse-Dense, 00:52:26/stopped</td>
</tr>
<tr>
<td>MVRF VPN_A, Forward/Sparse-Dense,</td>
</tr>
</tbody>
</table>
**Purpose**

**Command or Action**

<table>
<thead>
<tr>
<th>Time</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00:52:26</td>
<td>stopped</td>
</tr>
<tr>
<td>(.*, 224.0.1.40), 05:09:15/00:02:47, RP 0.0.0.0, flags: DCL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Incoming interface: Null, RPF nbr 0.0.0.0</td>
</tr>
<tr>
<td></td>
<td>Outgoing interface list:</td>
</tr>
<tr>
<td></td>
<td>TenGigabitEthernet0/2/0, Forward/Sparse-Dense,</td>
</tr>
<tr>
<td></td>
<td>01:08:04/stopped</td>
</tr>
<tr>
<td></td>
<td>Loopback50, Forward/Sparse-Dense,</td>
</tr>
<tr>
<td>05:09:15</td>
<td>stopped</td>
</tr>
</tbody>
</table>

**Verifying Information for the Multicast Routes**

**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, 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
```

Displays the multicast routing table in the client's VRF.
Displaying Multicast Forwarding Counters

```
router#show ip mfib vrf test
Entry Flags:  C - Directly Connected, S - Signal, IA - Inherit A flag, 
             ET - Data Rate Exceeds Threshold, K - Keepalive 
             DDE - Data Driven Event, HW - Hardware Installed 
             ME - MoFRR ECMP entry, MNE - MoFRR Non-ECMP entry, MP - MFIB 
             MoFRR Primary, RP - MRIB MoFRR Primary, P - MoFRR Primary 
             MS - MoFRR Entry in Sync, MC - MoFRR entry in MoFRR Client.
I/O Item Flags:  IC - Internal Copy, NP - Not platform switched, 
                 NS - Negate Signalling, SP - Signal Present, 
                 A - Accept, F - Forward, RA - MRIB Accept, RF - MRIB Forward, 
                 MA - MFIB Accept, A2 - Accept backup, 
                 RA2 - MRIB Accept backup, MA2 - MFIB Accept backup

Forwarding Counts:  Pkt Count/Pkts per second/Avg Pkt Size/Kbits per second
Other counts:  Total/RPF failed/Other drops
I/O Item Counts:  FS Pkt Count/PS Pkt Count

VRF test
(*,224.0.0.0/4) Flags: HW
SW Forwarding:  0/0/0/0, Other: 0/0/0
HW Forwarding:  0/0/0/0, Other: 0/0/0
BDI1101 Flags:  NS
Tunnel0, MDT/232.0.0.1 Flags: NS
(*,224.0.1.40) Flags: C HW
SW Forwarding:  0/0/0/0, Other: 0/0/0
HW Forwarding:  0/0/0/0, Other: 0/0/0
Tunnel0, MDT/232.0.0.1 Flags: F IC NS
Pkts: 0/0

(*,232.0.0.0/8) Flags: HW
SW Forwarding:  0/0/0/0, Other: 0/0/0
HW Forwarding:  0/0/0/0, Other: 0/0/0

(10.11.11.2,232.10.0.1) Flags: ET HW
SW Forwarding:  0/0/0/0, Other: 0/0/0
HW Forwarding:  571756659/1420644/68/754717, Other: 0/0/0
BDI1101 Flags:  A
Tunnel0, MDT/232.5.1.0 Flags: F NS
Pkts: 0/0
```

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:
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
```

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
mdt default 239.1.1.1
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
```
Example: Configuring BGP

```
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 7

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, on page 101
- Prerequisites for Configuring Multicast VPN Extranet Support, on page 101
- Restrictions for Configuring Multicast VPN Extranet Support, on page 102
- Information About Multicast VPN Extranet Support, on page 102
- How to Configure Multicast VPN Extranet Support, on page 107
- Configuration Examples for Multicast VPN Extranet Support, on page 115

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 lookup².

- 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 11: 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.
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 an 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.
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 14: 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 (the MVRF for VPN-Green) to the receiver MVRF (the MVRF for VPN-Red).
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 out: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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router&gt; enable</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| **Step 3** ip vrf vrf-name | 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. |
| Example: | |
| Router(config)# ip vrf VPN-Red | |

| **Step 4** rd route-distinguisher | 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 |
| Example: | |
| Router(config-vrf)# rd 55:2222 | |

| **Step 5** vpn id out:vpn-index | 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. |
| Example: | |
| Router(config-vrf)# vpn id 1:2 | |

| **Step 6** route-target { export | import | both } route-target-ext-community | Creates a list of import, export, or import and export route target communities for the specified VRF.  
Enter either an 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. |
| Example: | |
| Router(config-vrf)# route-target import 55:1111 | |

| **Step 7** mdt default mpls mldp PE/P router_loopback-ip | 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. |
| Example: | |
| Router(config-vrf)# mdt default mpls mldp 20.100.0.4 | |

| **Step 8** mdt data mpls mldp number of data mdt | Creating data mdt by mentioning the number of data mdts to be created. The range is from 1 to 5000. |
| Example: | |
| Router(config-vrf)# mdt default mpls mldp 10.1.1.10 | |

| **Step 9** mdt data threshold threshold value | The threshold is in kbps. The range is from 1 through 4294967. |
| Example: | |
| Router(config-vrf)# mdt default threshold 1 | |

| **Step 10** end | Exits VRF configuration mode and returns to privileged EXEC mode. |
| Example: | |
Configuring Multicast VPN Extranet Support

### 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 MVVPN 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 16: 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 out: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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>enable</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>configure terminal</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>ip vrf vrf-name</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# ip vrf VPN-Red</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Defines the VPN routing instance by assigning a VRF name and enters VRF configuration mode.</td>
</tr>
<tr>
<td></td>
<td>• The vrf-name argument is the name assigned to a VRF.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>rd route-distinguisher</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-vrf)# rd 55:1111</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Creates routing and forwarding tables.</td>
</tr>
<tr>
<td></td>
<td>• 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:</td>
</tr>
<tr>
<td></td>
<td>• 16-bit autonomous system number: your 32-bit number, for example, 101:3</td>
</tr>
<tr>
<td></td>
<td>• 32-bit IP address: your 16-bit number, for example, 192.168.122.15:1</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>vpn id out:vpn-index</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-vrf)# vpn id 1:2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>route-target { export</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-vrf)# route-target import 55:1111</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Creates a list of import, export, or import and export route target communities for the specified VRF.</td>
</tr>
<tr>
<td></td>
<td>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).</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>This command works only if BGP is running.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 7</strong> mdt default mpls mldp PE/P router_loopback-ip</td>
<td>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.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-vrf)# mdt default mpls mldp 20.100.0.4</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> mdt data mpls mldp number of data mdt</td>
<td>Creating data mdt by mentioning the number of data mdts to be created. The range is from 1 to 5000.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-vrf)# mdt default mpls mldp 10.1.1.10</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong> mdt data threshold threshold value</td>
<td>The threshold is in kbps. The range is from 1 through 4294967.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-vrf)# mdt default threshold 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong> end</td>
<td>Exits VRF configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-vrf)# end</td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong> show ip mroute [vrf vrf-name] [group-address] verbose</td>
<td>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.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# show ip mroute vrf vpn_a 224.1.1.1 verbose</td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong> show mls ip multicast group group-address</td>
<td>(Optional) Displays MLS information related to a specific multicast group.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# show mls ip multicast group 232.3.3.3</td>
<td></td>
</tr>
</tbody>
</table>

**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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><code>enable</code></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>(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.</td>
</tr>
<tr>
<td>`ip mroute vrf vrf-name source-address mask fallback-lookup {global</td>
<td>vrf vrf-name} [distance]`</td>
</tr>
<tr>
<td>Example:</td>
<td>• The <code>vrf</code> keyword and <code>vrf-name</code> argument are used to explicitly specify a VRF as the source MVRF.</td>
</tr>
<tr>
<td>Router(config)# ip mroute vrf VPN-Red 224.100.0.5 255.255.255.255 fallback-lookup vrf VPN-Green</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Exits global configuration mode and enters privileged EXEC mode.</td>
</tr>
<tr>
<td><code>end</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# end</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>(Optional) Displays the contents of the IP multicast mroute table for a specific group address.</td>
</tr>
<tr>
<td><code>show ip mroute [vrf vrf-name] group-address</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# show ip mroute 224.100.0.5</td>
<td></td>
</tr>
</tbody>
</table>

**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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>enable</td>
</tr>
<tr>
<td>Example:</td>
<td>Router&gt; enable</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>configure terminal</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# configure terminal</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Use the following commands:</td>
</tr>
<tr>
<td></td>
<td>• ip multicast [vrf receiver-vrf-name] rpf select {global</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# ip multicast vrf VPN-Green rpf select vrf VPN-Red group-list 1</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Repeat step 3 to create additional group-based VRF selection policies.</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>end</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# end</td>
</tr>
</tbody>
</table>
### 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 17: 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

```conf
! 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,
Example Configuring the Receiver VRF on the Source PE Router - Option 1 (SSC)

The following are sample outputs from the `show ip mroute` command on PE1 and PE2, when PE1 and PE2 are Catalyst 6500 series switches that have been 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.
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

PE2# show mls ip multicast group 232.3.3.3
Multicast hardware switched flows:
  (10.1.0.1, 232.3.3.3) Incoming interface: Gi4/1, Packets switched: 808
Hardware switched outgoing interfaces:
  MVRF VPN-Red
  RPF-MFD installed
(10.2.0.2, 232.3.3.3) Incoming interface: Lo0, Packets switched: 808
Hardware switched outgoing interfaces:
  Gi4/1
  RPF-MFD installed
Total hardware switched flows: 2

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,
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
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 a Catalyst 6500 Series Switch Configured for MVPN Extranet Support

The following are sample outputs from the `show ip mroute` command on PE1, when PE1 is a Catalyst 6500 series switch 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
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
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,
```

```
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
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
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.
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 a Catalyst 6500 Series Switches Configured for MVPN Extranet Support

The following is a sample output from the `show ip mroute` command on PE1, when PE1 is a Catalyst 6500 series switch 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

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:
  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, 00:00:12/00:03:18

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 a 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

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:
  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, 00:00:12/00:03:18, H
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: Tunnell, RPF nbr 10.1.0.1
Outgoing interface list:
  GigabitEthernet9/1, 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: Tunnell, RPF nbr 10.1.0.1
Outgoing interface list:
  GigabitEthernet9/1, Forward/Sparse-Dense, 00:00:00/00:03:29, H

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 a Catalyst 6500 Series Switch Configured for MVPN Extranet Support

The following are sample output from the `show ip mroute` command on PE2, when PE2 is a Catalyst 6500 series switch 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: Tunnell, 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: Tunnell, 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.
PE2 Configuration

```plaintext
!
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
```

Figure 18: Topology for MVPN Extranet Support Option 2 Configuration Example
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
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
States in the Global Table on PE1 and PE2 for the MDT Default Group 232.1.1.1 When PE1 and PE2 Are Catalyst 6500 Series Switches Configured for MVPN Extranet Support

The following are sample output from the `show ip mroute` command on PE1 and PE2, when PE1 and PE2 are Catalyst 6500 series switches that have been 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), 00:02:04/00:03:09, 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

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
(10.2.0.2, 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 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.
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 a Catalyst 6500 Series Switch Configured for MVPN Extranet Support

The following are sample output from the `show ip mroute` commands on PE1, when PE1 is a Catalyst 6500 series switch 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: Ethernet3/0, RPF nbr 10.1.1.5
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: Ethernet3/0, RPF nbr 10.1.1.5
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 MVPF mroute entries linked to it.
Example Configuring the Source VRF on the Receiver PE - Option 2 (RSC)

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 a Catalyst 6500 Series Switch Configured for MVPN Extranet Support

The following are sample output from the show ip mroute command on PE2, when PE2 is a Catalyst 6500 series switch 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: S
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 a Catalyst 6500 Series Switch Configured for MVPN Extranet Support

The following are sample output from the `show ip mroute` command on PE2, when PE2 is a Catalyst 6500 series switch 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: S
Incoming interface: Tunnel0, RPF nbr 10.1.0.1, using vrf VPN-Green, RPF-MFD
Outgoing interface list:
  GigabitEthernet9/1, Forward/Sparse-Dense, 00:01:32/00:03:01, H

IP Multicast: Multicast Configuration Guide, Cisco IOS XE Release 3S (Cisco ASR 900 Series)
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,
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,
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 examples show 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, NF - Not platform switched,
                NS - Negate Signalling, SP - Signal Present,
                A - Accept, F - Forward, RA - MRIB Accept, RF - MRIB

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.

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, NF - Not platform switched,
                NS - Negate Signalling, SP - Signal Present,
                A - Accept, F - Forward, RA - MRIB Accept, RF - MRIB

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
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 8

IPv6 Multicast PIM

This module describes how to configure IPv6 Multicast PIM features.

• Prerequisites for IPv6 Multicast, on page 135
• Restrictions for IPv6 Multicast, on page 135
• Information About IPv6 Multicast, on page 136
• How to Configure IPv6 Multicast, on page 145
• Using MFIB in IPv6 Multicast, on page 157
• Configuration Examples for IPv6 Multicast, on page 159

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 router does not support:

• IPv6 Multicast Listener Discovery (MLD) snooping
• Multiprotocol BGP Extensions for multicast
• Equal Cost Multipath (ECMP) for multicast
• Multicast Virtual Private Network (MVPN)
• Multicast QoS for IPv6
• Dynamic Domain Name System (DNS) PIM Source Specific Multicast (SSM) mapping for multicast
• Multicast Source Discovery Protocol (MSDP)
• Embedded RP is not supported on RSP3 module.

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 10:21:07.287: %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 MLD reports join rate is increased. When 90% of the system memory is used, the MLD 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.
Protocol Independent Multicast

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

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

PIM-Sparse Mode

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

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

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

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

A multicast data sender sends data destined for a multicast group. The designated router (DR) of the sender takes those data packets, unicast-encapsulates them, and sends them directly to the RP. The RP receives these encapsulated data packets, de-encapsulates them, and forwards them onto the shared tree. The packets then follow the (*, G) multicast tree state in the devices on the RP tree, being replicated wherever the RP tree branches, and eventually reaching all the receivers for that multicast group. The process of encapsulating data packets to the RP is called registering, and the encapsulation packets are called PIM register packets.
Designated Router

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

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

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

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

**Figure 20: Designated Router Election on a Multiaccess Segment**

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

---

**Tip**

Two PIM devices are neighbors if there is a direct connection between them. To display your PIM neighbors, use the `show ipv6 pim neighbor` command in privileged EXEC mode.
The DR election process is required only on multiaccess LANs.

**Note**

Embedded RP is not supported on Cisco RSP3 Module.

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.
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.

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

Note

PIM IPv6 Stub Routing

The PIM stub routing feature reduces resource usage by moving routed traffic closer to the end user.

In a network using PIM stub routing, the only allowable route for IPv6 traffic to the user is through a switch that is configured with PIM stub routing. PIM passive interfaces are connected to Layer 2 access domains, such as VLANs, or to interfaces that are connected to other Layer 2 devices. Only directly connected multicast receivers and sources are allowed in the Layer 2 access domains. The PIM passive interfaces do not send or process any received PIM control packets.

When using PIM stub routing, you should configure the distribution and remote routers to use IPv6 multicast routing and configure only the switch as a PIM stub router. The switch does not route transit traffic between distribution routers. You also need to configure a routed uplink port on the switch. The switch uplink port cannot be used with SVIs.

You must also configure EIGRP stub routing when configuring PIM stub routing on the switch.

The redundant PIM stub router topology is not supported. The redundant topology exists when there is more than one PIM router forwarding multicast traffic to a single access domain. PIM messages are blocked, and the PIM assert and designated router election mechanisms are not supported on the PIM passive interfaces. Only the non-redundant access router topology is supported by the PIM stub feature. By using a non-redundant topology, the PIM passive interface assumes that it is the only interface and designated router on that access domain.
In the figure shown below, Switch A routed uplink port 25 is connected to the router and PIM stub routing is enabled on the VLAN 100 interfaces and on Host 3. This configuration allows the directly connected hosts to receive traffic from multicast source.

*Figure 22: PIM Stub Router Configuration*

---

**MRIB**

The Multicast Routing Information Base (MRIB) is a protocol-independent repository of multicast routing entries instantiated by multicast routing protocols (routing clients). Its main function is to provide independence between routing protocols and the Multicast Forwarding Information Base (MFIB). It also acts as a coordination and communication point among its clients.

Routing clients use the services provided by the MRIB to instantiate routing entries and retrieve changes made to routing entries by other clients. Besides routing clients, MRIB also has forwarding clients (MFIB instances) and special clients such as MLD. MFIB retrieves its forwarding entries from MRIB and notifies the MRIB of any events related to packet reception. These notifications can either be explicitly requested by routing clients or spontaneously generated by the MFIB.

Another important function of the MRIB is to allow for the coordination of multiple routing clients in establishing multicast connectivity within the same multicast session. MRIB also allows for the coordination between MLD and routing protocols.

**MFIB**

The MFIB is a platform-independent and routing-protocol-independent library for IPv6 software. Its main purpose is to provide a Cisco IOS platform with an interface with which to read the IPv6 multicast forwarding table and notifications when the forwarding table changes. The information provided by the MFIB has clearly defined forwarding semantics and is designed to make it easy for the platform to translate to its specific hardware or software forwarding mechanisms.

When routing or topology changes occur in the network, the IPv6 routing table is updated, and those changes are reflected in the MFIB. The MFIB maintains next-hop address information based on the information in the IPv6 routing table. Because there is a one-to-one correlation between MFIB entries and routing table entries, the MFIB contains all known routes and eliminates the need for route cache maintenance that is associated with switching paths such as fast switching and optimum switching.
MFIB

Distributed MFIB has its significance only in a stacked environment where the Master distributes the MFIB information to the other stack members. In the following section the line cards are nothing but the member switches in the stack.

MFIB (MFIB) is used to switch multicast IPv6 packets on distributed platforms. MFIB may also contain platform-specific information on replication across line cards. The basic MFIB routines that implement the core of the forwarding logic are common to all forwarding environments.

MFIB implements the following functions:

- Relays data-driven protocol events generated in the line cards to PIM.
- Provides an MFIB platform application program interface (API) to propagate MFIB changes to platform-specific code responsible for programming the hardware acceleration engine. This API also includes entry points to switch a packet in software (necessary if the packet is triggering a data-driven event) and to upload traffic statistics to the software.

The combination of MFIB and MRIB subsystems also allows the switch to have a "customized" copy of the MFIB database in each line card and to transport MFIB-related platform-specific information from the RP to the line cards.

IPv6 Multicast Process Switching and Fast Switching

A unified MFIB is used to provide both fast switching and process switching support for PIM-SM and PIM-SSM in IPv6 multicast. In process switching, the must examine, rewrite, and forward each packet. The packet is first received and copied into the system memory. The switch then looks up the Layer 3 network address in the routing table. The Layer 2 frame is then rewritten with the next-hop destination address and sent to the outgoing interface. The also computes the cyclic redundancy check (CRC). This switching method is the least scalable method for switching IPv6 packets.

IPv6 multicast fast switching allows switches to provide better packet forwarding performance than process switching. Information conventionally stored in a route cache is stored in several data structures for IPv6 multicast switching. The data structures provide optimized lookup for efficient packet forwarding.

In IPv6 multicast forwarding, the first packet is fast-switched if the PIM protocol logic allows it. In IPv6 multicast fast switching, the MAC encapsulation header is precomputed. IPv6 multicast fast switching uses the MFIB to make IPv6 destination prefix-based switching decisions. In addition to the MFIB, IPv6 multicast fast switching uses adjacency tables to prepend Layer 2 addressing information. The adjacency table maintains Layer 2 next-hop addresses for all MFIB entries.

The adjacency table is populated as adjacencies are discovered. Each time an adjacency entry is created (such as through ARP), a link-layer header for that adjacent node is precomputed and stored in the adjacency table. Once a route is determined, it points to a next hop and corresponding adjacency entry. It is subsequently used for encapsulation during switching of packets.

A route might have several paths to a destination prefix, such as when a switch is configured for simultaneous load balancing and redundancy. For each resolved path, a pointer is added for the adjacency corresponding to the next-hop interface for that path. This mechanism is used for load balancing across several paths.
Enabling IPv6 Multicast Routing

Beginning in privileged EXEC mode, follow these steps:

### Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure terminal</td>
<td>Enter global configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>ipv6 multicast-routing</td>
<td>Enables multicast routing on all IPv6-enabled interfaces and enables multicast forwarding for PIM and MLD on all enabled interfaces of the switch.</td>
</tr>
<tr>
<td>Example:</td>
<td>(config)# ipv6 multicast-routing</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>copy running-config startup-config</td>
<td>(Optional) Save your entries in the configuration file.</td>
</tr>
</tbody>
</table>

**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.
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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Device&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> ipv6 multicast-routing [vrf vrf-name]</td>
<td>Enables multicast routing on all IPv6-enabled interfaces and enables multicast forwarding for PIM and MLD on all enabled interfaces of the device.</td>
</tr>
<tr>
<td>Example: Device(config)# ipv6 multicast-routing</td>
<td>• 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.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Device&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> ipv6 pim [vrf vrf-name] spt-threshold infinity [group-list access-list-name]</td>
<td>Configures when a PIM leaf device joins the SPT for the specified groups.</td>
</tr>
<tr>
<td>Example: Device(config)# ipv6 pim spt-threshold infinity group-list acc-grp-1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> ipv6 pim [vrf vrf-name] accept-register {list access-list</td>
<td>route-map map-name}</td>
</tr>
<tr>
<td>Example: Device(config)# ipv6 pim accept-register route-map reg-filter</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> interface type number</td>
<td>Specifies an interface type and number, and places the device in interface configuration mode.</td>
</tr>
<tr>
<td>Example: Device(config)# interface gigabitethernet 0/1/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> ipv6 pim dr-priority value</td>
<td>Configures the DR priority on a PIM device.</td>
</tr>
<tr>
<td>Example: Device(config-if)# ipv6 pim dr-priority 3</td>
<td></td>
</tr>
</tbody>
</table>
### 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] [type interface 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 interface-number]`
10. `debug ipv6 pim [group-name | group-address | interface interface-type | bsr | group | mvpn | neighbor]`

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
</tbody>
</table>

---

**Purpose Command or Action**

<table>
<thead>
<tr>
<th>Step 7</th>
<th><code>ipv6 pim hello-interval seconds</code> Example:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Device(config-if)# ipv6 pim hello-interval 45</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 8</th>
<th><code>ipv6 pim join-prune-interval seconds</code> Example:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Device(config-if)# ipv6 pim join-prune-interval 75</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 9</th>
<th><code>exit</code> Example:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Device(config-if)# exit</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 10</th>
<th><code>show ipv6 pim [vrf vrf-name] join-prune statistic [interface-type]</code> Example:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Device# show ipv6 pim join-prune statistic</td>
</tr>
</tbody>
</table>

---
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 2</strong> configure terminal Example: Device# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 3</strong> ipv6 pim [vrf vrf-name] rp-address ipv6-address [group-access-list] [bidir] Example: Device(config)# ipv6 pim rp-address 2001:0DB8:01:800:200E:8C6C acc-grp-1</td>
<td>Configures the address of a PIM RP for a particular group range.</td>
</tr>
<tr>
<td><strong>Step 4</strong> end Example: Device(config-if)# end</td>
<td>Returns to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Step 5</strong> show ipv6 pim [vrf vrf-name] interface [state-on] [state-off] [type number] Example: Device# show ipv6 pim interface</td>
<td>Displays information about interfaces configured for PIM.</td>
</tr>
<tr>
<td><strong>Step 6</strong> show ipv6 pim [vrf vrf-name] group-map [group-name</td>
<td>group-address]</td>
</tr>
<tr>
<td><strong>Step 7</strong> show ipv6 pim [vrf vrf-name] neighbor [detail] [interface-type interface-number</td>
<td>count] Example: Device# show ipv6 pim neighbor</td>
</tr>
<tr>
<td><strong>Step 8</strong> show ipv6 pim [vrf vrf-name] range-list[config] [rp-address</td>
<td>rp-name] Example: Device# show ipv6 pim range-list</td>
</tr>
<tr>
<td><strong>Step 9</strong> show ipv6 pim [vrf vrf-name] tunnel [interface-type interface-number] Example: Device# show ipv6 pim tunnel</td>
<td>Displays information about the PIM register encapsulation and de-encapsulation tunnels on an interface.</td>
</tr>
</tbody>
</table>
### Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>debug ipv6 pim</code></td>
<td>Enables debugging on PIM protocol activity.</td>
</tr>
<tr>
<td><code>debug ipv6 pim</code></td>
<td>Example:</td>
</tr>
<tr>
<td><code>Device# debug ipv6 pim</code></td>
<td></td>
</tr>
</tbody>
</table>

### 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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><code>Device&gt; enable</code></td>
<td>* Enter your password if prompted.</td>
</tr>
<tr>
<td><code>clear ipv6 pim [vrf vrf-name] traffic</code></td>
<td>Resets the PIM traffic counters.</td>
</tr>
<tr>
<td><code>Device# clear ipv6 pim traffic</code></td>
<td></td>
</tr>
<tr>
<td><code>show ipv6 pim [vrf vrf-name] traffic</code></td>
<td>Displays the PIM traffic counters.</td>
</tr>
<tr>
<td><code>Device# show ipv6 pim traffic</code></td>
<td></td>
</tr>
</tbody>
</table>

### 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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><code>enable</code></td>
</tr>
<tr>
<td></td>
<td>Example: <code>Device&gt; enable</code></td>
</tr>
<tr>
<td></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>`clear ipv6 pim [vrf vrf-name] topology [group-name</td>
</tr>
<tr>
<td></td>
<td>Example: <code>Device# clear ipv6 pim topology FF04::10</code></td>
</tr>
<tr>
<td></td>
<td>Clears the PIM topology table.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>`show ipv6 mrib [vrf vrf-name] client [filter] [name {client-name</td>
</tr>
<tr>
<td></td>
<td>Example: <code>Device# show ipv6 mrib client</code></td>
</tr>
<tr>
<td></td>
<td>Displays multicast-related information about an interface.</td>
</tr>
</tbody>
</table>
| **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. |
### Purpose

**Command or Action** | **Purpose**  
--- | ---  
Device# debug ipv6 mrib client | Enables debugging on MRIB I/O events.  

**Step 7**  
**debug ipv6 mrib [vrf vrf-name] io**  
**Example:**  
Device# debug ipv6 mrib io

| **Step 8**  
**debug ipv6 mrib proxy**  
**Example:**  
Device# debug ipv6 mrib proxy | Enables debugging on MRIB proxy activity between the route processor and line cards on distributed router platforms.  

| **Step 9**  
**debug ipv6 mrib [vrf vrf-name] route [group-name | group-address]**  
**Example:**  
Device# debug ipv6 mrib route | Displays information about MRIB routing entry-related activity.  

| **Step 10**  
**debug ipv6 mrib [vrf vrf-name] table**  
**Example:**  
Device# debug ipv6 mrib table | Enables debugging on MRIB table management activity.  

---

#### 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:** | Enables privileged EXEC mode.  
  • Enter your password if prompted.  

---
### Command or Action

<table>
<thead>
<tr>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>

#### Step 2

**configure terminal**

**Example:**

Device# configure terminal

#### Step 3

**interface type number**

**Example:**

Device(config)# interface gigabitethernet 0/1/0

#### Step 4

**no ipv6 pim**

**Example:**

Device(config-if)# no ipv6 pim

---

**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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
</tbody>
</table>

<p>| <strong>Step 2</strong> configure terminal | Enters global configuration mode. |
| <strong>Example:</strong> | |</p>
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device# configure terminal</td>
<td></td>
</tr>
</tbody>
</table>

**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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device&gt; enable</td>
<td>- Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> ipv6 mld [vrf vrf-name] ssm-map enable</td>
<td>Enables the SSM mapping feature for groups in the configured SSM range.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config)# ipv6 mld ssm-map enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> no ipv6 mld [vrf vrf-name] ssm-map query dns</td>
<td>Disables DNS-based SSM mapping.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config)# no ipv6 mld ssm-map query dns</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> ipv6 mld [vrf vrf-name] ssm-map static access-list source-address</td>
<td>Configures static SSM mappings.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config)# ipv6 mld ssm-map static SSM_MAP_ACL_2 2001:DB8:1::1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> end</td>
<td>Returns to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config-if)# end</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> show ipv6 mld [vrf vrf-name] ssm-map [source-address]</td>
<td>Displays SSM mapping information.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# show ipv6 mld ssm-map</td>
<td></td>
</tr>
</tbody>
</table>

## 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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** | `enable` | Enables privileged EXEC mode.  
**Example:**  
Device> enable |
| **Step 2** | `configure terminal` | Enters global configuration mode.  
**Example:**  
Device# configure terminal |
| **Step 3** | `ipv6 multicast pim-passive-enable` | Enables the PIM passive feature on an IPv6 device.  
**Example:**  
Device(config)# ipv6 multicast pim-passive-enable |
| **Step 4** | `ipv6 mld state-limit number` | (Optional) Specifies maximum number of dynamic MLD groups allowed on a router.  
**Example:**  
Device(config)# ipv6 mld state-limit 100 |
| **Step 5** | `interface type number` | Specifies an interface type and number, and places the device in interface configuration mode.  
**Example:**  
Device(config)# interface GigabitEthernet 1/0/0 |
| **Step 6** | `ipv6 pim passive` | Enables the PIM passive feature on a specific interface.  
**Example:**  
Device(config-if)# ipv6 pim passive |
| **Step 7** | `ipv6 mld limit number` | (Optional) Configure the per-interface MLD state limit. You can use this command to limit the dynamic MLD groups joined.  
**Example:**  
Device(config-if)# ipv6 mld limit 300 |
| **Step 8** | `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
Configuring a BSR

The tasks included here are described below.

Configuring a BSR and Verifying BSR Information

Beginning in privileged EXEC mode, follow these steps:

**Procedure**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure terminal</td>
<td>Enter global configuration mode.</td>
</tr>
<tr>
<td>2</td>
<td>ipv6 pim bsr candidate bsr</td>
<td>Configures a switch to be a candidate BSR.</td>
</tr>
<tr>
<td></td>
<td>ipv6-address[hash-mask-length] [priority priority-value]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(config) # ipv6 pim bsr candidate bsr</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2001:DB8:3000:3000::42 124 priority 10</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>interface type number</td>
<td>Specifies an interface type and number, and places the switch in interface configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(config) # interface GigabitEthernet 1/0/1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>ipv6 pim bsr border</td>
<td>Specifies an interface type and number, and places the switch in interface configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(config-if) # ipv6 pim bsr border</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>exit</td>
<td>Enter this command twice to exit interface configuration mode and enter privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(config-if) # exit</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>show ipv6 pim bsr {election</td>
<td>rp-cache</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(config-if) # show ipv6 pim bsr election</td>
<td></td>
</tr>
</tbody>
</table>
### Sending PIM RP Advertisements to the BSR

Beginning in privileged EXEC mode, follow these steps:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure terminal</td>
<td>Enter global configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>ipv6 pim bsr candidate rp ipv6-address [group-list access-list-name] [priority priority-value] [interval seconds]</td>
<td>Sends PIM RP advertisements to the BSR.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(config) # ipv6 pim bsr candidate rp 2001:DB8:3000:3000::42 priority 0</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>interface type number</td>
<td>Specifies an interface type and number, and places the switch in interface configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(config) # interface GigabitEthernet 1/0/1</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>ipv6 pim bsr border</td>
<td>Configures a border for all BSMs of any scope on a specified interface.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(config-if) # ipv6 pim bsr border</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>copy running-config startup-config</td>
<td>(Optional) Save your entries in the configuration file.</td>
</tr>
</tbody>
</table>

### Using MFIB in IPv6 Multicast

Multicast forwarding is automatically enabled when IPv6 multicast routing is enabled.

### Verifying MFIB Operation in IPv6 Multicast

Beginning in privileged EXEC mode, follow these steps:

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>show ipv6 mfib [</td>
<td>verbose</td>
</tr>
</tbody>
</table>
### Command or Action

**Example:**

```
# show ipv6 mfib
```

**Step 2**

```
show ipv6 mfib [all | linkscope | group-name | group-address [source-name | source-address]] count
```

**Example:**

```
# show ipv6 mfib ff07::1
```

**Displaysthe contentsof the IPv6 multicastroutingtable.**

**Step 3**

```
show ipv6 mfib interface
```

**Example:**

```
# show ipv6 mfib interface
```

**Displays informationabout IPv6 multicast-enabled interfaces and their forwarding status.**

**Step 4**

```
show ipv6 mfib status
```

**Example:**

```
# show ipv6 mfib status
```

**Displays general MFIB configuration and operational status.**

**Step 5**

```
show ipv6 mfib summary
```

**Example:**

```
# show ipv6 mfib summary
```

**Displays summary information about the number of IPv6 MFIB entries and interfaces.**

**Step 6**

```
debug ipv6 mfib [group-name | group-address] [adjacency | db | fs | init | interface | mrib [detail] | nat | pak | platform | ppr | ps | signal | table]
```

**Example:**

```
# debug ipv6 mfib FF04::10 pak
```

**Enables debugging output on the IPv6 MFIB.**

---

### Resetting MFIB Traffic Counters

**Beginning in privileged EXEC mode, follow these steps:**

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1**
```
clear ipv6 mfib counters [group-name | group-address [source-address | source-name]]
```

**Example:**

```
# clear ipv6 mfib counters FF04::10
```

**Resets all active MFIB traffic counters.**
Sample Output from the show ipv6 rpf Command

The following example displays RPF information for the unicast host with the IPv6 address of 2001:DB8:1:1:2:

```
Router# show ipv6 rpf 2001:DB8:1:1:2
RPF information for 2001:DB8:1:1:2
 RPF interface:GigabitEthernet3/2/0
 RPF neighbor:FE80::40:1:3
 RPF route/mask:20::/64
 RPF type:Unicast
 RPF recursion count:0
 Metric preference:110
 Metric:30
```

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
```

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 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 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
Example: Displaying Information About PIM Traffic

Device# show ipv6 pim traffic

PIM Traffic Counters
Elapsed time since counters cleared:00:05:29

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Property Value</th>
<th>Property Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valid PIM Packets</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>Hello</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>Join-Prune</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Register</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Register Stop</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Assert</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Bidir DF Election</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Malformed Packets</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Bad Checksums</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Send Errors</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Packet Sent on Loopback Errors</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Packets Received on PIM-disabled Interface</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Packets Received with Unknown PIM Version</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

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
IPv6 Multicast Listener Discovery Protocol

• Finding Feature Information, on page 163
• Restrictions for IPv6 Multicast Listener Discovery Protocol, on page 163
• Information About IPv6 Multicast Listener Discovery Protocol, on page 164
• How to Configure IPv6 Multicast Listener Discovery Protocol, on page 166
• Verifying IPv6 Multicast Listener Discovery Protocol, on page 173

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 at the end of this module.

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 IPv6 Multicast Listener Discovery Protocol

• MLD snooping is not supported. IPv6 multicast traffic is flooded to all Ethernet Flow Points (EFPs) or Trunk EFPs (TEFPs) associated with a bridge domain.
• MLD proxy is not supported.
• For RSP1A, more than 1000 IPv6 multicast routes are not supported.
• For RSP1B, more than 2000 IPv6 multicast routes are not supported.
• IPv6 Multicast Listener Discovery protocol is not supported on the ASR 900 RSP3 module.
Information About IPv6 Multicast Listener Discovery Protocol

IPv6 Multicast Overview

An IPv6 multicast group is an arbitrary group of receivers that want to receive a particular data stream. This group has no physical or geographical boundaries; receivers can be located anywhere on the Internet or in any private network. Receivers that are interested in receiving data flowing to a particular group must join the group by signaling their local device. This signaling is achieved with the MLD protocol.

Devices use the MLD protocol to learn whether or not members of a group are present on their directly attached subnets. Hosts join multicast groups by sending MLD report messages. The network then delivers data to a potentially unlimited number of receivers, using only one copy of the multicast data on each subnet. IPv6 hosts that wish to receive the traffic are known as group members.

Packets delivered to group members are identified by a single multicast group address. Multicast packets are delivered to a group using best-effort reliability, just like IPv6 unicast packets.

The multicast environment consists of senders and receivers. Any host, regardless of whether it is a member of a group, can send to a group. However, only the members of a group receive the message.

A multicast address is chosen for the receivers in a multicast group. Senders use this address as the destination address of a datagram to reach all members of the group.

Membership in a multicast group is dynamic; hosts can join and leave at any time. There is no restriction on the location or number of members in a multicast group. A host can be a member of more than one multicast group at a time.

How active a multicast group is, its duration, and its membership can vary from group to group and from time to time. A group that has members may have no activity.

IPv6 Multicast Routing Implementation

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

- MLD is used by IPv6 devices to discover multicast listeners 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 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.
Multicast Listener Discovery Protocol for IPv6

To start implementing multicasting in the campus network, users must first define who receives the multicast. The MLD protocol is used by IPv6 devices to discover the presence of multicast listeners (for example, nodes that want to receive multicast packets) on their directly attached links, and to discover specifically which multicast addresses are of interest to those neighboring nodes. It is used for discovering local group and source-specific group membership. The MLD protocol provides a means to automatically control and limit the flow of multicast traffic throughout your network with the use of special multicast queriers and hosts.

The difference between multicast queriers and hosts is as follows:

- A querier is a network device that sends query messages to discover which network devices are members of a given multicast group.

- A host is a receiver that sends report messages to inform the querier of a host membership.

A set of queriers and hosts that receive multicast data streams from the same source is called a multicast group. Queriers and hosts use MLD reports to join and leave multicast groups and to begin receiving group traffic.

MLD uses the Internet Control Message Protocol (ICMP) to carry its messages. All MLD messages are link-local with a hop limit of 1, and they all have the alert option set. The alert option implies an implementation of the hop-by-hop option header.

MLD has three types of messages:

- **Query**—General, group-specific, and multicast-address-specific. In a query message, the multicast address field is set to 0 when MLD sends a general query. The general query learns which multicast addresses have listeners on an attached link.

  **Note**  
  Group-specific and multicast-address-specific queries are the same. A group address is a multicast address.

- **Report**—In a report message, the multicast address field is that of the specific IPv6 multicast address to which the sender is listening.

- **Done**—In a done message, the multicast address field is that of the specific IPv6 multicast address to which the source of the MLD message is no longer listening.
An MLD report must be sent with a valid IPv6 link-local source address, or the unspecified address (::), if the sending interface has not yet acquired a valid link-local address. Sending reports with the unspecified address is allowed to support the use of IPv6 multicast in the Neighbor Discovery Protocol.

For stateless autoconfiguration, a node is required to join several IPv6 multicast groups in order to perform duplicate address detection (DAD). Prior to DAD, the only address the reporting node has for the sending interface is a tentative one, which cannot be used for communication. Therefore, the unspecified address must be used.

MLD states that result from MLD version 2 or MLD version 1 membership reports can be limited globally or by interface. The MLD group limits feature provides protection against denial of service (DoS) attacks caused by MLD packets. Membership reports in excess of the configured limits is not entered in the MLD cache, and traffic for those excess membership reports will not be forwarded.

MLD provides support for source filtering. Source filtering allows a node to report interest in listening to packets only from specific source addresses (as required to support SSM), or from all addresses except specific source addresses sent to a particular multicast address.

When a host using MLD version 1 sends a leave message, the device needs to send query messages to reconfirm that this host was the last MLD version 1 host joined to the group before it can stop forwarding traffic. This function takes about 2 seconds. This "leave latency" is also present in IGMP version 2 for IPv4 multicast.

**MLD Access Group**

MLD access groups provide receiver access control in Cisco IPv6 multicast devices. This feature limits the list of groups a receiver can join, and it allows or denies sources used to join SSM channels.

**How to Configure IPv6 Multicast Listener Discovery Protocol**

**Enabling IPv6 Multicast Routing**

To enable IPv6 multicast routing, complete the following steps:

**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]`
4. `end`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
</tbody>
</table>
### Command or Action

**Example:**

Device> enable

**Step 2**
configure terminal

**Example:**

Device# configure terminal

**Step 3**
ipv6 multicast-routing [vrf vrf-name]

**Example:**

Device(config)# ipv6 multicast-routing

**Step 4**
end

**Example:**

Device(config)# end

---

### Customizing MLD on an Interface

To customize MLD on an interface, complete the following steps:

**SUMMARY STEPS**

1. enable
2. configure terminal
3. ipv6 mld state-limit number
4. ipv6 mld [vrf vrf-name] ssm-map enable
5. interface type number
6. ipv6 mld access-group access-list-name
7. ipv6 mld static-group [group-address] [[include exclude] {source-address | source-list [acl]}
8. ipv6 mld query-max-response-time seconds
9. ipv6 mld query-timeout seconds
10. ipv6 mld query-interval seconds
11. ipv6 mld limit number [except access-list]
12. end
### Detailed Steps

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Configures a limit on the number of MLD states resulting from MLD membership reports on a global basis. Membership reports sent after the configured limits have been exceeded are not entered in the MLD cache and traffic for the excess membership reports is not forwarded.</td>
</tr>
<tr>
<td>ipv6 mld state-limit number</td>
<td>• number—Maximum number of MLD states allowed on a router. The valid range is from 1 to 64000.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(config)# ipv6 mld state-limit 300</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Enables the Source Specific Multicast (SSM) mapping feature for groups in the configured SSM range.</td>
</tr>
<tr>
<td>ipv6 mld [vrf vrf-name] ssm-map enable</td>
<td>• vrf vrf-name—(Optional) Specifies a virtual routing and forwarding (VRF) configuration.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(config)# ipv6 mld ssm-map enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Specifies an interface type and number, and places the device in interface configuration mode.</td>
</tr>
<tr>
<td>interface type number</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(config)# interface GigabitEthernet 1/0/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>Allows the user to perform IPv6 multicast receiver access control.</td>
</tr>
<tr>
<td>ipv6 mld access-group access-list-name</td>
<td>• access-list-name—A standard IPv6 named access list that defines the multicast groups and sources to allow or deny.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(config-if)# ipv6 access-list acc-grp-1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>Statically forwards traffic for the multicast group onto a specified interface and cause the interface to behave as if a MLD joiner were present on the interface.</td>
</tr>
<tr>
<td>ipv6 mld static-group [group-address] [[include</td>
<td>exclude] {source-address</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>• include—(Optional) Enables include mode.</td>
</tr>
<tr>
<td>Device(config-if)# ipv6 mld static-group ff04::10 include 100::1</td>
<td>• exclude—(Optional) Enables exclude mode.</td>
</tr>
<tr>
<td></td>
<td>• source-address—Unicast source address to include or exclude.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| **Step 8** | ipv6 mld query-max-response-time *seconds*  
  Example:  
  Device(config-if)# ipv6 mld query-max-response-time 20  
  Configures the maximum response time advertised in MLD queries.  
  *seconds*—Maximum response time, in seconds, advertised in MLD queries. The default value is 10 seconds. |
| **Step 9** | ipv6 mld query-timeout *seconds*  
  Example:  
  Device(config-if)# ipv6 mld query-timeout 130  
  Configures the timeout value before the device takes over as the querier for the interface.  
  *seconds*—Number of seconds that the router waits after the previous querier has stopped querying and before it takes over as the querier. |
| **Step 10** | ipv6 mld query-interval *seconds*  
  Example:  
  Device(config-if)# ipv6 mld query-interval 60  
  Configures the frequency at which the Cisco IOS XE software sends MLD host-query messages.  
  *seconds*—Frequency, in seconds, at which to send MLD host-query messages. It can be a number from 0 to 65535. The default is 125 seconds.  
  **Caution** Changing this value may severely impact multicast forwarding. |
| **Step 11** | ipv6 mld limit *number* [except *access-list*]  
  Example:  
  Device(config-if)# ipv6 mld limit 100  
  Configures a limit on the number of MLD states resulting from MLD membership reports on a per-interface basis. Membership reports sent after the configured limits have been exceeded are not entered in the MLD cache, and traffic for the excess membership reports is not forwarded.  
  Per-interface and per-system limits operate independently of each other and can enforce different configured limits. A membership state is ignored if it exceeds either the per-interface limit or global limit.  
  If you do not configure the except *access-list* keyword and argument, all MLD states are counted toward the configured cache limit on an interface. Use the except *access-list* keyword and argument to exclude particular groups or channels from counting toward the MLD cache limit. An MLD membership report is counted against the per-interface limit if it is permitted by the extended access list specified by the except *access-list* keyword and argument. |
Disabling MLD Device-Side Processing

A user might only want specified interfaces to perform IPv6 multicast and therefore want to turn off MLD device-side processing on a specified interface. To disable MLD device-side processing, complete the following steps:

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `interface type number`
4. `no ipv6 mld router`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>interface type number</code></td>
<td>Specifies an interface type and number, and places the device in interface configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config)# interface GigabitEthernet 1/0/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> <code>no ipv6 mld router</code></td>
<td>Disables MLD device-side processing on a specified interface.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
</tbody>
</table>
Resetting the MLD Traffic Counters

To reset the MLD traffic counters, complete the following steps:

**SUMMARY STEPS**

1. enable
2. clear ipv6 mld [vrf vrf-name] traffic

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>Resets all MLD traffic counters.</td>
</tr>
<tr>
<td>clear ipv6 mld [vrf vrf-name] traffic</td>
<td>• vrf vrf-name—(Optional) Specifies a virtual routing and forwarding (VRF) configuration.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device# clear ipv6 mld traffic</td>
<td></td>
</tr>
</tbody>
</table>

Clearing the MLD Interface Counters

To clear the MLD interface counters, complete the following steps:

**SUMMARY STEPS**

1. enable
2. clear ipv6 mld [vrf vrf-name] counters interface-type

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>Clears the MLD interface counters.</td>
</tr>
<tr>
<td>clear ipv6 mld [vrf vrf-name] counters interface-type</td>
<td>• vrf vrf-name—(Optional) Specifies a virtual routing and forwarding (VRF) configuration.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
</tbody>
</table>
Clearing the MLD Groups

To clear MLD related information in the IPv6 multicast routing table, complete the following steps:

**SUMMARY STEPS**

1. **enable**
2. **configure terminal**
3. **clear ipv6 [icmp] mld groups {* | group-prefix | group [source]} [vrf {vrf-name | all}]**
4. **end**

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> clear ipv6 [icmp] mld groups {*</td>
<td>group-prefix</td>
</tr>
<tr>
<td><strong>Example:</strong> Device (config)# clear ipv6 mld groups *</td>
<td>• icmp—(Optional) Clears ICMP information.</td>
</tr>
<tr>
<td></td>
<td>• *—Specifies all routes.</td>
</tr>
<tr>
<td></td>
<td>• group-prefix—Group prefix.</td>
</tr>
<tr>
<td></td>
<td>• group—Group address.</td>
</tr>
<tr>
<td></td>
<td>• source—(Optional) Source (S, G) route.</td>
</tr>
<tr>
<td></td>
<td>• vrf—(Optional) Applies to a virtual routing and forwarding (VRF) instance.</td>
</tr>
<tr>
<td></td>
<td>• vrf-name—(Optional) VRF name. The name can be alphanumeric, case sensitive, or a maximum of 32 characters.</td>
</tr>
<tr>
<td></td>
<td>• all—Specifies all VRFs.</td>
</tr>
</tbody>
</table>
### Verifying IPv6 Multicast Listener Discovery Protocol

- Use the `show ipv6 mld groups [link-local] [group-name | group-address] [interface-type interface-number] [detail | explicit]` command to display the multicast groups that are directly connected to the device and that were learned through MLD:

```
Router# show ipv6 mld groups
```

MLD Connected Group Membership

<table>
<thead>
<tr>
<th>Group Address</th>
<th>Interface</th>
<th>Uptime</th>
<th>Expires</th>
</tr>
</thead>
<tbody>
<tr>
<td>FF08::1</td>
<td>Gi0/4/4</td>
<td>00:10:22</td>
<td>00:04:19</td>
</tr>
</tbody>
</table>

- Use the `show ipv6 mfib [vrf vrf-name] [all | linkscope | verbose | group-address-name | ipv6-prefix/prefix-length | source-address-name | interface | status | summary]` command display the forwarding entries and interfaces in the IPv6 Multicast Forwarding Information Base (MFIB).

The following example shows forwarding entries and interfaces in the MFIB specified with a group address of FF08::1::1:

```
Router# show ipv6 mfib ff08::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

Default

(*,FF08::1) Flags:  C HW
SW Forwarding:  0/0/0/0, Other: 264/264/0
HW Forwarding:  0/0/0/0, Other: 0/0/0
Tunnel Flags:  A NS
GigabitEthernet0/4/4 Flags:  F NS Pkts: 0/0
(2000::8,FF08::1) Flags:  HW
SW Forwarding:  1/0/132/0, Other: 0/0/0
HW Forwarding:  75852/1047/150/1226, Other: 0/0/0
GigabitEthernet0/4/2 Flags:  A
GigabitEthernet0/4/4 Flags:  F NS Pkts: 0/1

- Use the `show ipv6 mld interface [type number]` command to display multicast-related information about an interface.
The following is sample output from the `show ipv6 mld interface` command for Gigabit Ethernet interface 0/4/4:

```
Router# show ipv6 mld interface gigabitethernet 0/4/4
GigabitEthernet0/4/4 is up, line protocol is up
Internet address is FE80::D2C2:82FF:FE17:77C4/10
MLD is enabled on interface
Current MLD version is 2
MLD query interval is 125 seconds
MLD querier timeout is 255 seconds
MLD max query response time is 10 seconds
Last member query response interval is 1 seconds
MLD activity: 7 joins, 0 leaves
MLD querying router is FE80::D2C2:82FF:FE17:77C4 (this system)
```

• Use the `show ipv6 mld [vrf vrf-name] traffic` command to display the MLD traffic counters:

```
Router# show ipv6 mld traffic
MLD Traffic Counters
Elapsed time since counters cleared: 00:11:25

<table>
<thead>
<tr>
<th></th>
<th>Received</th>
<th>Sent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valid MLD Packets</td>
<td>672</td>
<td>85</td>
</tr>
<tr>
<td>Queries</td>
<td>7</td>
<td>20</td>
</tr>
<tr>
<td>Reports</td>
<td>665</td>
<td>65</td>
</tr>
<tr>
<td>Leaves</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mtrace packets</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Errors:
- Malformed Packets: 0
- Martian source: 2
- Non link-local source: 0
- Hop limit is not equal to 1: 0
```

• Use the `show ipv6 mroute [vrf vrf-name] [link-local | [group-name | group-address [source-address | source-name] ] ]` command to display the information in the PIM topology table:

```
Router# show ipv6 mroute ff08::1
Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group,
       C - Connected, L - Local, I - Received Source Specific Host Report,
       P - Pruned, R - RP-bit set, F - Register flag, T - SPT-bit set,
       J - Join SPT, Y - Joined MDT-data group,
       y - Sending to MDT-data group
       g - BGP signal originated, G - BGP Signal received,
       N - BGP Shared-Tree Prune received, n - BGP C-Mroute suppressed,
       q - BGP Src-Active originated, Q - BGP Src-Active received
       E - Extranet
Timers: Uptime/Expires
Interface state: Interface, State
(*, FF08::1), 00:11:18/never, RP 4000::1, flags: SCJ
  Incoming interface: Tunnel1
  RPF nbr: 4000::1
Immediate Outgoing interface list:
  GigabitEthernet0/4/4, Forward, 00:11:18/never
(2000::8, FF08::1), 00:02:07/00:01:22, flags: SFJT
```
Incoming interface: GigabitEthernet0/4/2
RPF nbr: 2000::8
Inherited Outgoing interface list:
  GigabitEthernet0/4/4, Forward, 00:11:18/never
Verifying IPv6 Multicast Listener Discovery Protocol
CHAPTER 10

Using MSDP to Interconnect Multiple PIM-SM Domains

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

- Finding Feature Information, on page 177
- , on page 178
- Information About Using MSDP to Interconnect Multiple PIM-SM Domains, on page 178
- How to Use MSDP to Interconnect Multiple PIM-SM Domains, on page 190
- Configuration Examples for Using MSDP to Interconnect Multiple PIM-SM Domains, on page 212
- Additional References, on page 214
- Feature Information for Multicast Source Discovery Protocol, on page 215

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.
Information About Using MSDP to Interconnect Multiple PIM-SM Domains

Benefits of Using MSDP to Interconnect Multiple PIM-SM Domains

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

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

Note

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

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

The figure illustrates MSDP operating between two MSDP peers. PIM uses MSDP as the standard mechanism to register a source with the RP of a domain.
When MSDP is implemented, the following sequence of events occurs:

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

---

**Note**

1. The SA message identifies the source address, the group that the source is sending to, and the address or the originator ID of the RP, if configured.

2. Each MSDP peer that receives the SA message floods the SA message to all of its peers downstream from the originator. In some cases (such as the case with the RPs in PIM-SM domains B and C in the figure), an RP may receive a copy of an SA message from more than one MSDP peer. To prevent looping, the RP consults the BGP next-hop database to determine the next hop toward the originator of the SA message. If both MBGP and unicast BGP are configured, MBGP is checked first, and then unicast BGP. That next-hop neighbor is the RPF-peer for the originator. SA messages that are received from the originator on any interface other than the interface to the RPF peer are dropped. The SA message flooding process, therefore, is referred to as peer-RPF flooding. Because of the peer-RPF flooding mechanism, BGP or MBGP must be running in conjunction with MSDP.
1. When an RP receives an SA message, it checks to see whether there are any members of the advertised groups in its domain by checking to see whether there are interfaces on the group’s (*, G) outgoing interface list. If there are no group members, the RP does nothing. If there are group members, the RP sends an (S, G) join toward the source. As a result, a branch of the interdomain source tree is constructed across autonomous system boundaries to the RP. As multicast packets arrive at the RP, they are then forwarded down its own shared tree to the group members in the RP’s domain. The members’ DRs then have the option of joining the rendezvous point tree (RPT) to the source using standard PIM-SM procedures.

2. The originating RP continues to send periodic SA messages for the (S, G) state every 60 seconds for as long as the source is sending packets to the group. When an RP receives an SA message, it caches the SA message. Suppose, for example, that an RP receives an SA message for (172.16.5.4, 228.1.2.3) from originating RP 10.5.4.3. The RP consults its mroute table and finds that there are no active members for group 228.1.2.3, so it passes the SA message to its peers downstream of 10.5.4.3. If a host in the domain then sends a join to the RP for group 228.1.2.3, the RP adds the interface toward the host to the outgoing interface list of its (*, 224.1.2.3) entry. Because the RP caches SA messages, the device will have an entry for (172.16.5.4, 228.1.2.3) and can join the source tree as soon as a host requests a join.

**Note**

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

### MSDP Message Types

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

**SA Messages**

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

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

**Note**

For more information about SA messages, see the SA Message Origination Receipt and Processing, on page 181 section.

### SA Request Messages

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

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

Keepalive Messages

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

SA Message Origination Receipt and Processing

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

SA Message Origination

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

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

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

SA Message Receipt

SA messages are only accepted from the MSDP RPF peer that is in the best path back toward the originator. The same SA message arriving from other MSDP peers must be ignored or SA loops can occur.

Deterministically selecting the MSDP RPF peer for an arriving SA message requires knowledge of the MSDP topology. However, MSDP does not distribute topology information in the form of routing updates. MSDP infers this information by using (M)BGP routing data as the best approximation of the MSDP topology for the SA RPF check mechanism. An MSDP topology, therefore, must follow the same general topology as the BGP peer topology. Besides a few exceptions (such as default MSDP peers and MSDP peers in MSDP mesh groups), MSDP peers, in general should also be (M)BGP peers.

How RPF Check Rules Are Applied to SA Messages

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

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

RPF checks are not performed in the following cases:
• If the sending MSDP peer is the only MSDP peer, which would be the case if only a single MSDP peer or a default MSDP peer is configured.
• If the sending MSDP peer is a member of a mesh group.
• If the sending MSDP peer address is the RP address contained in the SA message.

How the Software Determines the Rule to Apply to RPF Checks

The software uses the following logic to determine which RPF rule to apply to RPF checks:
• Find the (M)BGP neighbor that has the same IP address as the sending MSDP peer.
  • If the matching (M)BGP neighbor is an internal BGP (iBGP) peer, apply Rule 1.
  • If the matching (M)BGP neighbor is an external BGP (eBGP) peer, apply Rule 2.
  • If no match is found, apply Rule 3.

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

Rule 1 of RPF Checking of SA Messages in MSDP

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

1. The peer searches the BGP Multicast Routing Information Base (MRIB) for the best path to the RP that originated the SA message. If a path is not found in the MRIB, the peer then searches the Unicast Routing Information Base (URIB). If a path is still not found, the RPF check fails.

2. If the previous search succeeds (that is, the best path is found), the peer then determines the address of the BGP neighbor for this best path, which will be the address of the BGP neighbor that sent the peer the path in BGP update messages.

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

Note

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

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

Implications of Rule 1 of RPF Checking on MSDP

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

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

Rule 2 of RPF Checking of SA Messages in MSDP

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

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

Implications of Rule 2 of RPF Checking on MSDP

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

Rule 3 of RPF Checking of SA Messages in MSDP

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

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

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

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

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

1. Using the group address G of the (S, G) pair in the SA message, the peer locates the associated (*, G) entry in the mroute table. If the (*, G) entry is found and its outgoing interface list is not null, then there are active receivers in the PIM-SM domain for the source advertised in the SA message.

2. The MSDP peer then creates an (S, G) entry for the advertised source.

3. If the (S, G) entry did not already exist, the MSDP peer immediately triggers an (S, G) join toward the source in order to join the source tree.

4. The peer then floods the SA message to all other MSDP peers with the exception of:
   - The MSDP peer from which the SA message was received.
   - Any MSDP peers that are in the same MSDP mesh group as this device (if the peer is a member of a mesh group).

Note

SA messages are stored locally in the device’s SA cache.

MSDP Peers

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

MSDP MD5 Password Authentication

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

How MSDP MD5 Password Authentication Works

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

Benefits of MSDP MD5 Password Authentication

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

SA Message Limits

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

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

MSDP Keepalive and Hold-Time Intervals

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

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

Note

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

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

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

MSDP Connection-Retry Interval

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

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

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

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

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

Figure 25: Default MSDP Peer Scenario

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

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

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

Benefits of MSDP Mesh Groups

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

SA Origination Filters

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

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

- You can configure an RP to prevent the device from advertising local sources in SA messages. The device will still forward SA messages from other MSDP peers in the normal fashion; it will just not originate any SA messages for local sources.

- You can configure the device to only originate SA messages for local sources sending to specific groups that match (S, G) pairs defined in the extended access list. All other local sources will not be advertised in SA messages.

- You can configure the device to only originate SA messages for local sources sending to specific groups that match AS paths defined in an AS-path access list. All other local sources will not be advertised in SA messages.

- You can configure the device to only originate SA messages for local sources that match the criteria defined in the route map. All other local sources will not be advertised in SA messages.

- You configure an SA origination filter that includes an extended access list, an AS-path access list, and route map, or a combination thereof. In this case, all conditions must be true before any local sources are advertised in SA messages.
Use of Outgoing Filter Lists in MSDP

By default, an MSDP-enabled device forwards all SA messages it receives to all of its MSDP peers. However, you can prevent SA messages from being forwarded to MSDP peers by creating outgoing filter lists. Outgoing filter lists apply to all SA messages, whether locally originated or received from another MSDP peer, whereas SA origination filters apply only to locally originated SA messages. For more information about enabling a filter for MSDP SA messages originated by the local device, see the Controlling SA Messages Originated by an RP for Local Sources section.

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

• You can filter all outgoing SA messages forwarded to a specified MSDP peer by configuring the device to stop forwarding its SA messages to the MSDP peer.

• You can filter a subset of outgoing SA messages forwarded to a specified MSDP peer based on (S, G) pairs defined in an extended access list by configuring the device to only forward SA messages to the MSDP peer that match the (S, G) pairs permitted in an extended access list. The forwarding of all other SA messages to the MSDP peer will be stopped.

• You can filter a subset of outgoing SA messages forwarded to a specified MSDP peer based on match criteria defined in a route map by configuring the device to only forward SA messages that match the criteria defined in the route map. The forwarding of all other SA messages to the MSDP peer will be stopped.

• You can filter a subset of outgoing SA messages from a specified peer based on the announcing RP address contained in the SA message by configuring the device to filter outgoing SA messages based on their origin, even after an SA message has been transmitted across one or more MSDP peers. The forwarding of all other SA messages to the MSDP peer will be stopped.

• You can configure an outgoing filter list that includes an extended access list, a route map, and either an RP access list or an RP route map. In this case, all conditions must be true for the MSDP peer to forward the outgoing SA message.

Caution

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

Use of Incoming Filter Lists in MSDP

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

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

• You can filter all incoming SA messages from a specified MSDP peer by configuring the device to ignore all SA messages sent to it from the specified MSDP peer.

• You can filter a subset of incoming SA messages from a specified peer based on (S, G) pairs defined in an extended access list by configuring the device to only receive SA messages from the MSDP peer that
match the (S, G) pairs defined in the extended access list. All other incoming SA messages from the MSDP peer will be ignored.

- You can filter a subset of incoming SA request messages from a specified peer based on match criteria defined in a route map by configuring the device to only receive SA messages that match the criteria defined in the route map. All other incoming SA messages from the MSDP peer will be ignored.

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

- You can filter a subset of incoming SA messages from a specified peer based on the announcing RP address contained in the SA message by configuring the device to filter incoming SA messages based on their origin, even after the SA message may have already been transmitted across one or more MSDP peers.

- You can configure an incoming filter list that includes an extended access list, a route map, and either an RP access list or an RP route map. In this case, all conditions must be true for the MSDP peer to receive the incoming SA message.

---

**Caution**

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

---

**TTL Thresholds in MSDP**

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

In general, a TTL-threshold problem can be introduced by the encapsulation of a source’s initial multicast packet in an SA message. Because the multicast packet is encapsulated inside of the unicast SA message (whose TTL is 255), its TTL is not decremented as the SA message travels to the MSDP peer. Furthermore, the total number of hops that the SA message traverses can be drastically different than a normal multicast packet because multicast and unicast traffic may follow completely different paths to the MSDP peer and hence the remote PIM-SM domain. As a result, encapsulated packets can end up violating TTL thresholds.

The solution to this problem is to configure a TTL threshold that is associated with any multicast packet that is encapsulated in an SA message sent to a particular MSDP peer using the `ip multicast ttl-threshold` command. The `ip msdp ttl-threshold` command prevents any multicast packet whose TTL in the IP header is less than the TTL value specified for the `ttl-value` argument from being encapsulated in SA messages sent to that peer.

---

**SA Request Messages**

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

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

**Note**
In all current and supported software releases, caching of MSDP SA messages is mandatory and cannot be manually enabled or disabled. By default, when an MSDP peer is configured, the configured commands are automatically added to the running configuration.

**SA Request Filters**

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

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

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

**How to Use MSDP to Interconnect Multiple PIM-SM Domains**

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

**Configuring an MSDP Peer**

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

**Before you begin**

- IP multicast routing must be enabled and PIM-SM must be configured.

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

**SUMMARY STEPS**

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

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Device# configure terminal</td>
</tr>
<tr>
<td><strong>Step 3</strong> ip msdp peer {peer-name peer-address} [connect-source type number] [remote-as as-number]</td>
<td>Enables MSDP and configures an MSDP peer as specified by the DNS name or IP address.</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config)# ip msdp peer 192.168.1.2 connect-source loopback0</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>The device that is selected to be configured as an MSDP peer is also usually a BGP neighbor. If it is not, see the Configuring a Default MSDP Peer, on page 197 section or the Configuring an MSDP Mesh Group, on page 198 section.</td>
</tr>
<tr>
<td></td>
<td>• If you specify the connect-source keyword, the primary address of the specified local interface type and number values are used as the source IP address for the TCP connection. The connect-source keyword is recommended, especially for MSDP peers on a border that peer with a device inside of a remote domain.</td>
</tr>
<tr>
<td><strong>Step 4</strong> ip msdp description {peer-name peer-address} text</td>
<td>(Optional) Configures a description for a specified peer to make it easier to identify in a configuration or in show command output.</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config)# ip msdp description 192.168.1.2 router at customer a</td>
</tr>
<tr>
<td><strong>Step 5</strong> end</td>
<td>Exits global configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config)# end</td>
</tr>
</tbody>
</table>

Shutting Down an MSDP Peer

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

**Note**

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

**Before you begin**

MSDP is running and the MSDP peers must be configured.

**SUMMARY STEPS**

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

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><code>enable</code>&lt;br&gt;<code>Example:</code>&lt;br&gt;<code>&gt; enable</code></td>
<td>Enables privileged EXEC mode.&lt;br&gt;- Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><code>configure terminal</code>&lt;br&gt;<code>Example:</code>&lt;br&gt;<code># configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>`ip msdp shutdown {peer-name</td>
<td>peer-address}<code>&lt;br&gt;</code>Example:<code>&lt;br&gt;</code>(config)# ip msdp shutdown 192.168.1.3`</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Repeat Step 3 to shut down additional MSDP peers.</td>
<td>--</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><code>end</code>&lt;br&gt;<code>Example:</code>&lt;br&gt;<code>(config)# end</code></td>
<td>Exits global configuration mode and returns to privileged EXEC mode.</td>
</tr>
</tbody>
</table>
Configuring MSDP MD5 Password Authentication Between MSDP Peers

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

**SUMMARY STEPS**

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

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1**
  * `enable`
  * **Example:**
    
    Device> enable
| Enables privileged EXEC mode.  
  • Enter your password if prompted. |
| **Step 2**
  * `configure terminal`
  * **Example:**
    
    Device# configure terminal
| Enters global configuration mode. |
| **Step 3**
  * `ip msdp password peer {peer-name | peer-address} [encryption-type] string`
  * **Example:**
    
    Device(config)# ip msdp password peer 10.32.43.144 0 test
| Enables MD5 password encryption for a TCP connection between two MSDP peers.  
  **Note** MD5 authentication must be configured with the same password on both MSDP peers; otherwise, the connection between them will not be made.  
  • If you configure or change the password or key, which is used for MD5 authentication between two MSDP peers, the local device does not disconnect the existing session after you configure the password. You must manually disconnect the session to activate the new or changed password. |
| **Step 4**
  * `exit`
  * **Example:**
    
    Device(config)# exit
| Exits global configuration mode and returns to privileged EXEC mode. |
| **Step 5**
  * `show ip msdp peer [peer-address | peer-name]`
  * **Example:**
    
    Device# show ip msdp peer
| (Optional) Displays detailed information about MSDP peers.  
  **Note** Use this command to verify whether MD5 password authentication is enabled on an MSDP peer. |
Troubleshooting Tips

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

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

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

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

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

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

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

Note

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

### SUMMARY STEPS

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

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td>enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> ip msdp sa-limit {peer-address</td>
<td>peer-name} sa-limit</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config)# ip msdp sa-limit 192.168.10.1 100</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> Repeat Step 3 to configure SA limits for additional MSDP peers.</td>
<td>--</td>
</tr>
<tr>
<td><strong>Step 5</strong> exit</td>
<td>Exits global configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> show ip msdp count [as-number]</td>
<td>(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.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device# show ip msdp count</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> show ip msdp peer [peer-address</td>
<td>peer-name]</td>
</tr>
<tr>
<td>Example:</td>
<td>Note: The output of this command displays the number of SA messages received from MSDP peers that are stored in the cache.</td>
</tr>
<tr>
<td>Device# show ip msdp peer</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> show ip msdp summary</td>
<td>(Optional) Displays MSDP peer status.</td>
</tr>
<tr>
<td>Example:</td>
<td>Note: The output of this command displays a per-peer “SA Count” field that displays the number of SAs stored in the cache.</td>
</tr>
<tr>
<td>Device# show ip msdp summary</td>
<td></td>
</tr>
</tbody>
</table>

**Adjusting the MSDP Keepalive and Hold-Time Intervals**

Perform this optional task to adjust the interval at which an MSDP peer will send keepalive messages and the interval at which the MSDP peer will wait for keepalive messages from other peers before declaring them down. By default, it may take as long as 75 seconds for an MSDP peer to detect that a peering session with another MSDP peer has gone down. In network environments with redundant MSDP peers, decreasing the hold-time interval can expedite the reconvergence time of MSDP peers in the event that an MSDP peer fails.
We recommend that you do not change the command defaults for the `ip msdp keepalive` command, because the command defaults are in accordance with RFC 3618, `Multicast Source Discovery Protocol`. If your network environment requires that you modify the defaults, you must configure the same time values for the `keepalive-interval` and `hold-time-interval` arguments on both ends of the MSDP peering session.

**SUMMARY STEPS**

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

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> ip msdp keepalive {peer-address</td>
<td>peer-name} keepalive-interval hold-time-interval</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config)# ip msdp keepalive 10.1.1.3 40 55</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> Repeat Step 3 to adjust the keepalive message interval for additional MSDP peers.</td>
<td>--</td>
</tr>
<tr>
<td><strong>Step 5</strong> exit</td>
<td>Exits global configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config)# exit</td>
<td></td>
</tr>
</tbody>
</table>

**Adjusting the MSDP Connection-Retry Interval**

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

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

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> ip msdp timer</td>
<td>Configures the interval at which MSDP peers will wait after peering sessions are reset before attempting to reestablish the peering sessions.</td>
</tr>
<tr>
<td>connection-retry-interval</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device# ip msdp timer 45</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> exit</td>
<td>Exits global configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config)# exit</td>
<td></td>
</tr>
</tbody>
</table>

Configuring a Default MSDP Peer

Perform this optional task to configure a default MSDP peer.

Before you begin

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

SUMMARY STEPS

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>enable</td>
<td></td>
</tr>
</tbody>
</table>

**Step 2**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>configure terminal</td>
<td></td>
</tr>
</tbody>
</table>

**Step 3**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Configures a default peer from which to accept all MSDP SA messages</td>
</tr>
<tr>
<td>ip msdp default-peer {peer-address</td>
<td>peer-name} [prefix-list list]</td>
</tr>
</tbody>
</table>

**Step 4**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Exits global configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>exit</td>
<td></td>
</tr>
</tbody>
</table>

Configuring an MSDP Mesh Group

Perform this optional task to configure an MSDP mesh group.

Note

You can configure multiple mesh groups per device.

SUMMARY STEPS

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

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>enable</td>
<td></td>
</tr>
</tbody>
</table>

**Step 2**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>configure terminal</td>
<td></td>
</tr>
</tbody>
</table>

**Step 3**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Configures a default peer from which to accept all MSDP SA messages</td>
</tr>
<tr>
<td>ip msdp default-peer {peer-address</td>
<td>peer-name} [prefix-list list]</td>
</tr>
</tbody>
</table>

**Step 4**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Exits global configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>exit</td>
<td></td>
</tr>
</tbody>
</table>
### Purpose

#### Command or Action

<table>
<thead>
<tr>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example:</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>configure terminal</code></td>
</tr>
</tbody>
</table>

| Configures an MSDP mesh group and indicates that an MSDP peer belongs to that mesh group. |

<table>
<thead>
<tr>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>All MSDP peers on a device that participate in a mesh group must be fully meshed with all other MSDP peers in the group. Each MSDP peer on each device must be configured as a peer using the <code>ip msdp peer</code> command and also as a member of the mesh group using the <code>ip msdp mesh-group</code> command.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>`ip msdp mesh-group mesh-name {peer-address</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example:</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>(config)# ip msdp mesh-group peermesh</code></td>
</tr>
</tbody>
</table>

| Repeat Step 3 to add MSDP peers as members of the mesh group. |

| Exit |

<table>
<thead>
<tr>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exits global configuration mode and returns to privileged EXEC mode.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example:</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>(config)# exit</code></td>
</tr>
</tbody>
</table>

## Controlling SA Messages Originated by an RP for Local Sources

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

### Note

For best practice information related to configuring MSDP SA message filters, see the [Multicast Source Discovery Protocol SA Filter Recommendations](#) tech note.

### SUMMARY STEPS

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

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>enable</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enables privileged EXEC mode.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example:</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>enable</code></td>
</tr>
</tbody>
</table>

| • Enter your password if prompted. |
Purpose

Command or Action

| Device> enable |

**Step 2**

**configure terminal**

**Example:**

Device# configure terminal

**Step 3**

**ip msdp redistribute [list access-list] [asn as-access-list] [route-map map-name]**

**Example:**

Device(config)# ip msdp redistribute route-map customer-sources

**Step 4**

**exit**

**Example:**

Device(config)# exit

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

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

**Note**

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

#### SUMMARY STEPS

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

#### DETAILED STEPS

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

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

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

**Note**

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

**SUMMARY STEPS**

1. enable
2. configure terminal
3. ip msdp sa-filter in \{peer-address | peer-name\} [list access-list] [route-map map-name] [rp-list access-list | rp-route-map map-name]
4. Repeat Step 3 to configure incoming filter lists for additional MSDP peers.
5. exit
**Using TTL Thresholds to Limit the Multicast Data Sent in SA Messages**

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

**SUMMARY STEPS**

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

**DETAILED STEPS**

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

Command or Action | Purpose |
--- | --- |
**Step 2** | Enters global configuration mode. |
configure terminal |  |
Example: |  |
Device# configure terminal |  |

**Step 3** | Sets a TTL value for MSDP messages originated by the local device. |
ip msdp ttl-threshold \{peer-address | peer-name\} ttl-value |  |
Example: Device(config)# ip msdp ttl-threshold 192.168.1.5 |  |
Example: |  |
Device(config)# ip msdp ttl-threshold 192.168.1.5 8 |  |
  • By default, multicast data packets in SA messages are sent to an MSDP peer, provided the TTL value of the packet is greater than 0, which is standard TTL behavior. |

**Step 4** | Exits global configuration mode and returns to privileged EXEC mode. |
exit |  |
Example: |  |
Device(config)# exit |  |

### Requesting Source Information from MSDP Peers

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

**Note**

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

### SUMMARY STEPS

1. enable
2. configure terminal
3. ip msdp sa-request \{peer-address | peer-name\}
4. Repeat Step 3 to specify that the device send SA request messages to additional MSDP caching peers.
5. exit

### DETAILED STEPS

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

**Step 2** | Enters global configuration mode. |
configure terminal |  |
Example: |  |
Controlling the Response to Outgoing SA Request Messages from MSDP Peers Using SA Request Filters

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

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `ip msdp filter-sa-request {peer-address | peer-name} [list access-list]`
4. Repeat Step 3 to configure SA request filters for additional MSDP peers.
5. `exit`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Device&gt; enable</code></td>
<td></td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Device# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td>`ip msdp filter-sa-request {peer-address</td>
<td>peer-name} [list access-list]`</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
</tbody>
</table>
Including a Bordering PIM Dense Mode Region in MSDP

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

You can have a device that borders a PIM-SM region and a PIM-DM region. By default, sources in the PIM-DM domain are not included in MSDP. You can configure this border device to send SA messages for sources active in the PIM-DM domain. If you do so, it is very important to also configure the `ip msdp redistribute` command to control what local sources from the PIM-DM domain are advertised. Not configuring this command can result in the (S, G) state remaining long after a source in the PIM-DM domain has stopped sending. For configuration information, see the Controlling SA Messages Originated by an RP for Local Sources, on page 199 section.

**SUMMARY STEPS**

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

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1**
  enable                        | Enables privileged EXEC mode.                                           |
  Example:
  ```
  Device> enable
  ```
| **Step 2**
  configure terminal             | Enters global configuration mode.                                       |
  Example:
  ```
  Device# configure terminal
  ```
Configuring an Originating Address Other Than the RP Address

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

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

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

Before you begin

MSDP is enabled and the MSDP peers are configured. For more information about configuring MSDP peers, see the Configuring an MSDP Peer, on page 190 section.

SUMMARY STEPS

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

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Device&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>

Example:

Device(config)# ip msdp border sa-address type number
Device(config)# exit

Example:

Device(config)# exit

Example:
**Monitoring MSDP**

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

### SUMMARY STEPS

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

### DETAILED STEPS

#### Step 1  enable

Example:

```
Device# enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

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

Use this command to debug MSDP activity.

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

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

Example:

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

Step 3 debug ip msdp resets

Use this command to debug MSDP peer reset reasons.

Example:

Device# debug ip msdp resets

Step 4 show ip msdp count [as-number]

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

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

Example:

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

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

Use this command to display detailed information about MSDP peers.

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

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

Example:

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

Step 6 show ip msdp sa-cache [group-address | source-address | group-name | source-name] [as-number]
Use this command to display the (S, G) state learned from MSDP peers.
The following is sample output from the show ip msdp sa-cache command:

Example:

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

Step 7 show ip msdp summary
Use this command to display MSDP peer status.
The following is sample output from the show ip msdp summary command:

Example:

Device# show ip msdp summary
MSDP Peer Status Summary
Peer Address  AS  State  Uptime/ Reset SA  Peer Name
        Downtime  Count  Count
192.168.4.4   4  Up  00:08:05   0  8  7

Clearing MSDP Connections Statistics and SA Cache Entries

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

SUMMARY STEPS

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

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> clear ip msdp peer [peer-address</td>
<td>peer-name]</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# clear ip msdp peer</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> clear ip msdp statistics [peer-address</td>
<td>peer-name]</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# clear ip msdp statistics</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> clear ip msdp sa-cache [group-address]</td>
<td>Clears SA cache entries.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# clear ip msdp sa-cache</td>
<td>• If the <code>clear ip msdp sa-cache</code> is specified with the optional <code>group-address</code> argument or <code>source-address</code> argument, all SA cache entries are cleared.</td>
</tr>
<tr>
<td></td>
<td>• Use the optional <code>group-address</code> argument to clear all SA cache entries associated with a specific group.</td>
</tr>
</tbody>
</table>

### Enabling SNMP Monitoring of MSDP

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

**Before you begin**

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

**Note**

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

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

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** `enable` | Enables privileged EXEC mode.  
* Enter your password if prompted. |
| **Example:**  
Device> enable | |
| **Step 2** `snmp-server enable traps msdp` | Enables the sending of MSDP notifications for use with SNMP. |
| **Example:**  
Device# snmp-server enable traps msdp | |
| **Step 3** `snmp-server host host [traps | informs] [version {1 | 2c | 3 [auth | priv | noauth]}] community-string [udp-port port-number] msdp` | Specifies the recipient (host) for MSDP traps or informs. |
| **Example:**  
Device# snmp-server host examplehost msdp | |
| **Step 4** `exit` | Exits global configuration mode and returns to privileged EXEC mode. |
| **Example:**  
Device(config)# exit | |

### Troubleshooting Tips

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

Example: Configuring an MSDP Peer

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

**Device A**

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

**Device B**

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

**Device C**

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

Example: Configuring MSDP MD5 Password Authentication

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

**Device A**

```yaml
! ip msdp peer 10.3.32.154
ip msdp password peer 10.3.32.154 0 test
```
Example: Configuring a Default MSDP Peer

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

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

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

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

If you specify a prefix list, the peer will be a default peer only for the prefixes in the list. You can have multiple active default peers when you have a prefix list associated with each. When you do not have any prefix lists, you can configure multiple default peers, but only the first one is the active default peer as long as the device has connectivity to this peer and the peer is alive. If the first configured peer goes down or the connectivity to this peer goes down, the second configured peer becomes the active default, and so on.
The following example shows a partial configuration of Device A and Device C in the figure. Each of these ISPs may have more than one customer using default peering, like the customer in the figure. In that case, they may have similar configurations. That is, they will only accept SAs from a default peer if the SA is permitted by the corresponding prefix list.

**Device A Configuration**

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

**Device C Configuration**

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

**Example: Configuring MSDP Mesh Groups**

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

**Device A Configuration**

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

**Device B Configuration**

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

**Device C Configuration**

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

**Additional References**

**Related Documents**

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv6 addressing and connectivity</td>
<td><em>IPv6 Configuration Guide</em></td>
</tr>
</tbody>
</table>
Feature Information for Multicast Source Discovery Protocol

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

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.
Table 4: Feature Information for Multicast Source Discovery Process

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSDP MD5 Password Authentication</td>
<td>12.4(2)T, 12.2(33)SXH, 12.2(33)SRE, 15.0(1)S, 15.1(1)SG, Cisco IOS XE Release 3.3SG</td>
<td>The MSDP MD5 password authentication feature is an enhancement to support MD5 signature protection on a TCP connection between two MSDP peers. This feature provides added security by protecting MSDP against the threat of spoofed TCP segments being introduced into the TCP connection stream. The following commands were introduced or modified: <code>ip msdp password peer</code>, <code>show ip msdp peer</code>.</td>
</tr>
</tbody>
</table>
The technology is not applicable for the Cisco ASR 900 RSP3 Module.

The MPLS Point-to-Multipoint Traffic Engineering feature enables you to forward Multiprotocol Label Switching (MPLS) traffic from one source to multiple destinations. Cisco nonstop forwarding (NSF) and stateful switchover (SSO) (NSF/SSO) provides for minimal disruption of Point-to-Multipoint (P2MP) Traffic Engineering (TE) tunnel traffic if a Route Processor has a catastrophic failure. Traffic loss varies by platform.

For more information on configuring NSF/SSO with this feature, see NSF/SSO—MPLS TE and RSVP Graceful Restart.

- Information About MPLS Point-to-Multipoint Traffic Engineering, on page 217
- How to Configure MPLS Point-to-Multipoint Traffic Engineering, on page 227
- Configuration Examples for MPLS Point-to-Multipoint Traffic Engineering, on page 237

Information About MPLS Point-to-Multipoint Traffic Engineering

MPLS Point-to-Multipoint Traffic Engineering Overview

A P2MP TE network contains the following elements, which are shown in the figure below:

- The headend router, also called the source or ingress router, is where the label switched path (LSP) is initiated. The headend router can also be a branch point, which means the router performs packet replication and the sub-LSPs split into different directions.

- The midpoint router is where the sub-LSP signaling is processed. The midpoint router can be a branch point.

- The tailend router, also called the destination, egress, or leaf-node router, is where sub-LSP signaling ends.

- A bud router is a midpoint and tailend router at the same time.

- A P2MP tunnel consists of one or more sub-LSPs. All sub-LSPs belonging to the same P2MP tunnel employ the same constraints, protection policies, and so on, which are configured at the headend router.
P2MP TE tunnels build on the features that exist in basic point-to-point TE tunnels. The P2MP TE tunnels have the following characteristics:

- There is one source (headend) but more than one destination (tailend).
- They are unidirectional.
- They are explicitly routed.
- Multiple sub-LSPs connect the headend router to various tailend routers.

The figure below shows a P2MP TE tunnel that has three destinations.

- PE1 is the headend router.
- P01 is a branch point router, where packet replication occurs.
- PE2, PE3, and PE4 are tailend routers, where the sub-LSP ends.

Between the PE and CE routers, PIM is enabled to exchange multicast routing information with the directly connected customer edge (CE) routers. PIM is not enabled across the P2MP TE tunnel.

**Database of Sub-LSP Failure Errors**

If any sub-LSP, whether P2MP or P2P, fails to recover after an SSO switchover, the failure is noted in an error database for troubleshooting. You can use the `show ip rsvp high database lsp` command to display the error database entries.
How P2MP TE Sub-LSPs Are Signaled

RSVP TE extensions defined in RFC 4875 allow multiple sub-LSPs to be signaled from the headend router. A P2MP TE tunnel consists of multiple sub-LSPs that connect the headend router to various tailend routers.

The headend router sends one RSVP path message to each destination. The tailend router replies with a RESV message. The Label Forwarding Information Base (LFIB) is populated using the RSVP labels allocated by the RESV messages.

The P2MP TE feature does not support signaling of multiple sub-LSPs in the same Path/Resv message. If multiple sub-LSPs occur in the same message, the router sends a PathErr Unknown Objects message, and the Path/Resv message with multiple sub-LSPs is not forwarded.

The tailend routers allocate unreserved labels, which are greater than 15 and do not include implicit or explicit null labels. Using unreserved labels allows IP multicast to perform a Reverse Path Forwarding (RPF) check on the tailend router. Because a sub-LSP tailend router cannot be represented as a regular interface, a special LSP virtual interface (VIF) is automatically created. The LSP VIF represents the originating interface for all IP multicast traffic originating from the P2MP TE tailend router.

The figure below shows the LSP signaling process.
How P2MP TE Traffic Is Forwarded

At the headend of the traffic engineering tunnel, through a static Internet Group Management Protocol (IGMP) group-to-tunnel mapping, IP multicast traffic is encapsulated with a unique MPLS label, which is associated with the P2MP TE tunnel. The multicast traffic is label switched in the P2MP tree and replicated at branch and bud nodes along the P2MP tree. When the labeled packet reaches the tailend (a PE router), the MPLS label is removed and forwarded to the IP multicast tree towards the end point. This process is shown in the figure below.

---

**Note**

The P2MP TE feature does not support penultimate-hop popping. Therefore, the egress router must allocate an explicit null or non-null label.
When sub-LSPs share a common router (branch point) and use the same ingress interface of the router, the same MPLS label is used for forwarding. The multicast state is built by reusing the MPLS labels at the branch points, as shown in the figure below, where MPLS label 17 is shared by two sub-LSPs that both use router C.

**Computing the IGP Path Using Dynamic Paths or Explicit Paths**

You can either specify explicit paths or allow paths to be created dynamically. You can also specify bandwidth parameters, which are flooded throughout the MPLS network through existing RSVP-TE extensions to Open Shortest Path First (OSPF) and Integrated Intermediate System-to-Intermediate System (IS-IS).

The MPLS core network uses RSVP to enable end-to-end IP multicast connectivity. The tail end router and the end point router use PIM to exchange multicast routing information with directly connected CE routers. PIM is not configured in the MPLS core.

P2MP TE tunnels can co-exist with regular P2P TE tunnels. Existing path calculation and bandwidth preemption rules apply in this case.
You create IGP paths by enabling dynamic path computation, configuring explicit paths through CLI commands, or using both methods in your P2MP TE network.

- Dynamic paths are created using Constrained Shortest Path First (CSPF) to determine the best path to a destination. CSPF uses path constraints, such as bandwidth, affinities, priorities, and so on, as part of the computation.
- Explicit paths allows you to manually specify the path a sub-LSP uses from the headend router to the tailend router. You configure static paths on the headend router.

Remerge Events

When explicit paths are configured with a limited number of equal cost links or paths, two sub-LSPs might connect at a midpoint router through different ingress interfaces, but use the same egress interface. This is called a remerge event, which can cause duplicate MPLS packets. If a router detects a remerge event, it sends a PathErr Routing Problem: Remerge Detected message toward the headend router and the sub-LSPs are not established. With dynamic paths, the router signals a path that avoids a remerge situation.

Crossover Events

With a P2MP tunnel, two sibling sub-LSPs (sub-LSPs that share the same link and label) are said to “cross over” when they have different incoming interfaces and different outgoing interfaces on the same intersecting node. The sibling sub-LSPs neither share input label nor output bandwidth. Avoid configuring crossover LSPs, because they waste bandwidth. However, the duplication of sub-LSPs does not result in an error.

Benefits of MPLS Point-to-Multipoint Traffic Engineering

The P2MP TE feature provides the following benefits:

- You can configure signaling attributes, such as affinities, administrative metrics, FRR protection, and bandwidth constraints, when you set up P2MP TE sub-LSPs.
- P2MP TE provides a single point of traffic control. You specify all the signaling and path parameters at the headend router.
- You can configure explicit paths to optimize traffic distribution.
- You can enable FRR link protection for P2MP TE sub-LSPs.
- Protocol Independent Multicast (PIM) is not needed in the MPLS core. Only the non-MPLS interfaces on the tailend routers need to be configured with PIM.

MPLS Point-to-Multipoint Traffic Engineering—Re-optimizing Traffic

A P2MP TE tunnel is operational (up) when the first sub-LSP has been successfully signaled. The P2MP TE tunnel is not operational (down) when all sub-LSPs are down. Certain events can trigger a tunnel re-optimization:

- One of the sub-LSPs is fast-rerouted to a backup tunnel (for dynamic LSPs).
- A link is operational. (if the command `mpls traffic-eng reoptimize events link-up` is configured).
- A periodic schedule optimization occurs through the `mpls traffic-eng reoptimize timers frequency` command.
• The network administrator forces a tunnel optimization through the `mpls traffic-eng reoptimize` command.

• A FRR protected interface becomes operational.

• A non-FRR LSP detects a remerge situation.

When a P2MP tunnel is reoptimized, a new LSP is signaled and traffic is moved to the new LSP.

To determine if a tunnel should be reoptimized, the router considers the following criteria:

• The router compares the number of reachable destinations between the new tree and current tree. If the new tree contains more reachable destinations than the current tree, the router performs a reoptimization. If the new tree contains fewer reachable destinations than the current tree, then the router keeps the current tree.

• The router verifies that the same set of reachable destinations in the current tree are also in the new tree. If the new tree does not contain the same destinations, the router keeps the current tree.

• The router compares the number of destinations in the new tree with the number of destinations in the old tree. If the number of destinations in the new tree is greater than the number of destinations in the current tree, the router switches to the new tree. This guarantees that the new tree will contain all of the existing destinations and more.

• The router compares the metric between the current and new tree to ensure the new tree and current tree contain the same set of reachable destinations.

• The router compares the administrative weights of the old tree and the new tree. The router switches to the new tree if the cumulative administrative weight is lower. This step applies as a tie breaker if all the other conditions are the same.

P2MP TE uses make-before-break reoptimization, which uses the following reoptimization process:

• The new LSP is signaled.

• The headend router initiates a timer to ensure sufficient time elapses before traffic moves from the current LSP to the new LSP.

• Traffic is redirected from the current LSP to the new LSP.

• The timer is started for the purpose of tearing down the old sub-LSPs.

**P2P TE Tunnels Coexist with P2MP TE Tunnels**

Both P2P and P2MP TE tunnels share the following characteristics:

• Tunnel bandwidth is configured the same way in both P2P and P2MP tunnels. In P2MP TE tunnels, any bandwidth parameters you configure are applied to all the destination routers. That is, the bandwidth parameters apply to all sub-LSPs. Both P2P and P2MP TE tunnels use the same IGP extension to flood link bandwidth information throughout the network.

• Tunnel setup and hold priorities, attributes flags, affinity and mask, and administrative weight parameters are configured the same way for P2P and P2MP TE tunnels. P2MP TE tunnel parameters apply to all sub-LSPs.

• FRR-enabled P2MP sub-LSPs coexist with FRR-enabled P2P LSPs in a network. For P2P TE, node, link, and bandwidth protection is supported. For P2MP TE, only link protection is supported.
• The method of computing the path dynamically through CSPF is the same for P2P and P2MP TE.

• Auto-tunnel backup behaves slightly different with P2P and P2MP tunnels. With P2P tunnels, auto-tunnel backup creates two backup tunnels: one for the node protection and one for the link protection. The node protection backup is preferred for P2P LSP protection. With P2MP tunnels, auto-tunnel backup creates one backup tunnel, which is the link protection. Only the link protection backup can be used for P2MP sub-LSPs. The P2P and P2MP tunnels can coexist and be protected.

Note
If P2MP sub-LSPs are signaled from R1->R2->R3 and a P2P tunnel is signaled from R3->R2->R1, then issue the `mpls traffic-eng multicast-intact` command on R3 in IGP configuration mode under router OSPF or IS-IS to ensure to accommodate multicast traffic for R3's sub-LSPs.

Using FRR to Protect P2MP TE Links

FRR applies to P2P LSPs and P2MP sub-LSPs in the same manner. No new protocol extensions are needed to support P2MP.

Note
For P2MP TE FRR protection, issue the `ip routing protocol purge interface` command on every penultimate hop router. Otherwise, the router can lose up to 6 seconds worth of traffic during a FRR cutover event.

FRR minimizes interruptions in traffic delivery as a result of link failure. FRR temporarily fast switches LSP traffic to a backup path around a network failure until the headend router signals a new end-to-end LSP.

FRR-enabled P2MP sub-LSPs coexist with FRR-enabled P2P LSPs in a network. For P2MP TE, only link protection is supported. For P2P TE, node, link, and bandwidth protection are supported.

You can configure P2P explicit backup tunnels on point of local repair (PLR) nodes for link protection of P2MP sub-LSPs, similar to LSPs for P2P TE tunnels. You can also enable automatic creation of backup tunnels using the Auto-tunnel Backup feature for P2P TE tunnels. All sibling sub-LSPs that share the same outgoing link are protected by the same backup tunnel. All cousin sub-LSPs that share the same outgoing link can be protected by multiple P2P backup tunnels.

Link protection for a P2MP TE tunnel is illustrated in the figure below, which shows PE1 as the tunnel headend router and PE2, PE3, and PE4 as tunnel tailend routers. The following sub-LSPs are signaled from PE1 in the network:

- From PE1 to PE2, the sub-LSP travels the following path: PE1 -> P01 -> P02 -> PE2
- From PE1 to PE3, the sub-LSP travels the following path: PE1 -> P01 -> P03 -> PE3
- From PE1 to PE4, the sub-LSP travels the following path: PE1 -> P01 -> P04 -> PE4

Node P01 is a branch node that does packet replication in the MPLS forwarding plane; ingress traffic originating from PE1 will be replicated towards routers P02, P03, and P04.
To protect the three sub-LSPs, separate point-to-point backup tunnels are signaled.

**Note**
Backup tunnels can be created only for links that have an alternative network path.

In this example, router P01 is the Point of Local Repair (PLR) and routers P02, P03, and P04 are Merge Points (MPs).

If a link failure occurs between routers P01 and P04, the following events are triggered:

1. Router P01 switches traffic destined to PE4 to the backup tunnel associated with P04.
2. Router P01 sends RSVP path error messages upstream to the P2MP TE headend router PE1. At the same time, P01 and P04 send IGP updates (link state advertisements (LSAs)) to all adjacent IGP neighbors, indicating that the interfaces associated with links P01 through P04 are down.
3. Upon receiving RSVP path error messages and IGP LSA updates, the headend router triggers a P2MP TE tunnel reoptimization and signals a new sub-LSP. (This occurs if you have specified dynamic path creation.)

**Note**
If only one sub-LSP becomes active, it remains down until all the sub-LSPs become active.
FRR Failure Detection Mechanisms

To detect link failures in a P2MP TE network, you can use native link and interface failure detection mechanisms, such as bidirectional forwarding detection (BFD), and RSVP hellos.

Bidirectional Forwarding Detection

The MPLS Traffic Engineering: BFD-triggered FRR feature allows you to obtain link by using the Bidirectional Forwarding Detection (BFD) protocol to provide fast forwarding path failure detection times for all media types, encapsulations, topologies, and routing protocols. In addition to fast forwarding path failure detection, BFD provides a consistent failure detection method for network administrators. For more information, see MPLS Traffic Engineering: BFD-triggered Fast Reroute (FRR).

RSVP Hellos

You can configure RSVP hellos on interfaces that do not provide FRR cutover notification during a link failure. The behavior for RSVP hellos is similar for both P2MP TE and P2P TE. For every sub-LSP that has a backup tunnel and has RSVP hellos enabled on its output interface, an RSVP hello instance is created to the neighbor, and the sub-LSP is added to the neighbor's FRR tree in the hello database.

Hello instances between an output interface and neighbor address are shared by fast reroutable P2MP sub-LSPs and P2P LSPs. When a hello session to a neighbor is declared down, all P2P LSPs and P2MP sub-LSPs that are protected by a backup LSP or sub-LSP are switched to their respective backups in the control and data planes.

RSVP hello sessions can also be used to inform the P2MP headend router of failures along a sub-LSP’s path before the RSVP state for the sub-LSP times out, which leads to faster reoptimization. If a sub-LSP cannot select a backup tunnel but has RSVP hellos enabled on its output interface, it looks for a hello instance to its neighbor. If none exists, a hello state time (HST) hello instance is created. If the neighbor goes down, that sub-LSP is torn down. For more information, see MPLS Traffic Engineering (TE) - Fast Reroute (FRR) Link and Node Protection.

Bandwidth Preemption for P2MP TE

Bandwidth Admission Control and preemption mechanisms for P2MP TE sub-LSPs are the same as for LSPs associated with P2P TE tunnels. Any link affinities or constraints defined for the P2MP TE tunnel will be taken into account. The bandwidth signaled for the sub-LSP is removed from the appropriate pool at the appropriate priority, and if needed, lower priority sub-LSPs are preempted with a higher priority sub-LSP.

A P2MP tunnel can be configured to use sub-pool or global-pool bandwidth. When bandwidth is configured, all sub-LSPs of the P2MP tunnel are signaled with the same bandwidth amount and type. If the bandwidth amount or type of a P2MP tunnel is changed, the P2MP tunnel ingress always signals a new set of sub-LSPs (a new P2MP LSP) with the new bandwidth amount and type.

Preemption procedures do not take into account the tunnel type. The same priority rules apply to P2P LSPs and P2MP sub-LSPs. A sub-LSP with a higher setup priority preempts a (sub-)LSP with a lower hold priority, regardless of tunnel type. Thus, a P2MP sub-LSP may preempt a P2P LSP, and vice versa. The determination of which LSPs get preempted is based on hold priority.

You can configure a P2MP TE tunnel to use subpool or global-pool bandwidth. All sub-LSPs associated with the P2MP TE tunnel are signaled with the same bandwidth amount and type. If the bandwidth amount or type is changed, the P2MP tunnel headend router signals a new set of sub-LSPs with the new bandwidth parameters.
Bandwidth sharing is similar for P2MP TE sub-LSPs and P2P TE LSPs. When adding a new sub-LSP, the P2MP-TE headend router determines whether it should share bandwidth with the other sub-LSPs. Two sub-LSPs can share bandwidth as long as they are a “Transit Pair,” meaning the sub-LSPs share the output interface, next-hop and output label.

LSPs and sub-LSPs cannot share bandwidth if they use different bandwidth pools. A change in bandwidth requires reoptimizing P2P or P2MP TE tunnels, which may result in double-counting bandwidth on common links.

Using FRR with Bandwidth Protection has the following requirements:

• A backup tunnel is required to maintain the service level agreement while the new sub-LSP is created.
• The PLR router selects the backup tunnel only if the tunnel has enough bandwidth capacity.
• The backup tunnel might not signal bandwidth.
• The PLR router decides the best backup path to protect the primary path, based on backup bandwidth and class type.

How to Configure MPLS Point-to-Multipoint Traffic Engineering

Configuring the Headend Routers

The following steps explain how to configure the headend routers for multicast and MPLS point-to-multipoint traffic engineering. As part of the configuration, you specify the tailend routers. You can also specify explicit paths that the tunnel should use or request that the paths be dynamically created or have a combination of dynamic and explicit paths.

Because the configuration of the P2MP TE tunnels is done at the headend router, this feature works best in situations where the destinations do not change often. The P2MP feature does not support dynamic grafting and pruning of sub-LSPs.

SUMMARY STEPS

1. enable
2. configure terminal
3. mpls traffic-eng tunnels
4. ip multicast-routing [vrfvrf-name] [distributed]
5. interface tunnel number
6. tunnel mode mpls traffic-eng point-to-multipoint
7. tunnel destination list mpls traffic-eng {identifierdest-list-id|namedest-list-name}
8. ip igmp static-group {* | group-address [source{source-address| ssm-map}] | class-mapclass-map-name}
9. ip pim {dense-mode [proxy-register {listaccess-list | route-mapmap-name}] | passive | sparse-mode | sparse-dense-mode}
10. exit
11. mpls traffic-eng destination list {namedest-list-name | identifierdest-list-id}
12. ip ip-address path-option id {dynamic | explicit {namename | identifierid}}
13. exit
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
  Example:  
  ```
  Router> enable
  ``` |
| **Step 2** configure terminal | Enters global configuration mode.  
  Example:  
  ```
  Router# configure terminal
  ``` |
| **Step 3** mpls traffic-eng tunnels | Globally enables MPLS Traffic Engineering.  
  Example:  
  ```
  Router(config)# mpls traffic-eng tunnels
  ``` |
| **Step 4** ip multicast-routing [vrf vrf-name] [distributed] | Globally enables IP multicast routing.  
  Example:  
  ```
  Router(config)# ip multicast-routing distributed
  ``` |
| **Step 5** interface tunnel number | Configures a tunnel and enters interface configuration mode.  
  Example:  
  ```
  Router(config)# interface tunnel 100
  ``` |
| **Step 6** tunnel mode mpls traffic-eng point-to-multipoint | Enables MPLS point-to-multipoint traffic engineering on the tunnel.  
  Example:  
  ```
  Router(config-if)# tunnel mode mpls traffic-eng point-to-multipoint
  ``` |
| **Step 7** tunnel destination list mpls traffic-eng {identifier|dest-list-id|namedest-list-name} | Specifies a destination list to specify the IP addresses of point-to-multipoint destinations.  
  Example:  
  ```
  Router(config-if)# tunnel destination list mpls traffic-eng name in-list-01
  ``` |
| **Step 8** ip igmp static-group {# | group-address  
  [source|source-address | ssm-map] |  
  class-map|class-map-name} | Configures static group membership entries on an interface.  
  Example:  
  ```
  ``` |

14. `ip explicit-path {name|word|identifier number} [enable | disable]`
15. `next-address [loose | strict] ip-address`
16. `end`
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| Router(config-if)# ip igmp static-group 239.100.100.101 source 10.11.11.11 | Enables Protocol Independent Multicast (PIM) on an interface.  
- An interface configured with passive mode does not pass or forward PIM control plane traffic; it passes or forwards only IGMP traffic. |

**Step 9**  
**Example:**  
Router(config-if)# ip pim passive

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>exit</td>
<td>Exits interface configuration mode.</td>
</tr>
</tbody>
</table>

**Step 10**  
**Example:**  
Router(config-if)# exit

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>mpls traffic-eng destination list {name</td>
<td>identifier}dest-list-name</td>
</tr>
</tbody>
</table>

**Step 11**  
**Example:**  
Router(config)# mpls traffic-eng destination list name in-list-01

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| ip ip-address path-option id {dynamic|explicit} | Specifies the IP addresses of MPLS point-to-multipoint traffic engineering tunnel destinations.  
- If you use the explicit keyword, you must configure explicit paths, using the **ip explicit-path** command.  
- Repeat this step for each destination. |

**Step 12**  
**Example:**  
Router(cfg-te-dest-list)# ip 10.10.10.10 path-option 1 dynamic

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>exit</td>
<td>Exits traffic engineering destination list configuration mode.</td>
</tr>
</tbody>
</table>

**Step 13**  
**Example:**  
Router(cfg-te-dest-list)# exit

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip explicit-path {name</td>
<td>identifier}number</td>
</tr>
</tbody>
</table>

**Step 14**  
**Example:**  
Router(config)# ip explicit-path name path1 enable

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>next-address [loose</td>
<td>strict] ip-address</td>
</tr>
</tbody>
</table>

**Step 15**  
**Example:**  
Router(cfg-ip-expl-path)# next-address 10.0.0.2
### Configuring the Midpoint Routers

No special configuration is needed to support the P2MP TE feature on the midpoint routers. The midpoint routers must have Cisco IOS Release 15.5(02)S or later release installed. They must be able to support and implement the P2MP signaling extensions. The MPLS TE configuration of the midpoint routers supports both P2P and P2MP TE. All multicast traffic is label switched. The midpoint routers do not require IPv4 multicast routing or PIM.

All the core interfaces on the mid-point routers, should have this configuration:

- `ip rsvp bandwidth`.
- `mpls traffic-eng tunnels`

The IGP, should have this configuration:

- `router ospf 1 mpls traffic-eng router-id Loopback0`, `mpls traffic-eng area 0`

For information on configuring MPLS TE, see MPLS Traffic Engineering and Enhancements.

### Configuring the Tailend Routers

The tailend routers remove the MPLS labels from the IP multicast packets and send the packets to the MFIB for regular multicast forwarding processing. You must issue the `ip mroute` command to configure a static route back to the headend router, thus enabling RPF checks.

The following task explains how to configure PIM on the egress interface of the PE router. PIM is needed when the egress PE router is connected to a CE router, which is connected to a LAN where one or more multicast receivers are connected.

If the egress PE router is directly connected to a decoder device/system (e.g., DCM), you must configure Internet Group Management Protocol (IGMP) on the egress interface of the PE router. For more information on configuring IGMP, see Customizing IGMP.

### SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `ip multicast-routing [vrf vrf-name] [distributed]`
4. `ip multicast mpls traffic-eng [range access-list-number | access-list-name]`
5. `interface type slot / port`
6. `ip pim {dense-mode [proxy-register {list access-list | route-map map-name}] | passive | sparse-mode | sparse-dense-mode}`
7. `exit`
8. `ip mroute [vrf vrf-name] source-address mask {fallback-lookup {global | vrf vrf-name} | rpf-address | interface-type interface-number} [distance]`
9. `end`
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** | **enable** | Enables privileged EXEC mode.  
*Example:*  
`Router> enable` |
| **Step 2** | **configure terminal** | Enters global configuration mode.  
*Example:*  
`Router# configure terminal` |
| **Step 3** | **ip multicast-routing [vrf vrf-name] [distributed]** | Enables IP multicast routing globally.  
*Example:*  
`Router(config)# ip multicast-routing` |
| **Step 4** | **ip multicast mpls traffic-eng [range access-list-number | access-list-name]** | Enables IP multicast routing for MPLS traffic engineering point-to-multipoint tunnels.  
*Example:*  
`Router(config)# ip multicast mpls traffic-eng` |
| **Step 5** | **interface type slot / port** | Configures an interface type and enters interface configuration mode.  
*Example:*  
*or*  
`interface type slot/port-adapter/port`  
*Example:*  
`Router(config)# interface ethernet 1/1`  
*or*  
`Router(config)# interface fastethernet 1/0/0` |
| **Step 6** | **ip pim {dense-mode [proxy-register {list access-list | route-map map-name}] | passive | sparse-mode | sparse-dense-mode}** | Enables Protocol Independent Multicast (PIM) on an interface.  
*Example:*  
  
---

**Example:**  
- The `type` argument specifies the type of interface to be configured.  
- The `slot` argument specifies the slot number. Refer to the appropriate hardware manual for slot and port information.  
- The `port` argument specifies the port number. Refer to the appropriate hardware manual for slot and port information.  
- The `port-adapter` argument specifies the port adapter number. Refer to the appropriate hardware manual for information about port adapter compatibility.
### Configuring FRR with P2MP TE Tunnels

To enable link protection for sub-LSPs associated with a P2MP TE tunnel, perform the following configuration tasks:

- Enable FRR on the headend router for each P2MP TE tunnel.
- Configure P2P backup tunnels for network interfaces that require protection.

See MPLS Traffic Engineering—Fast Reroute Link and Node Protection for information and configuration instructions.

### Enabling MPLS Traffic Engineering System Logging of Events

MPLS Traffic Engineering system logging allows you to view the following events:

- Setting up and tearing down of LSPs
- RSVP Path and RESV requests
- Sub-LSP status (through path-change messages)

Commands to enable system logging include:

- `mpls traffic-eng logging lsp path-errors`
- `mpls traffic-eng logging lsp preemption`
- `mpls traffic-eng logging lsp reservation-errors`
Verifying the Configuration of MPLS Point-to-Multipoint Traffic Engineering

This section includes the following tasks:

Verifying the Configuration of the Headend Router

At the headend router, use the following steps to verify that:

- All sub-LSPs are enabled.
- IP multicast traffic is being forwarded onto the P2MP TE tunnel.

The following commands may also be helpful in the verification of the headend router:

- `show cef path set` and `show cef path set detail` (when the headend router is also a branch point)
- `show ip mfib` and `show ip mfib verbose`
- `show ip rsvp fast-reroute`
- `show mpls traffic-eng destination list`
- `show mpls traffic-eng fast-reroute database`
- `show mpls traffic-eng tunnels with the dest-mode p2mp, detail, and summary` keywords

### SUMMARY STEPS

1. `enable`
2. `show mpls traffic-eng tunnels brief`
3. `show mpls traffic-eng forwarding path-set brief`
4. `show mpls traffic-eng forwarding path-set detail`
5. `show ip mroute`

### DETAILED STEPS

**Step 1** `enable`

Issue the `enable` command to enter privileged EXEC mode.

**Step 2** `show mpls traffic-eng tunnels brief`

Use the `show mpls traffic-eng tunnels brief` command to display the P2MP TE tunnels originating from the headend router. For example:

**Example:**

```
Router# show mpls traffic-eng tunnels brief
```
Verifying the Configuration of the Headend Router

signaling Summary:
- LSP Tunnels Process: running
- Passive LSP Listener: running
- RSVP Process: running
- Forwarding: enabled
- Periodic reoptimization: every 60 seconds, next in 5 seconds
- Periodic FRR Promotion: Not Running
- Periodic auto-bw collection: disabled

P2P TUNNELS:
- TUNNEL NAME DESTINATION UP IF DOWN IF STATE/PROT
  - p2p-LSP 10.2.0.1 - Se2/0 up/up

Displayed 2 (of 2) heads, 0 (of 0) midpoints, 0 (of 0) tails

P2MP TUNNELS:
- DEST CURRENT INTERFACE STATE/PROT UP/CFG TUNID LSPID
  - Tunnel2 up/up 3/10 2 1
  - Tunnel5 up/down 1/10 5 2

Displayed 2 (of 2) P2MP heads

P2MP SUB-LSPS:
- SOURCE TUNID LSPID DESTINATION SUBID ST UP IF DOWN IF
  - 10.1.0.1 2 1 10.2.0.1 1 up head Se2/0
  - 10.1.0.1 2 1 10.3.0.199 2 up head Et2/0
  - 10.1.0.1 2 1 19.4.0.1 2 up head s2/0
  - 10.1.0.1 5 2 10.5.0.1 7 up head e2/0
  - 100.100.100.100 1 3 200.200.200.200 1 up ge2/0 s2/0
  - 100.100.100.100 1 3 10.1.0.1 1 up e2/0 tail

Displayed 7 P2MP sub-LSPs:
  - 5 (of 5) heads, 1 (of 1) midpoints, 1 (of 1) tails

Step 3 show mpls traffic-eng forwarding path-set brief

Use the `show mpls traffic-eng forwarding path-set brief` command to show the sub-LSPs that originate from the headend router. The following example shows three sub-LSPs originating at the headend router and going to different destinations. All the sub-LSPs belong to the same path set, which is a collection of paths. The path set is given a unique ID, which is shown in the PSID column of the example:

Example:

```
Router# show mpls traffic-eng forwarding path-set brief
Sub-LSP Identifier
src_lspid[subid]->dst_tunid | InLabel | Next Hop | I/F | PSID
---------------------------------|---------|----------|----|------
10.0.0.1_19[16]->10.0.0.8_1 | none    | 10.0.1.2 | Et0/0 | C5000002
10.0.0.1_19[27]->10.0.0.6_1 | none    | 10.0.1.2 | Et0/0 | C5000002
10.0.0.1_19[31]->10.0.0.7_1 | none    | 10.0.1.2 | Et0/0 | C5000002
```

Step 4 show mpls traffic-eng forwarding path-set detail

Use the `show mpls traffic-eng forwarding path-set detail` command to show more information about the sub-LSPs that originate from the headend router. For example:

Example:

```
Router# show mpls traffic-eng forwarding path-set detail
LSP: Source: 10.1.0.1, TunID: 100, LSPID: 7
  Destination: 10.2.0.1, P2MP Subgroup ID: 1
  Path Set ID: 0x300000001
OutLabel : Serial2/0, 16
Next Hop : 10.1.3.2
```

IP Multicast: Multicast Configuration Guide, Cisco IOS XE Release 3S (Cisco ASR 900 Series)
**FRR OutLabel**: Tunnel666, 16
**LSP**: Source: 10.1.0.1, TunID: 100, LSPID: 7
**Destination**: 10.3.0.1, P2MP Subgroup ID: 2
  - Path Set ID: 0x30000001
  - OutLabel: Serial12/0, 16
  - Next Hop: 10.1.3.2
**FRR OutLabel**: Tunnel666, 16

**Step 5**

```
show ip mroute
```

Use the `show ip mroute` command to verify that IP multicast traffic is being forwarded to the P2MP TE tunnel. In the following example, the output shown in bold shows that Tunnel 1 is part of the outgoing interface list for multicast group 232.0.1.4 with a source address of 10.10.10:

**Example:**

```
Router# show ip mroute
IP Multicast Routing Table
  Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
  L - Local, P - Pruned, R - RP-bit set, F - Register flag,
  T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
  X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
  U - URD, I - Received Source Specific Host Report,
  Z - Multicast Tunnel, z - MDT-data group sender,
  Y - Joined MDT-data group, y - Sending to MDT-data group,
  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
(10.10.10.10, 232.0.1.4), 1d00h/stopped, flags: sTI
  Incoming interface: Ethernet2/0, RPF nbr 10.10.1.1
  Outgoing interface list:
    Tunnel1, Forward/Sparse-Dense, 1d00h/00:01:17
(*, 224.0.1.40), 1d00h/00:02:48, RP 0.0.0.0, flags: DCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet2/0, Forward/Sparse, 1d00h/00:02:48
```

**Verifying the Configuration of the Midpoint Routers**

At the midpoint router, use the following commands to verify that MPLS forwarding occurs. If the midpoint router is branch router, you can also use `show mpls forwarding-table labels` command to display show specific labels.

**SUMMARY STEPS**

1. `enable`
2. `show mpls forwarding-table`

**DETAILED STEPS**

**Step 1**

`enable`

Issue the `enable` command to enter privileged EXEC mode.

**Step 2**

`show mpls forwarding-table`
Verifying the Configuration of the Tailend Routers

At the tailend router, use the following steps to verify that:

- MPLS forwarding occurs.
- IP multicast forwarding occurs.

You can also use the `show ip mfb`, `showmpls traffic-eng destination list`, and `show mpls traffic-eng tunnels dest-mode p2mp` commands for verification.

**SUMMARY STEPS**

1. `enable`
2. `show mpls forwarding-table`
3. `show ip mroute`

**DETAILED STEPS**

**Step 1** enable

Issue the `enable` command to enter privileged EXEC mode.

**Step 2** `show mpls forwarding-table`

Use the `show mpls forwarding-table` command to show that MPLS labeled packets are forwarded from the tailend router without any label.

**Example:**

```
Router# show mpls forwarding-table
Local Outgoing Prefix Bytes Label Outgoing Next Hop
Label Label or Tunnel Id Switched interface
17 [T] No Label 10.0.0.1 1 [19] 342 aggregate
```

IP Multicast: Multicast Configuration Guide, Cisco IOS XE Release 3S (Cisco ASR 900 Series)
Forwarding through a LSP tunnel.

Router# show mpls forwarding-table detail

<table>
<thead>
<tr>
<th>Local Label</th>
<th>Outgoing Prefix</th>
<th>Bytes Label</th>
<th>Label or Tunnel Id</th>
<th>Switched interface</th>
<th>Outgoing Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>No Label</td>
<td>10.0.0.1</td>
<td>1 [19]</td>
<td>aggregate</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>MAC/Encaps=0/0, MRU=0, Label Stack{}, via Ls0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Step 3  show ip mroute

Use the `show ip mroute` command to display IP multicast traffic. In the following example, the output in bold shows the incoming interface is Lspvif0 and the outgoing interface is Ethernet1/0 is for multicast group 232.0.1.4 with source address 10.10.10.10:

Example:

```bash
Router# show ip mroute
IP Multicast Routing Table
...
(*, 232.0.1.4), 1d02h/stopped, RP 0.0.0.0, flags: SP
   Incoming interface: Null, RPF nbr 0.0.0.0
   Outgoing interface list: Null
(10.10.10.10, 232.0.1.4), 00:01:51/00:01:38, flags:
   Incoming interface: Lspvif0, RPF nbr 10.0.0.1, Mroute
   Outgoing interface list:
     Ethernet1/0, Forward/Sparse, 00:01:51/00:02:37
(*, 224.0.1.40), 1d02h/00:02:37, RP 0.0.0.0, flags: DCL
   Incoming interface: Null, RPF nbr 0.0.0.0
   Outgoing interface list:
     Ethernet1/0, Forward/Sparse, 1d02h/00:02:37
```

Configuration Examples for MPLS Point-to-Multipoint Traffic Engineering

The following examples show point-to-multipoint traffic engineering configurations on the headend router (PE5), a midpoint router (P1), and a tailend router (PE1):
Example Configuration of the Headend Router (PE5)

In the following example configuration of the headend router, note the following:

- IPv4 multicast routing is enabled with the `ipmulticast-routing` command.
- Two destination lists are specified, one for dynamic paths and one for explicit paths. The destination list specifies one path-option per destination.
- The `tunnelmodempstraffic-engpoint-to-multipoint` command enables the P2MP tunnel.
- On the tunnel interfaces, the `ippimpassive` command is used.
- On the non-MPLS interfaces, the `ippimsparse-mode` command is used.
- The `ipigmpstatic-group` commands map the multicast groups to the P2MP tunnel.
• FRR is enabled on the router, with tunnel 3 as the backup path. An explicit path called PE5->P1-BKUP provides the alternative path.

hostname [PE5]
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
!
clock timezone PST -8
ip subnet-zero
ip source-route
ip cef
no ip domain lookup
!
ip multicast-routing
!
no ipv6 cef
mpls traffic-eng tunnels
!
mlps traffic-eng destination list name P2MP-DYN-DST-LIST
ip 172.16.255.1 path-option 10 dynamic
ip 172.16.255.2 path-option 10 dynamic
ip 172.16.255.3 path-option 10 dynamic
ip 172.16.255.4 path-option 10 dynamic
!
mlps traffic-eng destination list name P2MP-EXCIT-DST-LIST
ip 172.16.255.1 path-option 10 explicit identifier 101
ip 172.16.255.2 path-option 10 explicit identifier 102
ip 172.16.255.3 path-option 10 explicit identifier 103
ip 172.16.255.4 path-option 10 explicit identifier 104
!
multilink bundle-name authenticated
!
interface Tunnel1
description PE5->PE1,PE2,PE3,PE4-DYN
ip unnumbered Loopback0
ip pim passive
ip igmp static-group 232.0.1.4 source 192.168.5.255
ip igmp static-group 232.0.1.3 source 192.168.5.255
ip igmp static-group 232.0.1.2 source 192.168.5.255
ip igmp static-group 232.0.1.1 source 192.168.5.255
tunnel mode mpls traffic-eng point-to-multipoint
tunnel destination list mpls traffic-eng name P2MP-DYN-DST-LIST
tunnel mpls traffic-eng priority 7 7
tunnel mpls traffic-eng bandwidth 10000
!
interface Tunnel2
description PE5->PE1,PE2,PE3,PE4-EXCIT
ip unnumbered Loopback0
ip pim passive
ip igmp static-group 232.0.1.8 source 192.168.5.255
ip igmp static-group 232.0.1.7 source 192.168.5.255
ip igmp static-group 232.0.1.6 source 192.168.5.255
ip igmp static-group 232.0.1.5 source 192.168.5.255
tunnel mode mpls traffic-eng point-to-multipoint
tunnel destination list mpls traffic-eng name P2MP-EXCIT-DST-LIST
tunnel mpls traffic-eng priority 7 7
tunnel mpls traffic-eng bandwidth 20000
!
interface Tunnel3
description PE5->P1
ip unnumbered Loopback0
tunnel mode mpls traffic-eng
tunnel destination 172.16.255.201
  tunnel mpls traffic-eng path-option 10 explicit name PE5->P1-BKUP

interface Loopback0
  ip address 172.16.255.5 255.255.255.255

interface Ethernet0/0
description CONNECTS to CE5
  ip address 192.168.5.1 255.255.255.252
  ip pim sparse-mode

interface Ethernet1/0
description CONNECTS to P1
  bandwidth 1000000
  ip address 172.16.0.13 255.255.255.254
  ip router isis
  mpls traffic-eng tunnels
  mpls traffic-eng backup-path Tunnel3
  isis network point-to-point
  ip rsvp bandwidth percent 100

interface Ethernet2/0
description CONNECTS to P2
  bandwidth 1000000
  ip address 172.16.0.14 255.255.255.254
  ip router isis
  mpls traffic-eng tunnels
  isis network point-to-point
  ip rsvp bandwidth percent 100

router isis
  net 49.0001.1720.1625.5005.00
  is-type level-2-only
  metric-style wide
  passive-interface Loopback0
  mpls traffic-eng router-id Loopback0
  mpls traffic-eng level-2

! ip classless
! no ip http server
! ip pim ssm default
! ip explicit-path identifier 101 enable
  next-address 172.16.0.12
  next-address 172.16.192.0
  next-address 172.16.0.0
! ip explicit-path identifier 102 enable
  next-address 172.16.0.12
  next-address 172.16.192.0
  next-address 172.16.0.3
! ip explicit-path identifier 103 enable
  next-address 172.16.0.12
  next-address 172.16.192.0
  next-address 172.16.192.6
  next-address 172.16.0.6
!
Example Configuration of the Midpoint Router (P1)

In the following example configuration of the midpoint router, note the following:

- MPLS Traffic Engineering is enabled both globally and on the interface connecting to other core routers.
- MPLE TE extensions are enabled through the `mplstraffic-engrouter-id` and `mplstraffic-englevel` commands.

```
hostname [P1]
!
no aaa new-model
clock timezone PST -8
ip subnet-zero
ip source-route
ip cef
no ip domain lookup
!
no ipv6 cef
mpls traffic-eng tunnels
multilink bundle-name authenticated
!
interface Loopback0
  ip address 172.16.255.201 255.255.255.255
!
interface Ethernet0/0
  description CONNECTS TO P2
  bandwidth 1000000
  ip address 172.16.192.2 255.255.255.254
  ip router isis
  mpls traffic-eng tunnels
  isis network point-to-point
  ip rsvp bandwidth percent 100
!
interface Ethernet0/1
  no ip address
  shutdown
!
interface Ethernet0/2
  no ip address
  shutdown
!
interface Ethernet0/3
  no ip address
  shutdown
!
interface Ethernet1/0
  description CONNECTS TO P3
  bandwidth 1000000
  ip address 172.16.192.1 255.255.255.254
  ip router isis
```
Example Configuration of the Tailend Router (PE1)

In the following example configuration of the tailend router, note the following:

- IPv4 multicast routing is enabled with the `ip multicast-routing` command.
- On the non-MPLS interfaces, the `ip pim sparse-mode` command is used.
- The `ip multicastmpls` commands enable multicast routing of traffic.

```
mpls traffic-eng tunnels
isis network point-to-point
ip rsvp bandwidth percent 100
!
interface Ethernet2/0
description CONNECTS TO PE5
bandwidth 1000000
ip address 172.16.0.12 255.255.255.254
ip router isis
mpls traffic-eng tunnels
isis network point-to-point
ip rsvp bandwidth percent 100
!
router isis
net 49.0001.1720.1625.5201.00
is-type level-2-only
metric-style wide
passive-interface Loopback0
mpls traffic-eng router-id Loopback0
mpls traffic-eng level-2
!
ip classless
!
no ip http server
```
no ip address
shutdown
mpls traffic-eng tunnels
ip rsvp bandwidth 155000
!
interface Ethernet1/0
description CONNECTS TO PE2
bandwidth 1000000
ip address 172.16.0.5 255.255.255.254
ip router isis
mpls traffic-eng tunnels
isis network point-to-point
ip rsvp bandwidth percent 100
!
interface Ethernet2/0
description CONNECTS TO P3
bandwidth 1000000
ip address 172.16.0.0 255.255.255.254
ip router isis
mpls traffic-eng tunnels
isis network point-to-point
ip rsvp bandwidth percent 100
!
router isis
net 49.0001.1720.1625.5001.00
is-type level-2-only
metric-style wide
passive-interface Loopback0
mpls traffic-eng router-id Loopback0
mpls traffic-eng level-2
!
!
ip classless
!
no ip http server
!
ip multicast mpls traffic-eng
ip pim ssm default
ip mroute 192.168.5.0 255.255.255.0 172.16.255.5
Example Configuration of the Tailend Router (PE1)
CHAPTER 12

IGMP Snooping

This module describes how to enable and configure the Ethernet Virtual Connection (EVC)-based IGMP Snooping feature both globally and on bridge domains.

• Finding Feature Information, on page 245
• Prerequisites for IGMP Snooping, on page 245
• Restrictions for IGMP Snooping, on page 246
• Information About IGMP Snooping, on page 246
• How to Configure IGMP Snooping, on page 247
• Verifying IGMP Snooping, on page 252

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

• 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.
• Maximum number of EFPs supported on a single BD with IGMP snooping enabled:
  • RSP1 is 48
  • RSP2 is 128
• EFPs are supported only on different ports of a single BD, but not on the same ports on the RSP3 module.
• Maximum number of multicast routes for Layer 2 is 1000.
• Maximum number of multicast routes for Layer 3 (including control entries) for RSP1A, RSP1B, and RSP2 modules is 1000.

• Maximum number of multicast routes for Layer 2 and Layer 3 for the RSP3 module is 1000.

---

**Note**

We recommend a delay of at least 2 minutes while performing the below actions:

• Removal and addition of EFP configuration operation.

• Removal and addition of bridge-domain interface (BDI) configuration operation.

• Changing the interface configuration to default and reconfiguring the EFP again.

• Removing and adding IGMP snooping to a bridge-domain.

---

**Restrictions for IGMP Snooping**

• Layer 2 multicast is not supported with IGMP snooping when static joins are configured in EFP or TEFP. However, Layer 2 multicast with IGMP snooping is supported for dynamic joins configured on the EFP or TEFP.

• IGMP snooping is not supported with bridge domain interfaces greater than 4094.

• IGMP snooping must be turned off on the bridge domain when VPLS is configured, for IGMP reports to be sent over the VPLS pseudowire.

• Stateful switchover (SSO) is not supported for IGMP snooping.

• Static mrouter configuration is not supported.

• IGMP snooping for Trunk EFPs, port channels interfaces and pseudowires is not supported on the RSP1 module.

• IGMP snooping for EFPs and Trunk EPFs is supported on the RSP3 module.

Starting with Cisco IOS Release 3.14, IGMP snooping is supported on the TEFPs on the RSP2 modules. IGMP snooping on RSP2 is not supported for port channels and pseudowires.

---

**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 router. IGMP snooping is supported with REP and G.8032.

---

Note

IGMP snooping is not supported with REP and G.8032 on the RSP3 module.

---

**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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device&gt; <code>enable</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# <code>configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>ip igmp snooping</code></td>
<td>Globally enables IGMP snooping after it has been disabled.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
</tbody>
</table>
### Purpose

**Command or Action**

Device(config)# ip igmp snooping

**Purpose**

(Optional) Enters bridge domain configuration mode.

<table>
<thead>
<tr>
<th>Step 4</th>
<th>bridge-domain bridge-id</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config)# bridge-domain 100</td>
</tr>
</tbody>
</table>

(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.

<table>
<thead>
<tr>
<th>Step 5</th>
<th>ip igmp snooping</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-bdomain)# ip igmp snooping</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 6</th>
<th>end</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-bdomain)# end</td>
</tr>
</tbody>
</table>

### 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**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>enable</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device&gt; enable</td>
</tr>
</tbody>
</table>

Enables privileged EXEC mode.

- Enter your password if prompted.

<table>
<thead>
<tr>
<th><strong>Step 2</strong></th>
<th>configure terminal</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Device# configure terminal</td>
</tr>
</tbody>
</table>

Enters global configuration mode.

<table>
<thead>
<tr>
<th><strong>Step 3</strong></th>
<th>ip igmp snooping robustness-variable variable</th>
</tr>
</thead>
</table>
| **Example:** | | Configures the IGMP defined robustness variable.
IGMP Snooping

Configuring IGMP Snooping on a Bridge Domain

Before you begin
- The bridge domain must be created.

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device(config)# ip igmp snooping robustness-variable 3</td>
<td>Enables report suppression for IGMP snooping.</td>
</tr>
<tr>
<td>Step 4 ip igmp snooping report-suppression</td>
<td>Configures how often IGMP snooping sends query messages in response to receiving an IGMP leave message. The default is 2.</td>
</tr>
<tr>
<td>Example: Device(config)# ip igmp snooping report-suppression</td>
<td></td>
</tr>
<tr>
<td>Step 5 ip igmp snooping last-member-query-count count</td>
<td>Configures the length of time after which the group record is deleted if no reports are received. The default is 1000 milliseconds.</td>
</tr>
<tr>
<td>Example: Device(config)# ip igmp snooping last-member-query-count 5</td>
<td></td>
</tr>
<tr>
<td>Step 6 ip igmp snooping last-member-query-interval interval</td>
<td></td>
</tr>
<tr>
<td>Example: Device(config)# ip igmp snooping last-member-query-interval 200</td>
<td></td>
</tr>
<tr>
<td>Step 7 ip igmp snooping check ttl</td>
<td>Enforces IGMP snooping check.</td>
</tr>
<tr>
<td>Example: Device(config)# ip igmp snooping check ttl</td>
<td></td>
</tr>
<tr>
<td>Step 8 exit</td>
<td>Exits global configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Device(config)# exit</td>
<td></td>
</tr>
</tbody>
</table>
9. `ip igmp snooping check ttl`
10. `end`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><code>enable</code></td>
<td>- Enter your password if prompted.</td>
</tr>
<tr>
<td>Example:</td>
<td>Device&gt; enable</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Device# configure terminal</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Enters bridge domain configuration mode.</td>
</tr>
<tr>
<td><code>bridge-domain bridge-id</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config)# bridge-domain 100</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Enables IGMPv2 immediate-leave processing.</td>
</tr>
<tr>
<td><code>ip igmp snooping immediate-leave</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-bdomain)# ip igmp snooping immediate-leave</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>When both immediate-leave processing and the query count are configured, fast-leave processing takes precedence.</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>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.</td>
</tr>
<tr>
<td><code>ip igmp snooping last-member-query-count count</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-bdomain)# ip igmp snooping last-member-query-count 5</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>When both immediate-leave processing and the query count are configured, fast-leave processing takes precedence.</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>Sets the last member query interval of the bridge domain. The valid range is from 100 to 32767. The default is 1000 milliseconds.</td>
</tr>
<tr>
<td><code>ip igmp snooping last-member-query-interval interval</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-bdomain)# ip igmp snooping last-member-query-interval 2000</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>Configures the IGMP snooping robustness variable. The default is 2.</td>
</tr>
<tr>
<td><code>ip igmp snooping robustness-variable variable</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-bdomain)# ip igmp snooping robustness-variable 3</td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>Enables report suppression for all hosts on the bridge domain.</td>
</tr>
<tr>
<td><code>ip igmp snooping report-suppression</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
</tbody>
</table>
### Disabling IGMP Snooping Globally

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `no ip igmp snooping`
4. `exit`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>Device&gt; enable</code></td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>Device# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td><code>no ip igmp snooping</code></td>
<td>Disables IGMP snooping on the router.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>Device(config)# no ip igmp snooping</code></td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td><code>exit</code></td>
<td>Exits global configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>Device(config)# exit</code></td>
<td></td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Device&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> bridge-domain bridge-id</td>
<td>Enters bridge domain configuration mode.</td>
</tr>
<tr>
<td>Example: Device(config)# bridge-domain 4000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> no ip igmp snooping</td>
<td>Disables IGMP snooping on the bridge domain.</td>
</tr>
<tr>
<td>Example: Device(config-bdomain)# no ip igmp snooping</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> end</td>
<td>Returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Device(config-bdomain)# end</td>
<td></td>
</tr>
</tbody>
</table>

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
```
IGMP Snooping

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 : Yes

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:

```plaintext
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:

```plaintext
Router# show ip igmp snooping counters
Counters of group "IGMP snooping counters" overall there are 15 counters

<table>
<thead>
<tr>
<th>Type</th>
<th>Value</th>
<th>Ovr</th>
<th>Und</th>
</tr>
</thead>
<tbody>
<tr>
<td>RX processed Query Count</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RX processed Group Specific Query</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RX processed Join</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RX processed Leave</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RX processed Total Valid Packets</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RX processed Other Packets</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RX Packets dropped for sanity errors</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RX Packets dropped for checksum errors</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RX Packets dropped for header length errors</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RX Packets dropped for other errors</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RX processed Topology change notification</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TX processed Query Count</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TX processed Group Specific Query</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TX processed Join</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TX processed Leave</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Counters of group "IGMP snooping V3 counters" overall there are 18 counters

<table>
<thead>
<tr>
<th>Type</th>
<th>Value</th>
<th>Ovr</th>
<th>Und</th>
</tr>
</thead>
<tbody>
<tr>
<td>RX processed V3 ALLOW NEW</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RX processed V3 BLOCK OLD</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RX processed V3 MODE IS INCLUDE</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RX processed V3 MODE IS EXCLUDE</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RX processed V3 CHANGE TO INCLUDE</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RX processed V3 CHANGE TO EXCLUDE</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RX processed V3 Query</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RX processed V3 Group Specific Query</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TX processed V3 ALLOW NEW</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TX processed V3 BLOCK OLD</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TX processed V3 MODE IS INCLUDE</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TX processed V3 MODE IS EXCLUDE</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TX processed V3 CHANGE TO INCLUDE</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TX processed V3 CHANGE TO EXCLUDE</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TX processed V3 Query</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TX processed V3 Group Specific Query</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

• **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
```
Verifying IGMP Snooping
CHAPTER 13

MLDP-Based MVPN

The MLDP-based MVPN feature provides extensions to Label Distribution Protocol (LDP) for the setup of point-to-multipoint (P2MP) and multipoint-to-multipoint (MP2MP) label switched paths (LSPs) for transport in the Multicast Virtual Private Network (MVPN) core network.

- Finding Feature Information, on page 257
- Prerequisites for MLDP-Based MVPN, on page 257
- Restrictions for MLDP-Based MVPN on the RSP2 Module, on page 258
- Information About MLDP-Based MVPN, on page 258
- Configuring MLDPv4-Based MVPN, on page 270
- MLDPv6-Based MVPN, on page 275
- Configuration Examples for MLDP-Based MVPN, on page 279

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 MLDP-Based MVPN

- You must be familiar with IPv4 multicast routing configuration tasks and concepts.
- Cisco Express Forwarding (CEF) must be enabled on the router for label switching.
- Unicast routing must be operational.
- To enable MLDP based multicast VPN, you must configure a VPN routing and forwarding (VRF) instance. For more information about configuring a VRF, see MPLS Layer 3 VPN Configuration Guide for your router.
Restrictions for MLDP-Based MVPN on the RSP2 Module

The following restrictions apply to the MLDP-Based MVPN feature on the RSP2 module:

- mLDPv4 extranet is supported from Cisco IOS XE Release 3.17.0S onwards on RSP2 module.
- Route leaking (Leaking the mroutes/traffic from Virtual Routing and Forwarding (VRF) to global table or to between VRFs) for mLDPv4 is supported from Cisco IOS XE Release 3.17.0S onwards.
- MLDP FRR is partially supported (only link level).
- Spanning of mLDP packets in the bud node is not supported.
- Bridge domain interfaces (BDI) in the core is not supported for mLDPv4 in Cisco IOS XE Release 3.17; Only routed port in the core is supported. Starting with Cisco IOS Release 3.18 and later, MLDPv4 and MLDPv6 with BDI interfaces in core is supported.
- MLDP and Point-to-Multipoint Traffic Engineering (P2MP TE) coexistence is not supported.
- MLDP and MVPN GRE cannot co-exist in the same VRF.
- More than 20 Multicast VRFs (MVRFs) for MLDP is not supported.
- Following multicast groups are not supported:
  - IPv4 on the RSP2 module - more than 2000
  - IPv6 on the RSP2 module - more than 1000
- The no mpls mlpd forwarding recursive command is required when multiple paths exist.
- When ip multicast-routing is disabled on bud node, traffic for the downstream tail ends is not be forwarded.
- Routed port-channel, EFP and trunk EFP-based port-channel is not supported as core facing interface on the RSP2 module.

Information About MLDP-Based MVPN

Overview of MLDP-Based MVPN

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

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

An MVPN allows an enterprise to transparently interconnect its private network across the network backbone of a service provider. The use of an MVPN to interconnect an enterprise network in this way does not change the way that the enterprise network is administered, nor does it change general enterprise connectivity.
As shown in the figure, in an MLDP-based MVPN, a static default multicast distribution tree (MDT) is established for each multicast domain. The default MDT defines the path used by provider edge (PE) devices to send multicast data and control messages to every other PE device in the multicast domain. A default MDT is created in the core network using a single MP2MP LSP. The default MDT behaves like a virtual LAN.

*Figure 34: MLDP with the Default MDT Scenario*

As shown in the figure, an MLDP-based MVPN also supports the dynamic creation of data MDTs for high-bandwidth transmission. For high-rate data sources, a data MDT is created using P2MP LSPs to off-load traffic from the default MDT to avoid unnecessary waste of bandwidth to PEs that did not join the stream. The creation of the data MDT is signaled dynamically using MDT Join TLV messages. Data MDTs are a feature unique to Cisco IOS software. Data MDTs are intended for high-bandwidth sources such as full-motion video inside the VPN to ensure optimal traffic forwarding in the MPLS VPN core. The threshold at which the data MDT is created can be configured on a per-device or a per-VRF basis. When the multicast transmission exceeds the defined threshold, the sending PE device creates the data MDT and sends a User Datagram Protocol (UDP) message, which contains information about the data MDT to all devices on the default MDT.
Data MDTs are created only for (S, G) multicast route entries within the VRF multicast routing table. They are not created for (*, G) entries regardless of the value of the individual source data rate.

The only transport mechanism previously available was Protocol Independent Multicast (PIM) with Multipoint Generic Routing Encapsulation (mGRE) over an IP core network. The introduction of Multicast Label Distribution Protocol (MLDP) provides transport by using MLDP with label encapsulation over an MPLS core network.

MLDP creates the MDTs as follows:

- The default MDT uses MP2MP LSPs.
  - Supports low bandwidth and control traffic between VRFs.
- The data MDT uses P2MP LSPs.
  - Supports a single high-bandwidth source stream from a VRF.

All other operations of MVPN remain the same regardless of the tunneling mechanism:

- PIM neighbors in a VRF are seen across a Label Switched Path virtual interface (LSP-VIF).
- The VPN multicast state is signaled by PIM.

The only other difference when using MLDP is that the MDT group address used in the mGRE solution is replaced with a VPN ID.

**Benefits of MLDP-Based MVPN**

- Enables the use of a single MPLS forwarding plane for both unicast and multicast traffic.
- Enables existing MPLS protection (for example, MPLS Traffic Engineering/Resource Reservation Protocol (TE/Rsvp link protection) and MPLS Operations Administration and Maintenance (OAM) mechanisms to be used for multicast traffic.
Reduces operational complexity due to the elimination of the need for PIM in the MPLS core network.

**P2MP and MP2MP Label Switched Paths**

MLDP is an application that sets up Multipoint Label Switched Paths (MP LSPs) in MPLS networks without requiring multicast routing protocols in the MPLS core. mLDP constructs the P2MP or MP2MP LSPs without interacting with or relying upon any other multicast tree construction protocol. Using LDP extensions for MP LSPs and Unicast IP routing, mLDP can setup MP LSPs. The two types of MP LSPs that can be setup are Point-to-Multipoint (P2MP) and Multipoint-to-Multipoint (MP2MP) type LSPs.

A P2MP LSP allows traffic from a single root (ingress node) to be delivered to a number of leaves (egress nodes), where each P2MP tree is uniquely identified with a 2-tuple (root node address, P2MP LSP identifier). A P2MP LSP consists of a single root node, zero or more transit nodes, and one or more leaf nodes, where typically root and leaf nodes are PEs and transit nodes are P routers. A P2MP LSP setup is receiver-driven and is signaled using mLDP P2MP FEC, where LSP identifier is represented by the MP Opaque Value element. MP Opaque Value carries information that is known to ingress LSRs and Leaf LSRs, but need not be interpreted by transit LSRs. There can be several MP LSPs rooted at a given ingress node, each with its own identifier.

A MP2MP LSP allows traffic from multiple ingress nodes to be delivered to multiple egress nodes, where a MP2MP tree is uniquely identified with a 2-tuple (root node address, MP2MP LSP identifier). For a MP2MP LSP, all egress nodes, except the sending node, receive a packet sent from an ingress node.

A MP2MP LSP is similar to a P2MP LSP, but each leaf node acts as both an ingress and egress node. To build an MP2MP LSP, you can setup a downstream path and an upstream path so that:

- Downstream path is setup just like a normal P2MP LSP
- Upstream path is setup like a P2P LSP towards the upstream router, but inherits the downstream labels from the downstream P2MP LSP.

---

**Note**

We recommend that you configure one P2MP MDT tree per prefix. For example, if 500 multicast routes are needed, then you should configure at least 500 P2MP MDT trees.

```bash
ip vrf vrf-name1
    mdt data mpls mldp 500
```

---

**Packet Flow in mLDP-based Multicast VPN**

For each packet coming in, MPLS creates multiple out-labels. Packets from the source network are replicated along the path to the receiver network. The CE1 router sends out the native IP multicast traffic. The PE1 router imposes a label on the incoming multicast packet and replicates the labeled packet towards the MPLS core network. When the packet reaches the core router (P), the packet is replicated with the appropriate labels for the MP2MP default MDT or the P2MP data MDT and transported to all the egress PEs. Once the packet reaches the egress PE, the label is removed and the IP multicast packet is replicated onto the VRF interface.
Realizing a mLDP-based Multicast VPN

There are different ways a Label Switched Path (LSP) built by mLDP can be used depending on the requirement and nature of application such as:

- P2MP LSPs for global table transit Multicast using in-band signaling.
- P2MP/MP2MP LSPs for MVPN based on MI-PMSI or Multidirectional Inclusive Provider Multicast Service Instance (Rosen Draft).
- P2MP/MP2MP LSPs for MVPN based on MS-PMSI or Multidirectional Selective Provider Multicast Service Instance (Partitioned E-LAN).

The router performs the following important functions for the implementation of MLDP:

1. Encapsulating VRF multicast IP packet with GRE/Label and replicating to core interfaces (imposition node).
2. Replicating multicast label packets to different interfaces with different labels (Mid node).
3. Decapsulate and replicate label packets into VRF interfaces (Disposition node).

Supported mLDP Profiles

<table>
<thead>
<tr>
<th>Profile Name</th>
<th>Supported on mLDPv4</th>
<th>Supported on mLDPv6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile 1 Default MDT - MLDP MP2MP - PIM C-mcast Signaling</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Profile 2 Partitioned MDT - MLDP MP2MP - PIM C-mcast Signaling</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Profile 4 Partitioned MDT - MLDP MP2MP - BGP-AD - PIM C-mcast Signaling</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Profile 5 Partitioned MDT - MLDP P2MP - BGP-AD - PIM C-mcast Signaling</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Profile 6 VRF MLDP - In-band Signaling</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Profile 7 Global MLDP In-band Signaling</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Profile 9 Default MDT - MLDP - MP2MP - BGP-AD - PIM C-mcast Signaling</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Profile 12 Default MDT - MLDP - P2MP - BGP-AD - BGP C-mcast Signaling</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Profile Name</td>
<td>Supported on mLDPv4</td>
<td>Supported on mLDPv6</td>
</tr>
<tr>
<td>------------------------------------------------------------------------------</td>
<td>---------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>Profile 13 Default MDT - MLDP - MP2MP - BGP-AD - BGP C-mcast Signaling</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Profile 14 Partitioned MDT - MLDP P2MP - BGP-AD - BGP C-mast Signaling</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Profile 15 Partitioned MDT - MLDP MP2MP - BGP-AD - BGP C-mast Signaling</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Profile 17 Default MDT - MLDP - P2MP - BGP-AD - PIM C-mcast Signaling</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>


**Initial Deployment of an MLDP-Based MVPN**

Initial deployment of an MLDP-based MVPN involves the configuration of a default MDT and one or more data MDTs.

A static default MDT is established for each multicast domain. The default MDT defines the path used by PE devices to send multicast data and control messages to every other PE device in the multicast domain. A default MDT is created in the core network using a single MP2MP LSP.

An MLDP-based MVPN also supports the dynamic creation of data MDTs for high-bandwidth transmission.

**Default MDT Creation**

The figure shows the default MDT scenario. The Opaque value used to signal a default MDT consists of two parameters: the VPN ID and the MDT number for the VPN in the format (vpn-id, 0) where vpn-id is a manually configured 7-byte number that uniquely identifies this VPN. The default MDT is set to zero.

In this scenario, each of the three PE devices belong to the VRF called VRF and they have the same VPN ID. Each PE device with the same VPN ID will join the same MP2MP tree. The PE devices have created a primary MP2MP tree rooted at P-Central (Root 1) and a backup MP2MP tree rooted at PE-North (Root 2). There are two sources at PE-West and interested receivers at both PE-North and PE-East. PE-West will choose one of the MP2MP trees to transmit the customer VPN traffic, but all PE devices can receive traffic on either of the MP2MP trees.
LSP Downstream Default MDT Creation

The figures show the downstream tree creation for each of the roots. Each PE device configured with VPN ID 100:2 creates the same Forwarding Equivalence Class (FEC) Type Length Value (TLV), but with a different root and downstream labels per MP2MP tree. The FEC type will be MP2MP Down, which prompts the receiving Label Switched Route (LSR) to respond with an upstream label mapping message to create the upstream path.
Figure 37: Default MDT Downstream--Root 1

TLV | Label Map Msg
---|------------------
Type = MP2MP Down
FEC: Root = 172.30.20.1
Opaque = [mdt 100:2 0]
Label: 208

TLV | Label Map Msg
---|------------------
Type = MP2MP Down
FEC: Root = 172.30.20.1
Opaque = [mdt 100:2 0]
Label: 307

MLDP-Based MVPN

LSP Downstream Default MDT Creation
LSP Upstream Default MDT Creation

The figures show the upstream LSP creation for the default MDTs. For each downstream label received, a corresponding upstream label is sent. In the first figure, P-Central sends out three upstream labels (111, 109, and 105) to each downstream directly connected neighbor (downstream is away from the root). The process for PE-North is the same except that it only sends a single upstream label (313) as there is only one directly connected downstream neighbor, as shown in the second figure.
**Figure 39: Default MDT Upstream--Root 1**

<table>
<thead>
<tr>
<th>TLV</th>
<th>Label Map Msg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>MP2MP Up</td>
</tr>
<tr>
<td>FEC:</td>
<td>Root = 172.30.20.1, Opaque = [mtd 100:2 0]</td>
</tr>
<tr>
<td>Label:</td>
<td>109</td>
</tr>
</tbody>
</table>

**TLV**

<table>
<thead>
<tr>
<th>Label Map Msg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type = MP2MP Up</td>
</tr>
<tr>
<td>FEC: Root = 172.30.20.1, Opaque = [mtd 100:2 0]</td>
</tr>
<tr>
<td>Label: 111</td>
</tr>
</tbody>
</table>

```
PE-North (Root 2)
172.30.20.3

Join
(10.5.200.2, 238.1.200.1)
```

```
PE-West
172.30.20.2

PE-East
172.30.20.4

Join
(10.5.200.3, 238.1.200.2)
```

```
P-Central (Root 1)
172.30.20.1
```

```
VRF
```
PIM Overlay Signaling of VPN Multicast State

The signaling of the multicast state within a VPN is via PIM. It is called overlay signaling because the PIM session runs over the multipoint LSP and maps the VPN multicast flow to the LSP. In an MVPN, the operation of PIM is independent of the underlying tunnel technology. In the MVPN solution, a PIM adjacency is created between PE devices, and the multicast states within a VRF are populated over the PIM sessions. When using MLDP, the PIM session runs over an LSP-VIF interface. The figure shows PIM signaling running over the default MDT MP2MP LSP. Access to the MP2MP LSP is via the LSP-VIF, which can see all the leaf PE devices at the end of branches, much like a LAN interface. In the figure, PE-East sends a downstream label mapping message to the root, P-Central, which in turn sends an upstream label mapping message to PE-West. These messages result in the creation of the LSP between the two leaf PE devices. A PIM session can then be activated over the top of the LSP allowing the (S, G) states and control messages to be signaled between PE-West and PE-East. In this case, PE-East receives a Join TLV message for (10.5.200.3, 238.1.200.2) within VRF, which it inserts into the mroute table. The Join TLV message is then sent via the PIM session to PE-West (BGP next-hop of 10.5.200.3), which populates its VRF mroute table. This procedure is identical to the procedure using an mGRE tunnel.
In an MVPN, traffic that exceeds a certain threshold can move off the default MDT onto a data MDT.

The figure shows the data MDT scenario. The Opaque value used to signal a data MDT consists of two parameters: the VPN ID and the MDT number in the format (vpn-id, MDT# > 0) where vpn-id is a manually configured 7-byte number that uniquely identifies this VPN. The second parameter is the unique data MDT number for this VPN, which is a number greater than zero.

In the scenario, two receivers at PE-North and PE-East are interested in two sources at PE-West. If the source 10.5.200.3 exceeds the threshold on the default MDT, PE-West will issue an MDT Join TLV message over the default MDT MP2MP LSP advising all PE devices that a new data MDT is being created.

Because PE-East has an interested receiver in VRF, it will build a multipoint LSP using P2MP back to PE-West, which will be the root of the tree. PE-North does not have a receiver for 10.5.200.3, therefore it will just cache the Join TLV message.
Configuring MLDPv4-Based MVPN

Configuring Initial MLDP Settings

Perform this task to configure the initial MLDP settings.

**SUMMARY STEPS**

1. enable
2. configure terminal
3. mpls mldp logging notifications
4. mpls mldp forwarding recursive
5. end
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> mpls mldp logging notifications</td>
<td>Enables MLDP logging notifications.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config)# mpls mldp logging notifications</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> mpls mldp forwarding recursive</td>
<td>Enables MLDP recursive forwarding over a P2MP LSP.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config)# mpls mldp forwarding recursive</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> end</td>
<td>Ends the current configuration session and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config)# end</td>
<td></td>
</tr>
</tbody>
</table>

## Configuring an MLDP-Based MVPN

Perform this task to configure an MLDP-based MVPN.

## SUMMARY STEPS

1. enable
2. configure terminal
3. ip multicast-routing
4. ip multicast-routing vrf vrf-name
5. ip vrf vrf-name
6. rd route-distinguisher
7. vpn id oui : vpn-index
8. route target export route-target-ext-community
9. route target import route-target-ext-community
10. mdt preference { mldp | pim }
11. mdt default mpls mldp group-address
12. mdt data mpls mldp number-of-data-mdt
13. mdt data threshold kb/s list access-list
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** | `enable` | Enables privileged EXEC mode.  
- Enter your password if prompted. |
| **Step 2** | `configure terminal` | Enters global configuration mode. |
| **Step 3** | `ip multicast-routing` | Enables IP multicast routing. |
| **Step 4** | `ip multicast-routing vrf vrf-name` | Enables IP multicast routing for the MVPN VRF specified for the `vrf-name` argument. |
| **Step 5** | `ip vrf vrf-name` | Defines a VRF instance and enters VRF configuration mode. |
| **Step 6** | `rd route-distinguisher` | Creates a route distinguisher (RD) (in order to make the VRF functional). Creates the routing and forwarding tables, associates the RD with the VRF instance, and specifies the default RD for a VPN. |
| **Step 7** | `vpn id oui : vpn-index` | Sets or updates the VPN ID on a VRF instance. |
| **Step 8** | `route target export route-target-ext-community` | Creates an export route target extended community for the specified VRF. |
| **Step 9** | `route target import route-target-ext-community` | Creates an import route target extended community for the specified VRF. |
Verifying the Configuration of an MLDP-Based MVPN

Perform this task in privileged EXEC mode to verify the configuration of an MLDP-based MVPN.

SUMMARY STEPS

1. show mpls mldp database
2. show ip pim neighbor [vrf vrf-name] neighbor [interface-type interface-number]
3. show ip mroute [vrf vrf-name] [active [kbps] [interface-type interface-number] | bidirectional | count [terse] | dense | interface-type interface-number | proxy | pruned | sparse | ssm | static | summary] | [group-address [source-address]] [count [terse] | interface-type interface-number | proxy | pruned | summary] | [source-address group-address] [count [terse] | interface-type interface-number | proxy | pruned | summary] | [group-address] active [kbps] [interface-type interface-number | verbose]]
4. show mpls forwarding-table [network {mask | length} | labels label [- label]] | interface interface | next-hop address | lsp-tunnel {tunnel-id} | [vrf vrf-name] | [detail]
5. show adjacency [ip-address] [interface-type interface-number] | null number | port-channel number | sysclock number | vlan number | fcpa number | serial number | connectionid number | [link | ipv4 | mpls] | [detail | encapsulation]
**DETAILED STEPS**

**Step 1**
show mpls mldp database

Enter the `show mpls mldp database` command to display information in the MLDP database. It shows the FEC, the Opaque value of the FEC decoded, and the replication clients associated with it:

**Example:**

```
Device# show mpls mldp database
* Indicates MLDP recursive forwarding is enabled
LSM ID : D3000001 (RNR LSM ID: 8A000002) Type: MP2MP Uptime : 00:04:54
FEC Root : 172.30.20.1
Opaque decoded : [mdt 100:2 0]
Opaque length : 11 bytes
Opaque value : 07 000B 0000010000000100000000
RNR active LSP : (this entry)
Upstream client(s) :
172.30.20.1:0 [Active]
Expires : Never
Path Set ID : 99000001
Out Label (U) : 32
Interface : Ethernet1/0*
Local Label (D): 30
Next Hop : 10.0.1.7
Replication client(s):
MDT (VRF VRF)
Uptime : 00:04:54
Path Set ID : 5000002
Interface : Lspvif0
```

**Step 2**
show ip pim neighbor [vrf vrf-name] neighbor [interface-type interface-number]

Enter the `show ip pim neighbor` command to display PIM adjacencies information:

**Example:**

```
Device# show ip pim vrf VRF neighbor
192.168.10.18 Serial6/0 04:53:19/00:01:18 v2 1 / G
172.30.20.3 Lspvif0 04:52:32/00:01:28 v2 1 / B S P G
172.30.20.2 Lspvif0 04:52:32/00:01:17 v2 1 / B S P G
```

**Step 3**
show ip mroute [vrf vrf-name] [[active [kbps] [interface type number] | bidirectional | count [ terse] | dense | interface type number | proxy | pruned | sparse | ssm | static | summary] | [group-address [source-address]] | [count [ terse] | interface type number | proxy | pruned | summary] | [source-address group-address] | [count [ terse] | interface type number | proxy | pruned | summary] | [group-address] active [kbps] [interface type number | verbose]]

Enter the `show ip mroute` command to display the contents of the multicast routing (mroute) table:

**Example:**

```
Device# show ip mroute vrf VRF 238.1.200.2 10.5.200.3
(10.5.200.3, 238.1.200.2), 04:54:18/00:02:40, flags: sT
Incoming interface: Lspvif0, RPF nbr 172.30.20.2
Outgoing interface list:
Serial6/0, Forward/Sparse-Dense, 04:54:18/00:02:40
```

**Step 4**
show mpls forwarding-table [network {mask | length} | labels label [- label] | interface interface | next-hop address | lsp-tunnel {tunnel-id}] [vrf vrf-name] [detail]

Enter the `show mpls forwarding-table` command to display the contents of the MPLS Label Forwarding Information Base (LFIB):

**Example:**
Device# show mpls forwarding-table | inc 1F000001
105 307 mLDP:1F000001 38468 Se5/0 point2point
208 mLDP:1F000001 38468 Se4/0 point2point
109 307 mLDP:1F000001 34738 Se5/0 point2point
408 mLDP:1F000001 34738 Se6/0 point2point
111 408 mLDP:1F000001 282 Se6/0 point2point
208 mLDP:1F000001 282 Se4/0 point2point

Step 5  
show adjacency [ip-address] [interface-type interface-number | null number | port-channel number | sysclock number | vlan number | fcpa number | serial number] [connectionid number] [link {ipv4 | mpls}] [detail | encapsulation]

Enter the show adjacency command to display adjacency information for the specified LSP-VIF interface:

Example:

Device# show adjacency lspvif0
105 307 mLDP:1F000001 38468 Se5/0 point2point
208 mLDP:1F000001 38468 Se4/0 point2point
109 307 mLDP:1F000001 34738 Se5/0 point2point
408 mLDP:1F000001 34738 Se6/0 point2point
111 408 mLDP:1F000001 282 Se6/0 point2point
208 mLDP:1F000001 282 Se4/0 point2point

MLDPv6-Based MVPN

The MLDPv6 based MVPN configuration enables IPv6 multicast packet delivery using MPLS. This configuration uses MPLS labels to construct default P2MP and MP2MP LSPs. The MPLS replication is used as a forwarding mechanism in the core network.

For MLDP MVPN configuration to work, ensure that the global MPLS MLDP configuration is enabled. These LSPs can be used for transporting IPv6 multicast packets, either in the global table or VPN context.

Figure 43: mLDPv6 topology

![MlDPv6 topology](image)
P2MP and MP2MP Label Switched Paths

MLDP is an application that sets up Multipoint Label Switched Paths (MP LSPs) in MPLS networks without requiring multicast routing protocols in the MPLS core. mLDP constructs the P2MP or MP2MP LSPs without interacting with or relying upon any other multicast tree construction protocol. Using LDP extensions for MP LSPs and Unicast IP routing, mLDP can setup MP LSPs. The two types of MP LSPs that can be setup are Point-to-Multipoint (P2MP) and Multipoint-to-Multipoint (MP2MP) type LSPs.

A P2MP LSP allows traffic from a single root (ingress node) to be delivered to a number of leaves (egress nodes), where each P2MP tree is uniquely identified with a 2-tuple (root node address, P2MP LSP identifier). A P2MP LSP consists of a single root node, zero or more transit nodes, and one or more leaf nodes, where typically root and leaf nodes are PEs and transit nodes are P routers. A P2MP LSP setup is receiver-driven and is signaled using mLDP P2MP FEC, where LSP identifier is represented by the MP Opaque Value element. MP Opaque Value carries information that is known to ingress LSRS and Leaf LSRS, but need not be interpreted by transit LSRS. There can be several MP LSPs rooted at a given ingress node, each with its own identifier.

A MP2MP LSP allows traffic from multiple ingress nodes to be delivered to multiple egress nodes, where a MP2MP tree is uniquely identified with a 2-tuple (root node address, MP2MP LSP identifier). For a MP2MP LSP, all egress nodes, except the sending node, receive a packet sent from an ingress node.

A MP2MP LSP is similar to a P2MP LSP, but each leaf node acts as both an ingress and egress node. To build an MP2MP LSP, you can setup a downstream path and an upstream path so that:

- Downstream path is setup just like a normal P2MP LSP
- Upstream path is setup like a P2P LSP towards the upstream router, but inherits the downstream labels from the downstream P2MP LSP.

**Note**
We recommend that you configure one P2MP MDT tree per prefix. For example, if 500 multicast routes are needed, then you should configure at least 500 P2MP MDT trees.

```
ip vrf vrf-name1
......!
......
mdt data mpls midp 500
```

Packet Flow in mLDP-based Multicast VPN

For each packet coming in, MPLS creates multiple out-labels. Packets from the source network are replicated along the path to the receiver network. The CE1 router sends out the native IP multicast traffic. The PE1 router imposes a label on the incoming multicast packet and replicates the labeled packet towards the MPLS core network. When the packet reaches the core router (P), the packet is replicated with the appropriate labels for the MP2MP default MDT or the P2MP data MDT and transported to all the egress PEs. Once the packet reaches the egress PE, the label is removed and the IP multicast packet is replicated onto the VRF interface.

Realizing a mLDP-based Multicast VPN

There are different ways a Label Switched Path (LSP) built by mLDP can be used depending on the requirement and nature of application such as:

- P2MP LSPs for global table transit Multicast using in-band signaling.
• P2MP/MP2MP LSPs for MVPN based on MI-PMSI or Multidirectional Inclusive Provider Multicast Service Instance (Rosen Draft).

• P2MP/MP2MP LSPs for MVPN based on MS-PMSI or Multidirectional Selective Provider Multicast Service Instance (Partitioned E-LAN).

The router performs the following important functions for the implementation of MLDP:

1. Encapsulating VRF multicast IP packet with GRE/Label and replicating to core interfaces (imposition node).

2. Replicating multicast label packets to different interfaces with different labels (Mid node).

3. Decapsulate and replicate label packets into VRF interfaces (Disposition node).

### Supported mLDP Profiles

<table>
<thead>
<tr>
<th>Profile Name</th>
<th>Supported on mLDPv4</th>
<th>Supported on mLDPv6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile 1 Default MDT - MLDP MP2MP - PIM C-mcast Signaling</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Profile 2 Partitioned MDT - MLDP MP2MP - PIM C-mcast Signaling</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Profile 4 Partitioned MDT - MLDP MP2MP - BGP-AD - PIM C-mcast Signaling</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Profile 5 Partitioned MDT - MLDP P2MP - BGP-AD - PIM C-mcast Signaling</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Profile 6 VRF MLDP - In-band Signaling</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Profile 7 Global MLDP In-band Signaling</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Profile 9 Default MDT - MLDP MP2MP - BGP-AD - PIM C-mcast Signaling</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Profile 12 Default MDT - MLDP P2MP - BGP-AD - BGP C-mcast Signaling</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Profile 13 Default MDT - MLDP MP2MP - BGP-AD - BGP C-mcast Signaling</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Profile 14 Partitioned MDT - MLDP P2MP - BGP-AD - BGP C-mcast Signaling</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Profile Name</td>
<td>Supported on mLDPv4</td>
<td>Supported on mLDPv6</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>Profile 15 Partitioned MDT - MLDP MP2MP - BGP-AD - BGP C-mast Signaling</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Profile 17 Default MDT - MLDP - P2MP - BGP-AD - PIM C-mcast Signaling</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>


## Configuration Rules for Profiles

### Rules for Rosen-mGRE profiles (profiles- 0, 3, 11)
- All profiles require VPNv4 or v6 unicast reachability.
- By default, encapsulate 1400-byte size c-multicast IP packet is supported. To support decap or encapsulate larger packet size, use the `mdt mtu` command.
- Loopback configuration is required. Use the `mdt source loopback0` command. Other loopbacks can be used for different VRFs, but this is not recommended.

### Rules for Rosen-mLDP profiles (profiles- 1, 9, 12, 13, 17)
- mLDP must be globally enabled.
- VPN-id is mandatory for Rosen-mLDP MP2MP profiles.
- Root node must be specified manually. Multiple root nodes can be configured for Root Node Redundancy.
- If only profile 1 is configured, MVPN must be enabled under bgp.
- For BGP-AD profiles, the remote PE address is required.

### Rules for mLDP profiles (profiles- 2, 4, 5, 14, 15)

### Rules for mLDP profiles (profile- 4)
- MVPN must be enabled under bgp, if only profile 2 is configured.
- Support only for static RP for customer RP.

### Rules for inband mLDP profiles (profiles- 6, 7)
- MVPN must be enabled under bgp for vrf-inband profiles.
- Data MDT is not supported.
- Backbone facing interface (BFI) must be enabled on tail PE.
- Source route of SSM must be advertise to tail PE by iBGP.
Initial deployment of an MLDP-based MVPN involves the configuration of a default MDT and one or more data MDTs.

Default MDT Configuration

The following example shows how to configure the default MDT for an MLDP-based MVPN. This configuration is based on the sample topology illustrated in the figure.

This configuration is consistent for every PE device participating in the same VPN ID. The `vpn id 100:2` command replaces the MDT group address used with the mGRE transport method. To provide redundancy, two default MDT trees are statically configured, rooted at P-Central and PE-North. The selection as to which MP2MP tree the default MDT will use at a particular PE device is determined by Interior Gateway Protocol (IGP) metrics. An MP2MP LSP is implicit for the default MDT.
ip pim mpls source Loopback0
ip multicast-routing
ip multicast-routing vrf VRF
ip vrf VRF
rd 100:2
vpn id 100:2
route-target export 200:2
route-target import 200:2
mdt default mpls mldp 172.30.20.1 (P-Central)
mdt default mpls mldp 172.30.20.3 (PE-North)

**PIM Adjacencies**

PIM operates over the LSP-VIF as if it were a regular tunnel interface. That means PIM hellos are exchanged over the LSP-VIF to establish PIM adjacencies over the default MDT. The sample output in this section displays the three PIM adjacencies in VRF of PE-East. The first is the adjacency to the receiver network over serial interface 6/0, and the next two are the adjacencies to PE-West and PE-North over the MP2MP LSP via LSP-VIF interface 0.

```
PE-East# show ip pim vrf VRF neighbor
192.168.10.18 Serial6/0 04:53:19/00:01:18 v2 1 / G
172.30.20.3 Lspvif0 04:52:32/00:01:28 v2 1 / B S P G
172.30.20.2 Lspvif0 04:52:32/00:01:17 v2 1 / B S P G
```

The output from the `show ip mroute` command also shows the (S, G) entry for VRF. The stream 238.1.200.2 has the Reverse Path Forwarding (RPF) interface of LSP-VIF interface 0 and the neighbor 172.30.20.2, which is PE-West.

```
PE-East# show ip mroute vrf VRF 238.1.200.2 10.5.200.3
(10.5.200.3, 238.1.200.2), 04:54:18/00:02:40, flags: sT
Incoming interface: Lspvif0, RPF nbr 172.30.20.2
Outgoing interface list:
Serial6/0, Forward/Sparse-Dense, 04:54:18/00:02:40
```

**MLDP Database Entry--PE-East**

The sample output in this section displays the database entries for the MP2MP trees supporting the default MDT at PE-East. The database is searched by Opaque value MDT 100:2, which results in information for two MP2MP trees (one for each root) being returned. Both trees have different system IDs (2E000001, F2000005) and use the same Opaque value ([mdt 100:2 0]), but with different roots. The last 0 in the Opaque value indicates this tree is a default MDT. Entry 79000004 is the primary MP2MP tree, therefore PE-East will transmit all source multicast traffic on this LSP, and B2000006 will be the backup root. Note that interface LSP-VIF interface 0 represents both MP2MP LSPs. The Local Label (D) is the downstream label allocated by PE-East for this tree. In other words, traffic from the root will be received with either label 408 (Primary Tree) or 407 (Backup Tree). The Out Label (U) is the label that PE-East will use to send traffic into the tree; upstream towards the root, either 105 for the Primary Tree or 108 for the Backup Tree. Both these labels were received from P-Central.

```
PE-East# show mpls mldp database opaque_type mdt 100:2
* Indicates MLDP recursive forwarding is enabled
LSM ID : 79000004 (RNR LSM ID: 8A000002) Type: MP2MP Uptime : 00:04:54
  FEC Root : 172.30.20.1
  Opaque decoded : [mdt 100:2 0]
  Opaque length : 11 bytes
  Opaque value : 07 000B 00000100000000010000000
  RNR active LSP : (this entry)
```
Upstream client(s):
172.30.20.1:0 [Active]
Expires : Never Path Set ID : 99000001
Out Label (U) : 32 Interface : Ethernet1/0*
Local Label (D): 30 Next Hop : 10.0.1.7
Replication client(s):
MDT (VRF VRF)
Uptime : 00:04:54 Path Set ID : 5000002
Interface : Lspvif0
LSM ID : 79000005 (RNR LSM ID: 8A000003) Type: MP2MP Uptime : 00:04:54
FEC Root : 172.30.20.3
Opaque decoded : [mdt 100:2 0]
Opaque length : 11 bytes
Opaque value : 07 000B 000001000000100000001
RNR active LSP : (this entry)
Upstream client(s):
172.30.20.1:0 [Active]
Expires : Never Path Set ID : 99000002
Out Label (U) : 32 Interface : Ethernet1/0*
Local Label (D): 30 Next Hop : 10.0.1.7
Replication client(s):
MDT (VRF VRF)
Uptime : 00:04:54 Path Set ID : 5000003
Interface : Lspvif0

Label Forwarding Entry--P-Central (Root 1)

The sample output shown in this section displays the VRF (MDT 100:2) MLDP database entry 1F000001 for the primary MP2MP LSP, which is P-Central. Because the local device P-Central is the root, there is no upstream peer ID, therefore no labels are allocated locally. However there are three replication clients, representing each of the three PE devices: PE-North, PE-West, and PE-East. These replication clients are the downstream nodes of the MP2MP LSP. These clients receive multipoint replicated traffic.

In the replication entry looking from the perspective of the root, there are two types of labels:

- **Out label (D)**--These are labels received from remote peers that are downstream to the root (remember traffic flows downstream away from the root).

- **Local label (U)**--These are labels provided by P-Central to its neighbors to be used as upstream labels (sending traffic to the root). It is easy to identify these labels as they all start in the 100 range, which we have configured for P-Central to use. P-Central sends these labels out when it receives a FEC with the type as MP2MP Down.

From the labels received and sent in the replication entries, the Label Forwarding Information Base (LFIB) is created. The LFIB has one entry per upstream path and one entry per downstream path. In this case because P-Central is the root, there are only upstream entries in the LFIB that have been merged with the corresponding downstream labels. For example, label 105 is the label P-Central sent to PE-East to send source traffic upstream. Traffic received from PE-East will then be replicated using the downstream labels 307 to PE-West and 208 to PE-North.

```
P-Central# show mpls mldp database opaque_type mdt 100:2
LSM ID : 79000006 (RNR LSM ID: 1F000001) Type: MP2MP Uptime : 00:04:54
FEC Root : 172.30.20.1
Opaque decoded : [mdt 100:2 0]
Opaque length : 11 bytes
Opaque value : 07 000B 000001000000100000001
RNR active LSP : (this entry)
Upstream client(s) : None
Replication client(s):
172.3.20.2:0
```

---

**MLDP-Based MVPN**

**Label Forwarding Entry--P-Central (Root 1)**
The sample output shown in this section displays the entry on P-Central for the MP2MP LSP rooted at PE-North (backup root). In this tree P-Central is a branch of the tree, not a root, therefore there are some minor differences to note:

- The upstream peer ID is PE-North, therefore P-Central has allocated label 104 in the downstream direction towards PE-North and subsequently PE-North has responded with an upstream label of 313.

- Two replication entries representing PE-East and PE-West are displayed.

- The merged LFIB shows three entries:
  - One downstream entry label 104 receiving traffic from Root 2 (PE-North), which is then directed further downstream using labels 207 PE-West and 407 PE-East.
  - Two upstream entries 108 and 115 receiving traffic from the leaves and directing it either downstream 207, 407 or upstream using label 313.

```bash
Central_P# show mpls forwarding-table | inc 1F000001
105 307 mLDP:1F000001 38468 Se5/0 point2point
208 mLDP:1F000001 38468 Se4/0 point2point
109 307 mLDP:1F000001 34738 Se5/0 point2point
408 mLDP:1F000001 34738 Se6/0 point2point
111 408 mLDP:1F000001 282 Se6/0 point2point
208 mLDP:1F000001 282 Se4/0 point2point

Central_P# show mpls mldp database opaque_type mdt 100:2
LSM ID : E6000004
Uptime : 00:42:03
Tree type : MP2MP
FEC Root : 172.30.20.3
Opaque length : 14 bytes
Opaque value : 07000B00 01000000 00020000 00009C
Opaque decoded : [mdt 100:2 0]
Upstream peer ID : 172.30.20.3, Label local (D): 104 remote (U): 313 active
Path Set ID : 48000003
Replication client(s):
172.30.20.2:0 uptime: 00:42:03 Path Set ID: CF000004
remote label (D): 207 local label (U): 115
nhop: 172.30.10.2 intrf: Serial14/0
172.30.20.4:0 uptime: 00:41:44 Path Set ID: 5800000E
remote label (D): 407 local label (U): 108
nhop: 172.30.10.10 intrf: Serial16/0
```

```bash
Central_P# show mpls forwarding-table | inc E6000004
104 207 mLDP:E6000004 251228 Se4/0 point2point
407 mLDP:E6000004 251334 Se6/0 point2point
108 207 mLDP:E6000004 0 Se4/0 point2point
313 mLDP:E6000004 0 Se5/0 point2point
115 313 mLDP:E6000004 0 Se5/0 point2point
407 mLDP:E6000004 0 Se6/0 point2point
```
Data MDT Configuration

The following example shows how to configure the data MDT for an MLDP-based MVPN. This configuration is based on the sample topology illustrated in the figure.

Figure 45: Data MDT Example

The sample output in this section displays the data MDT configuration for all the PE devices. The `mdt data` commands are the only additional commands necessary. The first `mdt data` command allows a maximum of 60 data MDTs to be created, and the second `mdt data` command sets the threshold. If the number of data MDTs exceeds 60, then the data MDTs will be reused in the same way as they are for the mGRE tunnel method (the one with the lowest reference count).

```
ip pim vrf VRF mpls source Loopback0
!
ip vrf VRF
  rd 100:2
  vpn id 100:2
  route-target export 200:2
  route-target import 200:2
  mdt default mpls mldp 172.30.20.1 (P-Central)
  mdt default mpls mldp 172.30.20.3 (PE-North)
  mdt data mpls mldp 60
  mdt data threshold 1
```
VRF mroute Table--PE-West

The sample output in this section displays the VRF mroute table on PE-West before the high-bandwidth source exceeds the threshold. At this point there are two streams, representing each of the two VPN sources at PE-West, on a single MP2MP LSP (System ID D8000000). The LSP represents the default MDT accessed via LSP-VIF interface 0.

```
PE-West# show ip mroute vrf VRF verbose
... ...
(10.5.200.2, 238.1.200.1), 00:00:25/00:03:29, flags: sT
  Incoming interface: Serial6/0, RPF nbr 192.168.10.6
  Outgoing interface list:
    Lspvif0, LSM MDT: D8000000 (default),Forward/Sparse-Dense,
...
(10.5.200.3, 238.1.200.2), 00:11:14/00:02:48, flags: sT
  Incoming interface: Serial6/0, RPF nbr 192.168.10.6
  Outgoing interface list:
    Lspvif0, LSM MDT: D8000000 (default),Forward/Sparse-Dense,
... ...
```

The sample output in this section displays the output after the source transmission exceeds the threshold. PE-West sends an MDT Join TLV message to signal the creation of a data MDT. In this case, the data MDT number is 1, therefore PE-East will send a label mapping message back to PE-West with a FEC TLV containing root=PE-West, Opaque value=(mdt vpn-id 1). The System ID is now changed to 4E000003 signaling a different LSP; however, the LSP-VIF is still LSP-VIF interface 0. The (S, G) entry also has the “y” flag set indicating this stream has switched to a data MDT.

```
PE-West# show ip mroute vrf VRF 10.5.200.3 238.1.200.2 verbose
... ...
(10.5.200.3, 238.1.200.2), 00:00:08/00:03:27, flags: sTy
  Incoming interface: Serial6/0, RPF nbr 192.168.10.6
   MDT TX nr: 1 LSM-ID 4E000003
   Outgoing interface list:
    Lspvif0, LSM MDT: 4E000003 (data) Forward/Sparse-Dense,
... ...
```

LSP-VIF Adjacencies--PE-West

For the interface LSP-VIF, each virtual circuit represents a unique multipoint LSP forwarding instance. The correct adjacency is selected when sending the multicast packet. The sample output in this section displays the application of that concept on PE-West. There is a single LSP-VIF interface 0 interface, but it has three adjacencies as follows:

- 4E000003 is the single data MDT created for (10.5.200.3, 238.1.200.2)
- 58000000 is the default MDT (backup root)
- D8000000 is the default MDT (primary root)

```
PE-West# show adjacency lspvif 0
Protocol Interface Address
... ...
```
**MLDP Database Entries**

The sample output in this section displays the MLDP entry for the data MDT (4E0000003) on the ingress device PE-West. The following points about this entry should be noted:

- The tree type is P2MP with PE-West (172.30.20.2) as the root.
- The Opaque value is `[mdt 100:2 1]` denoting the first data MDT.
- There are no labels allocated as it is the root.
- There are two replication client entries on this tree.
- Label 112 will be used to send the traffic downstream towards PE-East (via P-Central).
- The MDT entry is an internal construct.

PE-West# `show mpls mldp database id 4E0000003`

<table>
<thead>
<tr>
<th>LSM ID</th>
<th>(RNR LSM ID: 8A0000002)</th>
<th>Type: P2MP</th>
<th>Uptime : 00:04:54</th>
</tr>
</thead>
<tbody>
<tr>
<td>FEC Root</td>
<td>172.30.20.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Opaque decoded</td>
<td>[mdt 100:2 1]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Opaque length</td>
<td>11 bytes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Opaque value</td>
<td>07 0008 0000010000000100000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RNR active LSP</td>
<td>(this entry)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Upstream client(s):</td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Replication client(s):</td>
<td>MDT (VRF VRF)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Uptime</td>
<td>00:04:54</td>
<td>Path Set ID : 5000002</td>
<td></td>
</tr>
<tr>
<td>Interface</td>
<td>Lspvif0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

172.30.20.1:0

| Uptime          | 01:41:43                 | Path Set ID : D9000007 |
| Out label (D)   | 27                       | Interface : Serial4/0  |
| Local label (U) | 112                      | Next Hop : 172.30.10.1 |

The sample output in this section displays the database entry for the data MDT on PE-East, the egress device. Also shown is the MDT Join TLV message that was sent from PE-West over the default MDT. The MDT Join TLV message contains all the necessary information to allow PE-East to create a label mapping message P2MP LSP back to the root of PE-West. Label 414 will be used by P-Central to send traffic to PE-East.

*Feb 19 04:43:24.039: PIM(1): MDT join TLV received for (10.5.200.3,238.1.200.2)*

*Feb 19 04:43:24.039: MLDP: LDP root 172.30.20.2 added*

*Feb 19 04:43:24.039: MLDP: [mdt 100:2 1] label mapping msg sent to 172.30.20.1:0*

PE-East# `show mpls mldp database opaque_type mdt 100:2 1`

<table>
<thead>
<tr>
<th>LSM ID</th>
<th>(RNR LSM ID: 8A0000002)</th>
<th>Type: P2MP</th>
<th>Uptime : 00:04:54</th>
</tr>
</thead>
<tbody>
<tr>
<td>FEC Root</td>
<td>172.30.20.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Opaque decoded</td>
<td>[mdt 100:2 1]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Opaque length</td>
<td>11 bytes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Opaque value</td>
<td>07 0008 0000010000000100000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RNR active LSP</td>
<td>(this entry)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Upstream client(s):</td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Replication client(s):</td>
<td>MDT (VRF VRF)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
LFIB Entry for the Data MDT

The sample output in this section displays the LFIB entry for the data MDT as it passes through P-Central and PE-East. The Tunnel ID used for the LSP is the Opaque value [mdt 100:2 1].

P-Central# show mpls for label 112
Label   Label or Tunnel Id Switched interface
111     [mdt 100:2 1] 2993584 Se6/0 point2point

PE-East# show mpls for label 400
Label   Label or Tunnel Id Switched interface
414     [mdt 100:2 1][V] 3297312 aggregate/green
CHAPTER 14

MVPN mLDP Partitioned MDT

The MVPN mLDP partitioned MDT feature uses Upstream Multicast Hop-Provider Multicast Service Interface (UMS-PMSI), a subset of provider edge routers (PEs) to transmit data to other PEs; similar to the usage of multiple selective-PMSI (S-PMSI) by data multicast distribution tree (MDT). In the partitioned MDT approach, egress PE routers that have interested receivers for traffic from a particular ingress PE joins a point-to-point (P2P) connection rooted at that ingress PE. This makes the number of ingress PE routers in a network to be low resulting in a limited number of trees in the core.

- Finding Feature Information, on page 287
- Prerequisites for MVPN mLDP Partitioned MDT, on page 287
- Restrictions for MVPN mLDP Partitioned MDT, on page 287
- Information About MVPN mLDP Partitioned MDT, on page 288
- How to Configure MVPN mLDP Partitioned MDT, on page 289
- Configuration Examples for MVPN mLDP Partitioned MDT, on page 291

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 MVPN mLDP Partitioned MDT

MVPN BGP auto discovery should be configured.

Restrictions for MVPN mLDP Partitioned MDT

- PIM Dense mode (except for Auto-RP) and PIM-Bidir in the VRF are not supported.
- BGP multicast signaling is supported and PIM signaling is not supported.
Only point-to-multipoint (P2MP) mLDP label switch path is supported.

Same VRF (for which mLDP in-band signaling is configured) needs to be configured on IPv4 and IPv6 address families.

Rosen mLDP recursive FEC is not supported. Partitioned MDT is applicable to inter-AS VPN (Inter AS option B and option C are not supported).

mLDP filtering is not supported.

Only interface-based strict RPF is supported with partitioned MDT.

Information About MVPN mLDP Partitioned MDT

Overview of MVPN mLDP Partitioned MDT

MVPN allows a service provider to configure and support multicast traffic in an MPLS VPN environment. This type supports routing and forwarding of multicast packets for each individual VPN routing and forwarding (VRF) instance, and it also provides a mechanism to transport VPN multicast packets across the service provider backbone. In the MLDP case, the regular label switch path forwarding is used, so core does not need to run PIM protocol. In this scenario, the c-packets are encapsulated in the MPLS labels and forwarding is based on the MPLS Label Switched Paths (LSPs).

The MVPN mLDP service allows you to build a Protocol Independent Multicast (PIM) domain that has sources and receivers located in different sites. To provide Layer 3 multicast services to customers with multiple distributed sites, service providers look for a secure and scalable mechanism to transmit customer multicast traffic across the provider network. Multicast VPN (MVPN) provides such services over a shared service provider backbone, using native multicast technology similar to BGP/MPLS VPN.

MVPN emulates MPLS VPN technology in its adoption of the multicast domain (MD) concept, in which provider edge (PE) routers establish virtual PIM neighbor connections with other PE routers that are connected to the same customer VPN. These PE routers thereby form a secure, virtual multicast domain over the provider network. Multicast traffic is then transmitted across the core network from one site to another, as if the traffic were going through a dedicated provider network.

Separate multicast routing and forwarding tables are maintained for each VPN routing and forwarding (VRF) instance, with traffic being sent through VPN tunnels across the service provider backbone. In the Rosen MVPN mLDP solution, a multipoint-to-multipoint (MP2MP) default MDT is setup to carry control plane and data traffic. A disadvantage with this solution is that all PE routers that are part of the MVPN need to join this default MDT tree. Setting up a MP2MP tree between all PE routers of a MVPN is equivalent to creating N P2MP trees rooted at each PE (Where N is the number of PE routers). In an Inter-AS (Option A) solution this problem is exacerbated since all PE routers across all AS’ es need to join the default MDT. Another disadvantage of this solution is that any packet sent through a default MDT reaches all the PE routers even if there is no requirement.

In the partitioned MDT approach, only those egress PE routers that receive traffic requests from a particular ingress PE join the PMSI configured at that ingress PE. This makes the number of ingress PE routers in a network to be low resulting in a limited number of trees in the core.
How to Configure MVPN mLDP Partitioned MDT

Configuring MVPN mLDP Partitioned MDT

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `ip multicast-routing vrf vrf-name`
4. `ip vrf vrf-name`
5. `rd route-distinguisher`
6. `route target export route-target-ext-community`
7. `route target import route-target-ext-community`
8. `mdt strict-rpf interface`
9. `mdt partitioned mldp p2mp`
10. `mdt auto-discovery mldp [inter-as]`
11. `exit`
12. `show ip pim mdt`
13. `show ip pim vrf mdt [send | receive]`
14. `show ip multicast mpls vif`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><code>enable</code></td>
<td><em>Enter your password if prompted.</em></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device&gt; enable</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device# configure terminal</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Enables IP multicast routing for the MVPN VRF specified for the <code>vrf-name</code> argument.</td>
</tr>
<tr>
<td><code>ip multicast-routing vrf vrf-name</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config)# ip multicast-routing vrf VRF</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Defines a VRF instance and enters VRF configuration mode.</td>
</tr>
<tr>
<td><code>ip vrf vrf-name</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-vrf)# ip vrf VRF</td>
</tr>
</tbody>
</table>
### Command or Action

#### Step 5
**rd route-distinguisher**  
**Example:**  
Device(config-vrf)# rd 50:11  
**Purpose:** Creates a route distinguisher (RD) (in order to make the VRF functional).  
- Creates the routing and forwarding tables, associates the RD with the VRF instance, and specifies the default RD for a VPN.

#### Step 6
**route target export route-target-ext-community**  
**Example:**  
Device(config-vrf)# route target export 100:100  
**Purpose:** Creates an export route target extended community for the specified VRF.

#### Step 7
**route target import route-target-ext-community**  
**Example:**  
Device(config-vrf)# route target import 100:100  
**Purpose:** Creates an import route target extended community for the specified VRF.

#### Step 8
**mdt strict-rpf interface**  
**Example:**  
Device(config-vrf)# mdt strict-rpf interface  
**Purpose:** Creates per-PE LSPVIF interface to implement strict-RPF check.

#### Step 9
**mdt partitioned mldp p2mp**  
**Example:**  
Device(config-vrf)# mdt partitioned mldp p2mp  
**Purpose:** Configures partitioned MDT.  
- If both IPv4 and IPv6 address-families need to be configured for partitioned MDT, configure this command under both the VRF address-family sub-modes.

#### Step 10
**mdt auto-discovery mldp [inter-as]**  
**Example:**  
Device(config-vrf)# mdt auto-discovery mldp inter-as  
**Purpose:** Enables inter-AS operation with BGP A-D.

#### Step 11
**exit**  
**Example:**  
Device(config-vrf)# exit  
**Purpose:** Exits the VRF configuration mode and returns to privileged EXEC mode.

#### Step 12
**show ip pim mdt**  
**Example:**  
Device# show ip pim mdt  
**Purpose:** Displays information on wildcard S-PMXI A-D route.
**Configuration Examples for MVPN mLDP Partitioned MDT**

**Example: MVPN mLDP Partitioned MDT**

```conf
! vrf definition cu1
   rd 1:1
   vpn id 1:1
!
   address-family ipv4
   mdt auto-discovery mldp
   mdt strict-rpf interface
   mdt partitioned mldp p2mp
   mdt data mpls mldp 1
   mdt overlay use-bgp
   route-target export 1:1
   route-target import 1:1
   exit-address-family
!
   ip multicast-routing distributed
   ip multicast-routing vrf cu1 distributed
!
   mpls label protocol ldp
   mpls ldp session protection
   mpls ldp igp sync holddown 10000
   mpls ldp discovery targeted-hello accept
   no mpls mldp forwarding recursive
   mpls mldp path traffic-eng
   mpls traffic-eng tunnels
   mpls traffic-eng auto-tunnel backup nhop-only
   mpls traffic-eng auto-tunnel primary onehop
!
   redundancy
   mode sso
   bridge-domain 1
!
!
interface Loopback0
   ip address 10.10.10.1 255.255.255.255
   ip ospf 100 area 0
```
load-interval 30
!
interface Loopback1
  vrf forwarding cul
  ip address 11.11.11.1 255.255.255.0
  ip pim sparse-mode
  load-interval 30
!
interface GigabitEthernet0/3/0
  ip address 13.0.0.1 255.255.255.0
  ip ospf 100 area 0
  negotiation auto
  mpls ip
  mpls label protocol ldp
  mpls traffic-eng tunnels
  cdp enable
  ip rsvp bandwidth
!
interface GigabitEthernet0/3/4
  no ip address
  negotiation auto
  service instance 1 ethernet
  encapsulation dot1q 1
  rewrite ingress tag pop 1 symmetric
  bridge-domain 1
!
interface GigabitEthernet0/4/1
  ip address 12.0.0.1 255.255.255.0
  ip ospf 100 area 0
  load-interval 30
  negotiation auto
  mpls ip
  mpls label protocol ldp
  mpls traffic-eng tunnels
  cdp enable
  ip rsvp bandwidth
!
interface BD1
  vrf forwarding cul
  ip address 11.0.1.1 255.255.255.0
  ip pim sparse-mode
  load-interval 30
!
routing ospf 100
  router-id 10.10.10.1
  fast-reroute per-prefix enable prefix-priority low
  timers throttle spf 50 200 5000
  timers throttle lsa 50 200 5000
  timers lsa arrival 100
  network 1.1.1.1 0.0.0.0 area 0
  mpls traffic-eng router-id Loopback0
  mpls traffic-eng area 0
  mpls traffic-eng multicast-intact
!
routing bgp 100
  bgp log-neighbor-changes
  neighbor 10.10.10.2 remote-as 100
  neighbor 10.10.10.2 update-source Loopback0
  neighbor 10.10.10.3 remote-as 100
  neighbor 10.10.10.3 update-source Loopback0
!
address-family ipv4
  redistribute connected
neighbor 10.10.10.2 activate
neighbor 10.10.10.2 send-community extended
neighbor 10.10.10.3 activate
neighbor 10.10.10.3 send-community extended
exit-address-family

address-family ipv4 mvpn
neighbor 10.10.10.2 activate
neighbor 10.10.10.2 send-community extended
neighbor 10.10.10.3 activate
neighbor 10.10.10.3 send-community extended
exit-address-family

address-family vpnv4
neighbor 10.10.10.2 activate
neighbor 10.10.10.2 send-community extended
neighbor 10.10.10.3 activate
neighbor 10.10.10.3 send-community extended
exit-address-family

address-family ipv4 vrf cu1
redistribute connected
exit-address-family

ip forward-protocol nd

no ip http server
ox ip http secure-server
ip pim vrf cu1 rp-address 11.11.11.1

Example: MVPN mLDP Partitioned MDT
Example: MVPN mLDP Partitioned MDT
MLDP In-Band Signaling/Transit Mode

This module contains information for configuring Multicast Label Distribution Protocol (MLDP) in-band signaling to enable the MLDP core to create (S,G) or (*,G) state without using out-of-band signaling such as Border Gateway protocol (BGP) or Protocol Independent Multicast (PIM).

- Finding Feature Information, on page 295
- Restrictions for MLDP In-Band Signaling, on page 295
- Information About MLDP In-Band Signaling/Transit Mode, on page 295
- How to Configure MLDP In-Band Signaling/Transit Mode, on page 296
- Configuration Examples for MLDP In-Band Signaling/Transit Mode, on page 297

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 MLDP In-Band Signaling

- MLDP in-band signaling supports SOURCE-SPECIFIC MULTICAST (SSM) multicast traffic only.

- MLDP in-band signaling is not supported in the same VRF for which Rosen Model MLDP-based MVPN or GRE-based MVPN is configured.

Information About MLDP In-Band Signaling/Transit Mode

MLDP In-Band Signaling/Transit Mode

Multicast Label Distribution Protocol (MLDP)-supported multicast VPN (MVPN) allows VPN multicast streams to be aggregated over a VPN-specific tree. No customer state is created in the MLDP core; there is
only state for default and data multicast distribution trees (MDTs). In certain scenarios, the state created for VPN streams is limited and does not appear to be a risk or limiting factor. In these scenarios, MLDP can build in-band MDTs that are transit Label Switched Paths (LSPs).

Trees used in a VPN space are MDTs. Trees used in the global table are transit point-to-multipoint (P2MP) or multipoint-to-multipoint (MP2MP) LSPs. In both cases, a single multicast stream (VPN or not) is associated with a single LSP in the MPLS core. The stream information is encoded in the Forwarding Equivalence Class (FEC) of the LSP. This is in-band signaling.

MLDP in-band signaling uses access control lists (ACLs) with the range of the multicast (S, G) to be transported by the MLDP LSP. Each multicast channel (S, G) maps, one-to-one, to each tree in the in-band tree. The (S,G) join is registered in the Multicast Routing Information Base (MRIB), which is a client of MLDP. Each MLDP LSP is identified by the FEC of [(S,G) + RD], where RD is the Route Distinguisher (RD) obtained from BGP. This differs from MLDP-based MVPN, where the identity is in a FEC of [MDT #, VPN ID, Tree #]).

The ingress Provider Edge (PE) device uses the FEC to decode the stream information and associate the multicast stream with the LSP (in the FEC). This service model is only applicable for transporting Protocol Independent Multicast (PIM) source-specific multicast (SSM) traffic. There is no need to run PIM over the LSP because the stream signaling is done in-band.

The MLDP In-Band Signaling/Transit Mode feature is supported on IPv4 and IPv6 networks. MLDP in-band signaling and MLDP-based MVPN cannot be supported in the same VRF.

How to Configure MLDP In-Band Signaling/Transit Mode

Enabling In-Band Signaling on a PE Device

Before you begin

- VRF instances for in-band signaling must be configured.
- Access control lists (ACLs) for controlling streams must be configured.

SUMMARY STEPS

1. enable
2. configure terminal
3. ip multicast [vrf vrf] mpls mldp [range acl]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>enable</td>
<td></td>
</tr>
<tr>
<td>Example: Device&gt; enable</td>
<td>Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>configure terminal</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
</tbody>
</table>
MLDP In-Band Signaling/Transit Mode

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Device# configure terminal</code></td>
<td>Brings up the MLDP MRIB process and registers MLDP with the MRIB.</td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>ip multicast [vrf vrf] mpls mldp [range acl]</code></td>
<td>Brings up the MLDP MRIB process and registers MLDP with the MRIB.</td>
</tr>
<tr>
<td><strong>Example:</strong> <code>Device (config)# ip multicast vrf vrf1 mpls mldp</code></td>
<td>• To enable in-band signaling globally, use this command without the <code>vrf vrf</code> keyword and argument combination.</td>
</tr>
<tr>
<td></td>
<td>• To identify streams for in-band signaling, use this command with the <code>range</code> keyword on the egress PE.</td>
</tr>
</tbody>
</table>

### Configuration Examples for MLDP In-Band Signaling/Transit Mode

#### Example: In-Band Signaling

```plaintext
vrf definition cu1
  rd 1:1
  vpn id 1:1
  
  address-family ipv4
    route-target export 1:1
    route-target import 1:1
    exit-address-family
  
  ip multicast-routing distributed
  ip multicast-routing vrf cu1 distributed
  
  ip multicast mpls mldp
  ip multicast mpls traffic-eng
  ip multicast vrf cu1 mpls mldp
  
  mpls label protocol ldp
  mpls ldp session protection
  mpls ldp igp sync holddown 10000
  mpls ldp discovery targeted-hello accept
  no mpls mldp forwarding recursive
  
  mpls mldp path traffic-eng
  mpls traffic-eng tunnels
  mpls traffic-eng auto-tunnel backup nhop-only
  mpls traffic-eng auto-tunnel primary onehop
  
  redundancy
  mode sso

interface Loopback0
  ip address 10.10.10.3 255.255.255.255
  ip ospf 100 area 0
  load-interval 30
```

IP Multicast: Multicast Configuration Guide, Cisco IOS XE Release 3S (Cisco ASR 900 Series)
Example: In-Band Signaling

interface Loopback1
  vrf forwarding cu1
  ip address 33.33.33.1 255.255.255.0
  ip pim sparse-mode
  load-interval 30

interface GigabitEthernet0/2/0
  ip address 13.0.0.2 255.255.255.0
  negotiation auto
  mpls ip
  mpls label protocol ldp
  mpls traffic-eng tunnels
cdp enable
  ip rsvp bandwidth

interface GigabitEthernet0/2/1
  no ip address
  negotiation auto
  service instance 1 ethernet
    encapsulation dot1q 1
    rewrite ingress tag pop 1 symmetric
    bridge-domain 1

interface BDI1
  vrf forwarding cu1
  ip address 31.0.1.1 255.255.0.0
  ip pim sparse-mode
  ip igmp version 3
  load-interval 30

router ospf 100
  router-id 10.10.10.3
  timers throttle spf 50 200 5000
  timers throttle lsa 50 200 5000
  timers lsa arrival 100
  network 1.1.1.1 0.0.0.0 area 0
  mpls traffic-eng router-id Loopback0
  mpls traffic-eng area 0
  mpls traffic-eng multicast-intact

router bgp 100
  bgp log-neighbor-changes
  neighbor 10.10.10.1 remote-as 100
  neighbor 10.10.10.1 update-source Loopback0
  neighbor 10.10.10.2 remote-as 100
  neighbor 10.10.10.2 update-source Loopback0

  address-family ipv4
    redistribute connected
    neighbor 10.10.10.1 activate
    neighbor 10.10.10.1 send-community extended
    neighbor 10.10.10.2 activate
    neighbor 10.10.10.2 send-community extended
    exit-address-family

  address-family ipv4 mvpn
    neighbor 10.10.10.1 activate
    neighbor 10.10.10.1 send-community extended
    neighbor 10.10.10.2 activate
    neighbor 10.10.10.2 send-community extended
    exit-address-family
address-family vpnv4
  neighbor 10.10.10.1 activate
  neighbor 10.10.10.1 send-community extended
  neighbor 10.10.10.2 activate
  neighbor 10.10.10.2 send-community extended
exit-address-family
!
address-family ipv4 vrf cu1
  redistribute connected
exit-address-family
!
address-family ipv4 vrf cu10
  redistribute connected
!
ip pim ssm range SSM-range
ip pim mpls source Loopback0
!
ip pim vrf cu1 ssm range SSM-range
!
ip access-list standard SSM-range
  permit 226.0.0.0 0.255.255.255