Multicast Configuration Guide

First Published: 2025-09-15

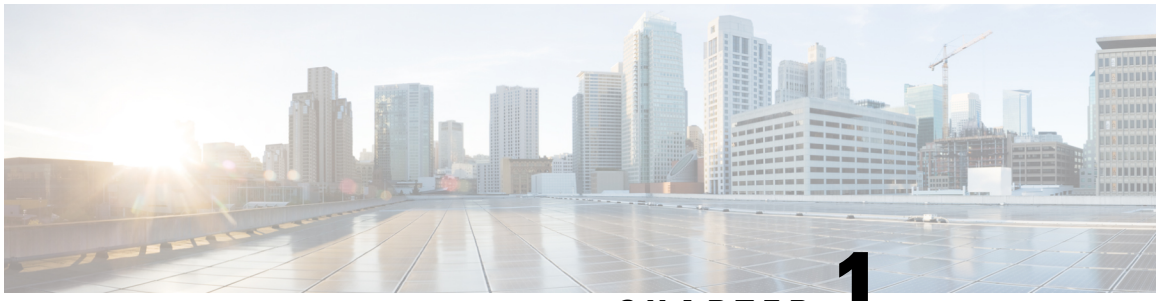
Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Read Me First

Only supported features are documented. To confirm or clarify all the supported features for a platform, go to Cisco Feature Navigator.

Read Me First

CONTENTS



CHAPTER 1

Basic IP Multicast Routing

Feature history for IP multicast routing

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature name and description	Supported platform
Cisco IOS XE 17.18.1	IP multicast routing: IP multicast routing is a method used in IP networks to efficiently deliver data from one source to multiple destinations simultaneously.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

Understand IP multicast routing

IP multicast routing is a method used in IP networks to efficiently deliver data from one source to multiple destinations simultaneously. Instead of sending separate copies of the same data to each recipient, multicast routing allows the source to send a single stream of data that is then distributed by routers only to those networks where interested receivers are present. This optimizes bandwidth usage and reduces network load.

IP multicast routing communication

IP unicast involves a source IP host sending packets to a specific destination IP host. In IP unicast, the destination address in an IP packet corresponds to a single, unique host within the IP network. These IP packets are forwarded across the network from the source to the destination host by devices. Devices use a unicast

routing table to make forwarding decisions at each point on a path between the source and destination, using the IP destination address in the packet.

At the other end of the IP communication spectrum is an IP broadcast, where a source host sends packets to all hosts on a network segment. The destination address of an IP broadcast packet has the host portion of the destination IP address set to all ones and the network portion set to the address of the subnet. IP hosts, including devices, understand that packets, which contain an IP broadcast address as the destination address, are addressed to all IP hosts on the subnet. Devices do not forward IP broadcast packets unless specifically configured to do so, limiting IP broadcast communication to a local subnet.

IP multicasting falls between IP unicast and IP broadcast communication. IP multicast communication enables a host to send IP packets to a group of hosts anywhere within the IP network. To send information to a specific group, IP multicast communication uses a special form of IP destination address called an IP multicast group address. The IP destination address field of the packet specifies the IP multicast group address.

To multicast IP information, Layer 3 switches and devices must forward an incoming IP packet to all output interfaces that lead to members of the IP multicast group.

We tend to think of IP multicasting and video conferencing as the same thing. Video conferencing is often the first application to use IP multicast; however, it is just one of many applications that enhance a company's business model. Multimedia conferencing, data replication, real-time data multicasts, and simulation applications enhance productivity.

Role of IP multicast in information delivery

IP multicast uses a bandwidth-conserving approach to reduce traffic by delivering a single stream of information simultaneously to potentially thousands of businesses and homes. You can use multicast for video conferencing, corporate communications, distance learning, software distribution, stock quoting, and news sharing.

IP multicast routing allows a host to send packets to multiple receivers in a network using an IP multicast group address. The sending host places the multicast group address in the IP destination address field, and multicast routers forward packets to interfaces leading to group members. Any host can send to a group, even if it is not a member. However, only the members of a group receive the message. Controlling the transmission rate to a multicast group is not supported.

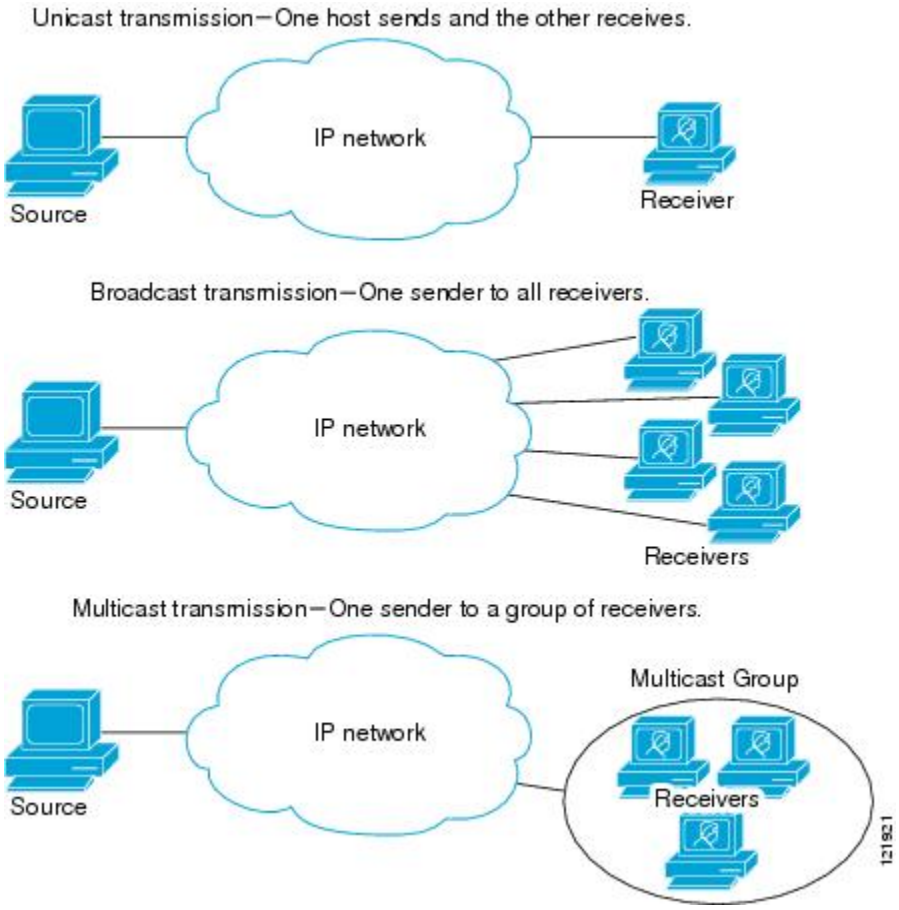
IP multicast delivery modes differ for the receiver hosts and not for the source hosts. A source host sends IP multicast packets using its own IP address as the source address and a group address as the destination address.

Multicast group transmission scheme

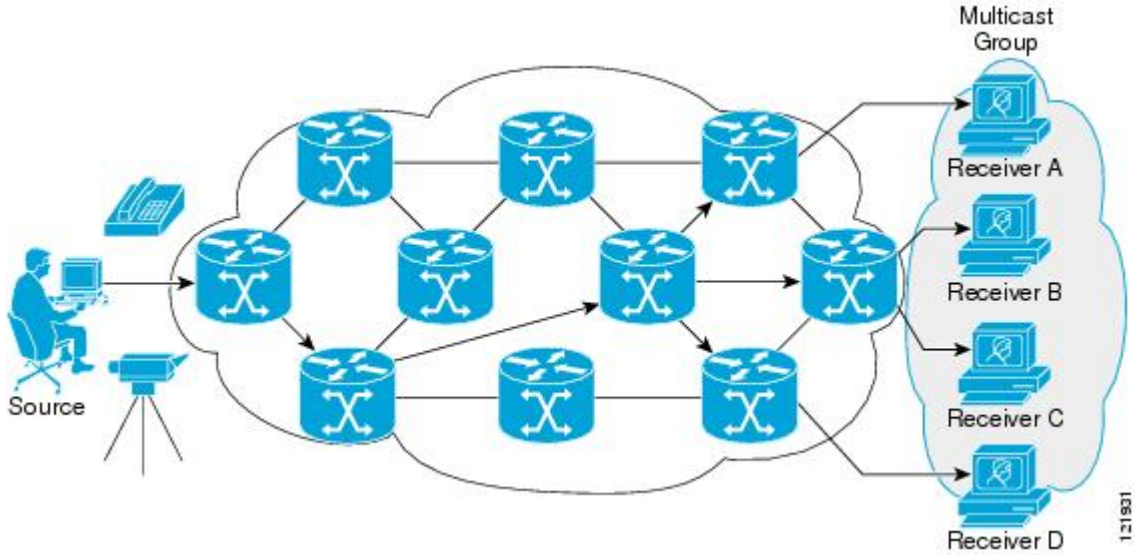
IP communication consists of hosts that act as senders and receivers of traffic as shown in the first figure. Senders are called sources. Traditional IP communication is accomplished by a single host source sending packets to another single host (unicast transmission) or to all hosts (broadcast transmission). With IP multicast, you can send packets to a specific group of hosts instead of all hosts (multicast transmission). This subset, comprising 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 consists of receivers joining a group to receive a specific 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 interested in receiving data to a particular group must join the group from the source. A host receiver joins a group using 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. Only members of a group receive packets sent to that group. Multicast packets, like IP unicast packets, use best-effort reliability for delivery.



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.



Source specific multicast

Source Specific Multicast (SSM) is a datagram delivery model that effectively supports one-to-many applications, also known as broadcast applications. SSM is a core network technology in the Cisco implementation of IP multicast, aimed at 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 indicates a desire 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. Applications on different source hosts can reuse SSM group addresses without causing excess network traffic.

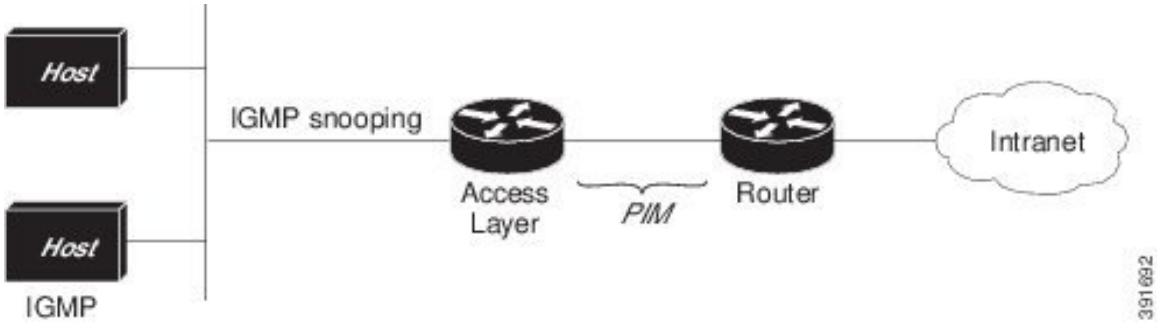
IP multicast routing protocols

The software supports these protocols to implement IP multicast routing:

- IGMP is used between hosts on a LAN and the routers (and multilayer devices) on that LAN to track the multicast groups of which hosts are members. To participate in IP multicasting, multicast hosts, routers, and multilayer devices must have the Internet Group Management Protocol (IGMP) operating.
- PIM is used between routers to track which multicast packets to forward to other routers and their directly connected LANs.
- IGMP Snooping is used for multicasting in a Layer 2 switching environment. It reduces multicast traffic flooding by dynamically configuring Layer 2 interfaces to forward traffic only to interfaces associated with IP multicast devices.

The figure illustrates the role of each protocol within the IP multicast environment.

Figure 1: IP multicast routing protocols



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

When the destination IPv4 address differs from the destination MAC address, a multicast packet is considered unmatched. The device forwards the unmatched packet in hardware based upon the MAC address table. If the destination MAC address is absent from the MAC address table, the device will flood the packet to all ports within the VLAN of the receiving port.

Internet group management protocol

IGMP messages are used by IP multicast hosts to send their local Layer 3 switch or router a request to join a specific multicast group and begin receiving multicast traffic. With extensions in IGMPv2, IP hosts can request a Layer 3 switch or router to leave an IP multicast group and cease receiving the multicast group traffic.

A Layer 3 switch or router maintains a list of multicast group memberships on a per-interface basis using the information obtained by IGMP. Multicast group membership stays active on an interface if a host on that interface sends an IGMP request for multicast group traffic.

Protocol-independent multicast

Protocol-Independent Multicast (PIM) leverages whichever unicast routing protocol you use to populate the unicast routing table, including EIGRP, OSPF, or static route, to support IP multicast.

PIM performs the reverse path forwarding (RPF) check using a unicast routing table, avoiding the need for a separate multicast routing table. PIM does not send and receive multicast routing updates between routers like other routing protocols do.

PIM sparse mode

PIM Sparse Mode (PIM-SM) delivers multicast traffic only to networks with active receivers that request the data explicitly. PIM-SM is intended for networks featuring several different multicast activities, including desktop video conferencing and collaborative computing, targeting a small number of receivers and occurring simultaneously.

Rendezvous point

If you configure PIM to operate in sparse mode, choose one or more devices to be rendezvous points (RPs). Senders in a multicast group use RPs to register their activity, while receivers use them to learn about new senders. You can configure Cisco IOS software so that packets for a single multicast group can use one or more RPs.

RP addresses allow 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. 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 the same group. The access list conditions determine which groups the device is an RP for; different groups can have different RPs.

IGMP snooping

Use IGMP snooping for multicasting in a Layer 2 switching environment. With IGMP snooping, a Layer 3 switch or router examines Layer 3 information in the IGMP packets in transit between hosts and a device. When you receive the IGMP Host Report, add the host's port number to the multicast table entry. When you receive the IGMP Leave Group message, remove the port from the table entry.

IGMP control messages, transmitted as multicast packets, resemble multicast data when examining only the Layer 2 header. A switch running IGMP snooping examines every multicast data packet to determine whether it contains any pertinent IGMP control information. Implement IGMP snooping on a switch with a strong CPU to maintain performance even at high data transmission rates.

IP multicast addressing and scoping

This section describes about IP multicast addressing and its supported range of addresses.

IP multicast group addressing

A multicast group is identified and addressed by its multicast group address, which multicast packets are delivered to. Unlike unicast addresses, which identify a single host, multicast IP addresses do not specify 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. This indicates that they wish to receive traffic sent to that group. The source assigns the multicast group address to a group. Network administrators must ensure multicast group addresses conform to the address range reserved by the Internet Assigned Numbers Authority (IANA).

Note On Cisco C9610 Series Smart Switches, Simple Service Discovery Protocol (SSDP) group IP 239.255.255.250 is blocked by default. To use this group IP, configure the **platform ip multicast ssdp** command.

IP class D addresses

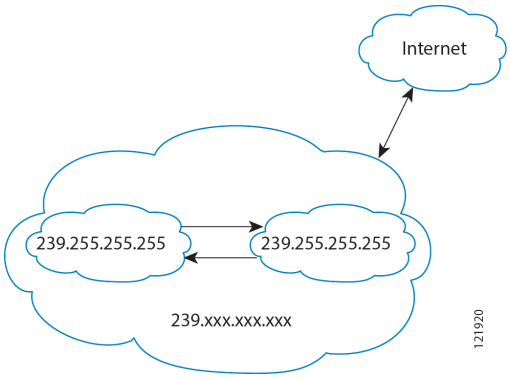
IP multicast addresses have been assigned to the IPv4 Class D address space by IANA. The high-order four bits of a Class D address are 1110. Host group addresses are in the range 224.0.0.0 to 239.255.255.255. The source chooses a multicast address for the receivers in a multicast group.

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

IP multicast boundary

The figure illustrates how address scoping defines domain boundaries to prevent domains, each with RPs sharing the same IP address, from leaking into one another. Scoping is performed on the subnet boundaries within large domains and on the boundaries between the domain and the Internet.

Figure 2: Address scoping at boundaries



Select the **ip multicast boundary** command with the *access-list* to set up an administratively scoped boundary on an interface. A standard access list defines the range of addresses affected. Boundaries prevent multicast data packets from flowing across in 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. These addresses are considered local rather than 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. Auto-RP group range announcements are permitted and passed by the boundary only if all addresses in the Auto-RP group range are permitted by the boundary ACL. If any address is not permitted, the entire group range is filtered and removed from the Auto-RP message before the Auto-RP message is forwarded.

IP multicast address scoping

The multicast address range is subdivided to provide predictable behavior and enable address reuse within smaller domains. The table provides a summary of the multicast address ranges and each range is briefly described.

Table 1: Multicast address range assignments

Name	Range	Description
Reserved link-local addresses	224.0.0.0 to 224.0.0.255	Reserved for use by network protocols on a local network segment.
Globally scoped addresses	224.0.1.0 to 238.255.255.255	Reserved to send multicast data between organizations and across the Internet.
Source specific multicast	232.0.0.0 to 232.255.255.255	Reserved for use with the SSM datagram delivery model where data is forwarded only to receivers that have explicitly joined the group.

Name	Range	Description
GLOP addresses	233.0.0.0 to 233.255.255.255	Reserved for statically defined addresses by organizations that already have an assigned autonomous system (AS) domain number.
Limited scope address	239.0.0.0 to 239.255.255.255	Reserved as administratively or limited scope addresses for use in private multicast domains.

Reserved link-local addresses

The IANA has reserved the range 224.0.0.0 to 224.0.0.255 for use by network protocols on a local network segment. IP routers do not forward packets within this range which are local in scope. Typically, packets with link local destination addresses are sent with a TTL value of 1 and remain unforwarded by routers.

Reserved link-local addresses serve network protocol functions. Network protocols use these addresses for automatic router discovery and to communicate important routing information. For example, Open Shortest Path First (OSPF) uses the IP addresses 224.0.0.5 and 224.0.0.6 to exchange link-state information.

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

Globally scoped addresses

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

Source specific multicast addresses

Addresses in the range 232.0.0.0/8 are reserved for Source Specific Multicast (SSM) by IANA. Use the **ip pim ssm** command to configure SSM for arbitrary IP multicast addresses also. SSM is an extension of PIM that allows for an efficient data delivery mechanism in one-to-many communications.

GLOP addresses

GLOP addressing, as proposed by RFC 2770, reserves the 233.0.0.0/8 range for organizations with a reserved AS number for statically defined addresses. 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 only received 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 differentiate between several multicast groups. One method to achieve this is mapping IP multicast Class D addresses directly to a MAC address. With this method, NICs can receive packets destined for many different MAC addresses.

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

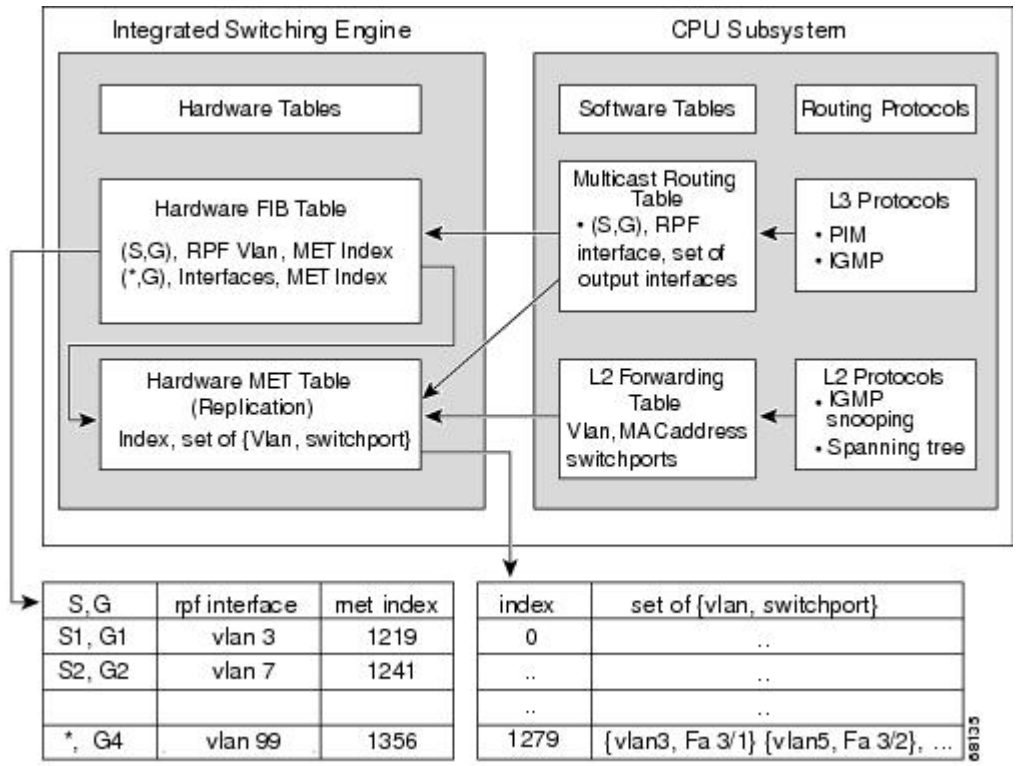
IP multicast forwarding and routing

This section describes about multicast packet forwarding and routing.

IP multicast tables

This illustration presents key data structures used by the device to forward IP multicast packets in hardware.

Figure 3: IP multicast tables and protocols



The Integrated Switching Engine maintains the hardware FIB table to identify individual IP multicast routes. Each entry consists of a destination group IP address and an optional source IP address. Multicast traffic flows on primarily two types of routes: (S,G) and (*,G). The (S,G) routes flow from a source to a group based on the IP address of the multicast source and the IP address of the multicast group destination. Traffic on a (*,G) route flows from the PIM RP to all receivers of group G. Sparse-mode groups are the only ones that use (*,G) routes. The Integrated Switching Engine hardware supports 128,000 routes shared by unicast, multicast, and multicast fast-drop entries.

The multicast expansion table (MET) stores output interface lists. The MET has room for up to 32,000 output interface lists. (For RET, we can have up to 102 K entries (32 K used for floodsets, 70,000 used for multicast entries)). The MET resources are shared by both Layer 3 multicast routes and by Layer 2 multicast entries. The actual number of output interface lists available in hardware depends on the specific configuration. If the total number of multicast routes exceeds 32,000, the Integrated Switching Engine might not switch the multicast packets. They would be forwarded by the CPU subsystem at much slower speeds.

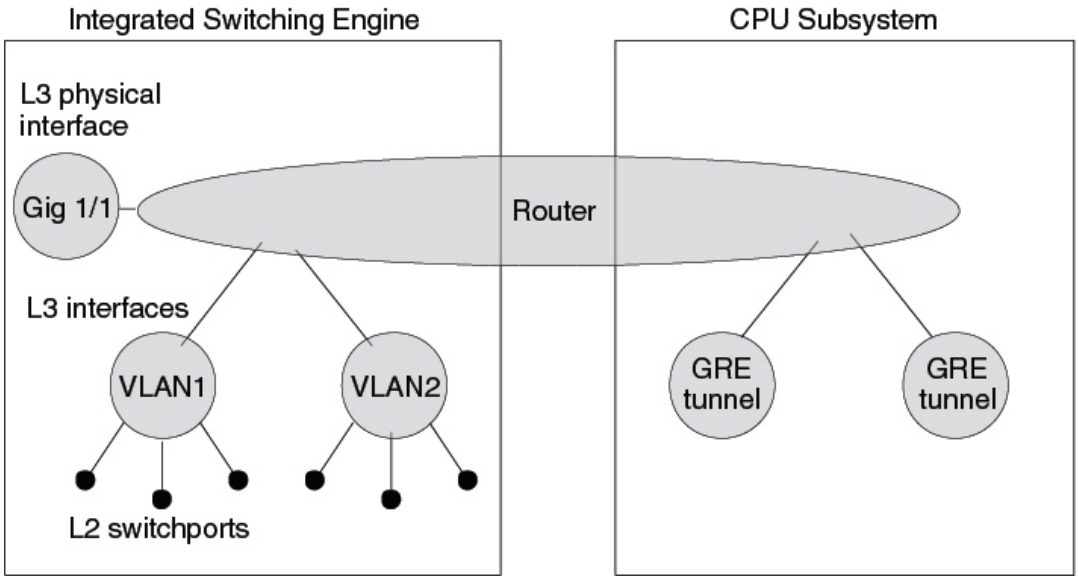
Note For RET, a maximum of 102 K entries is supported (32 K used for floodsets, 70 K used for multicast entries).

Hardware and software forwarding

The Integrated Switching Engine forwards most packets in hardware at high speeds. The CPU subsystem forwards exception packets in software. Statistical reports show that the Integrated Switching Engine forwards most packets in hardware.

This illustration shows a logical view of the hardware and software forwarding components.

Figure 4: Hardware and software forwarding components



In the normal mode of operation, the Integrated Switching Engine performs inter-VLAN routing in hardware. The CPU subsystem supports generic routing encapsulation (GRE) tunnels for forwarding in software.

Replication is a type of forwarding in which the packet is duplicated, and multiple copies are sent out instead of a single copy. Replication occurs only for multicast packets at Layer 3; unicast packets are never replicated to multiple Layer 3 interfaces. In IP multicasting, for each incoming IP multicast packet that is received, many replicas of the packet are sent out.

IP multicast packets can be transmitted on the following types of routes:

- Hardware routes
- Software routes
- Partial routes

Hardware routes occur when all packet replicas are forwarded by the Integrated Switching Engine. Software routes occur when all packet replicas are forwarded by the CPU subsystem. Partial routes occur when some replicas are forwarded by the Integrated Switching Engine, and others are forwarded by the CPU subsystem.

Partial routes

These conditions prompt the CPU subsystem software to forward replicas of a packet; however, they do not affect the performance of replicas forwarded in hardware.

- The switch is configured with the **ip igmp join-group** command as a member of the IP multicast group on the RPF interface of the multicast source.
- The switch is the first-hop to the source in PIM sparse mode. The switch must send PIM-register messages to the RP.

Software routes

Note If a condition is set on either the RPF interface or the output interface, output replication is conducted in the software.

These conditions cause all replicas of a packet for a route to be forwarded by the CPU subsystem software:

- The interface is configured with multicast helper.
- The interface is a generic routing encapsulation (GRE) or Distance Vector Multicast Routing Protocol (DVMRP) tunnel.
- The interface uses non-Advanced Research Products Agency (ARPA) encapsulation.

These packets are always forwarded in software:

- Packets sent to multicast groups that fall into the range 224.0.0.* (where * is in the range from 0 to 255). This range is used by routing protocols. Layer 3 switching supports all other multicast group addresses.
- Packets with IP options.

Static multicast route

Static multicast routes (mroutes) calculate RPF information but do not forward traffic, cannot be redistributed, and are strictly local to the device they are defined on.

Because Protocol Independent Multicast (PIM) does not have its own routing protocol, there is no mechanism to distribute static mroutes throughout the network. Static mroutes require careful attention during administration compared to unicast static routes.

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

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

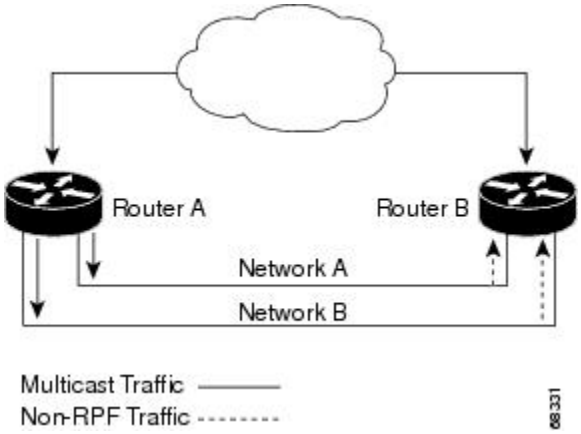
You can specify the administrative distance of an mroute with the optional distance argument. If you do not specify a value for the distance argument, the distance of the mroute defaults to zero. If the static mroute has the same distance as another RPF source, the static mroute will take precedence. There are only two exceptions to this rule: directly connected routes and the default unicast route.

Non-reverse path forwarding traffic

Traffic failing a Reverse Path Forwarding (RPF) check is called non-RPF traffic. Non-RPF traffic is forwarded by the Integrated Switching Engine by filtering (persistently dropping) or rate limiting the non-RPF traffic.

In a redundant configuration where multiple Layer 3 switches or routers connect to the same LAN segment, only one device forwards the multicast traffic from the source to the receivers on the outgoing interfaces. This network configuration can result in non-RPF traffic.

Figure 5: Redundant multicast router configuration in a stub network



In this kind of topology, only Router A, the PIM designated router (PIM DR), forwards data to the common VLAN. Router B receives the forwarded multicast traffic, but must drop this traffic because it has arrived on the wrong interface and fails the RPF check. Traffic that does not pass the RPF check is known as non-RPF traffic.

Multicast forwarding information base

The Multicast Forwarding Information Base (MFIB) is a multicast routing protocol independent forwarding engine that does not depend on PIM or any other multicast routing protocol. It is responsible for:

- Forwarding multicast packets
- Registering with the MRIB to learn the entry and interface flags set by the control plane
- Handling data-driven events that must be sent to the control plane
- Maintaining counts, rates, and bytes of received, dropped, and forwarded multicast packets

MFIB subsystem supports IP multicast routing in the Integrated Switching Engine hardware on Cisco devices. The MFIB resides logically between the IP multicast routing protocols in the CPU subsystem software and the platform-specific code that manages IP multicast routing in hardware. The MFIB simplifies routing table information for processing and forwarding by the Integrated Switching Engine hardware.

To view multicast routing table information, use the **show ip mroute** command. To display the MFIB table information, use the **show ip mfib** command.

The MFIB table has IP multicast routes like (S,G) and (*,G). Each route in the MFIB table can have one or more optional flags associated with it. The route flags indicate how a packet that matches a route should be forwarded. The Internal Copy (IC) flag on an MFIB route indicates that a switch process needs to receive a copy of the packet, for example. These flags can be associated with MFIB routes:

- Internal Copy (IC) flag: Sets on a route when a process on the router needs to receive packets matching the specified route.

- Signalling (S) flag: Sets on a route when a process needs to be notified when a packet matching the route is received; the expected behavior is that the protocol code updates the MFIB state in response to receiving a packet on a signalling interface.
- Connected (C) flag: When set on an MFIB route, the C flag means that packets from directly connected hosts are signaled to a protocol process, similar to the Signaling (S) flag.

A route can also have a set of optional flags associated with one or more interfaces. For example, an (S,G) route with the flags on VLAN 1 indicates how packets arriving on VLAN 1 should be handled, and whether packets matching the route should be forwarded onto VLAN 1. The per-interface flags supported in the MFIB include the following:

- Accepting (A): Sets on the interface that is known in multicast routing as the RPF interface. A packet that arrives on an interface that is marked as Accepting (A) is forwarded to all Forwarding (F) interfaces.
- Forwarding (F): Used in conjunction with the Accepting (A) flag as described above. The set of Forwarding interfaces that form what is often referred to as the multicast "olist" or output interface list.
- Signaling (S): Sets on an interface when some multicast routing protocol process in Cisco IOS needs to be notified of packets arriving on that interface.

Note When PIM-SM routing is in use, the MFIB route might include an interface as in this example:

```
PimTunnel [1.2.3.4]
```

It is a virtual interface that the MFIB subsystem creates to indicate that packets are being tunnelled to the specified destination address. A PimTunnel interface cannot be displayed with the normal **show interface** command.

S/M 224/4 MFIB entry

An (S/M, 224/4) entry is created in the MFIB for every multicast-enabled interface. This entry ensures that all packets sent by directly connected neighbors are encapsulated and sent to the PIM-SM RP. You can forward a small number of packets using the (S/M,224/4) route until the (S,G) route is established by PIM-SM.

For example, on an interface with IP address 10.0.0.1 and a netmask of 255.0.0.0, a route is created to match all IP multicast packets where the source address is any address in the class A network 10. This route can be written in conventional subnet/masklength notation as (10/8,224/4). If an interface has multiple assigned IP addresses, one route is created for each IP address.

Cisco express forwarding, MFIB, and Layer 2 forwarding

The implementation of IP multicast is an extension of centralized Cisco Express Forwarding. Cisco Express Forwarding extracts information from the unicast routing table, which is created by unicast routing protocols, such as BGP, OSPF, and EIGRP and loads it into the hardware

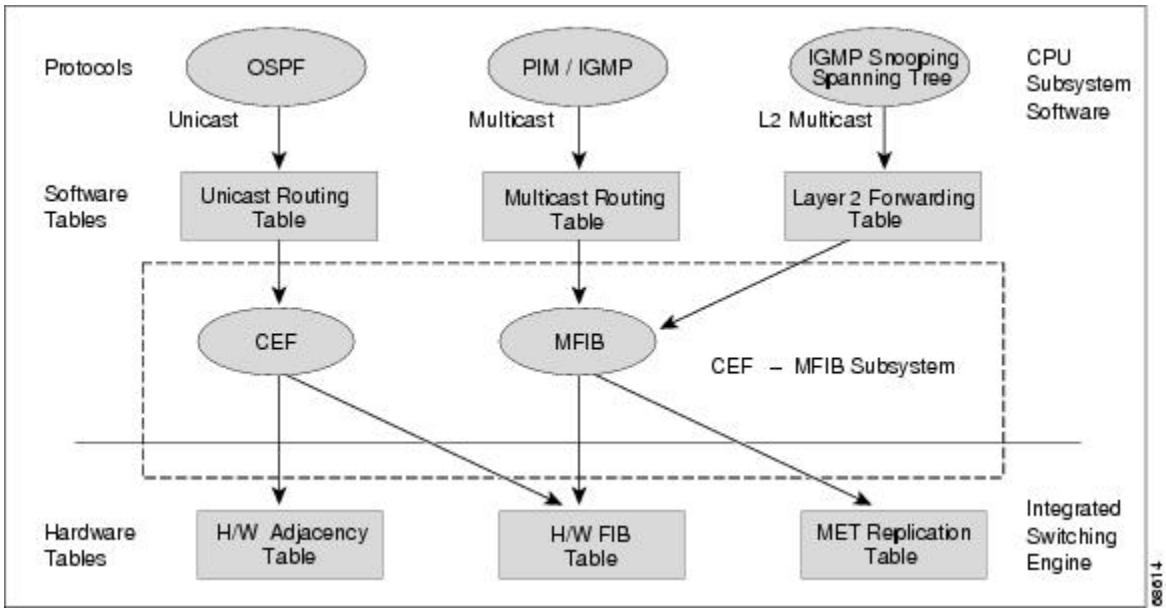
Forwarding Information Base (FIB). With the unicast routes in the FIB, when a route is changed in the upper-layer routing table, only one route needs to be changed in the hardware routing state. To forward unicast packets in hardware, the Integrated Switching Engine looks up source and destination routes in ternary content addressable memory (TCAM), takes the adjacency index from the hardware FIB, and gets the Layer 2 rewrite information and next-hop address from the hardware adjacency table.

The new Multicast Forwarding Information Base (MFIB) subsystem is the multicast analog of the unicast Cisco Express Forwarding. The MFIB subsystem extracts the multicast routes that PIM and IGMP create and refines them into a protocol-independent format for forwarding in hardware. The MFIB subsystem removes the protocol-specific information and leaves only the essential forwarding information.

Each entry in the MFIB table consists of an (S,G) or (*,G) route, an input RPF VLAN, and a list of Layer 3 output interfaces. The MFIB subsystem, together with platform-dependent management software, loads this multicast routing information into the hardware FIB and Replica Expansion Table (RET). The device performs Layer 3 routing and Layer 2 bridging at the same time. There can be multiple Layer 2 switch ports on any VLAN interface.

The following illustration shows a functional overview of how a Cisco device combines unicast routing, multicast routing, and Layer 2 bridging information to forward in hardware:

Figure 6: Combining Cisco express forwarding, MFIB, and Layer 2 forwarding information in hardware



Like the Cisco Express Forwarding unicast routes, the MFIB routes are Layer 3 and must be merged with the appropriate Layer 2 information. The following example shows an MFIB route:

```
(*,203.0.113.1)
RPF interface is Vlan3
Output Interfaces are:
Vlan 1
Vlan 2
```

The route (*,203.0.113.1) is loaded in the hardware FIB table and the list of output interfaces is loaded into the MET. A pointer to the list of output interfaces, the MET index, and the RPF interface are also loaded in the hardware FIB with the (*,203.0.113.1) route. With this information loaded in hardware, merging of the Layer 2 information can begin. For the output interfaces on VLAN1, the Integrated Switching Engine must send the packet to all switch ports in VLAN1 that are in the spanning tree forwarding state. The same process applies to VLAN 2. To determine the set of switch ports in VLAN 2, the Layer 2 Forwarding Table is used.

When the hardware routes a packet, in addition to sending it to all of the switch ports on all output interfaces, the hardware also sends the packet to all switch ports (other than the one it arrived on) in the input VLAN. For example, assume that VLAN 3 has two switch ports in it, GigabitEthernet 3/1 and GigabitEthernet 3/2. If a host on GigabitEthernet 3/1 sends a multicast packet, the host on GigabitEthernet 3/2 might also need to

receive the packet. To send a multicast packet to the host on GigabitEthernet 3/2, all of the switch ports in the ingress VLAN must be added to the port set that is loaded in the MET.

If VLAN 1 contains 1/1 and 1/2, VLAN 2 contains 2/1 and 2/2, and VLAN 3 contains 3/1 and 3/2, the MET chain for this route would contain these switch ports: (1/1,1/2,2/1,2/2,3/1, and 3/2).

If IGMP snooping is on, the packet should not be forwarded to all output switch ports on VLAN 2. The packet should be forwarded only to switch ports where IGMP snooping has determined that there is either a group member or router. For example, if VLAN 1 had IGMP snooping enabled, and IGMP snooping determined that only port 1/2 had a group member on it, then the MET chain would contain these switch ports: (1/1,1/2, 2/1, 2/2, 3/1, and 3/2).

Multicast fast drop

In IP multicast protocols, such as PIM-SM and PIM-DM, every (S,G) or (*,G) route has an incoming interface associated with it. This interface is called the reverse path forwarding interface. In some cases, when a packet arrives on an interface other than the expected RPF interface, the packet must be forwarded to the CPU subsystem software to allow PIM to perform special protocol processing on the packet. One example of this special protocol processing that PIM performs is the PIM Assert protocol.

By default, the Integrated Switching Engine hardware sends all packets that arrive on a non-RPF interface to the CPU subsystem software. However, processing in software is not necessary in many cases, because these non-RPF packets are often not needed by the multicast routing protocols. Non-RPF packets sent to the software can overwhelm the CPU if their processing is not managed.

Instead of installing fast-drop entries, the Cisco device uses Dynamic Buffer Limiting (DBL). This flow-based congestion avoidance mechanism provides active queue management by tracking the queue length for each traffic flow. When the queue length of a flow exceeds its set limit, DBL drops packets. Rate DBL limits the non-RPF traffic to the CPU subsystem so that the CPU is not overwhelmed. The packets are rate limited per flow to the CPU. Because installing fast-drop entries in the CAM is inaccessibly, the number of fast-drop flows that can be handled by the switch need not be limited.

Protocol events, such as a link going down or a change in the unicast routing table, can impact the set of packets that can safely be fast dropped. After a topology change, packets that were previously fast-dropped might require forwarding to the CPU subsystem for proper PIM processing. The CPU subsystem software handles flushing fast-drop entries in response to protocol events so that the PIM code in IOS can process all the necessary RPF failures.

The use of fast-drop entries in the hardware is critical in some common topologies because you may have persistent RPF failures. Without the fast-drop entries, the CPU is exhausted by RPF failed packets that it did not need to process.

Multicast high availability

The device supports multicast high availability, ensuring continuous multicast traffic flow in case of a supervisor engine failure. MFIB states are synced to the standby supervisor engine before a switchover, ensuring NSF availability with a fast convergence upon switchover during a supervisor engine failure.

Multicast HA (SSO / NSF / ISSU) is supported for the PIM Sparse mode, SSM mode, and in Layer 2 for IGMP and MLD Snooping.

Default IP multicast routing configuration

This table displays the default IP multicast routing configuration.

Table 2: Default IP multicast routing configuration

Feature	Default Setting
Multicast routing	Disabled on all interfaces.
PIM version	Version 2.
PIM mode	No mode is defined.
PIM stub routing	None configured.
PIM RP address	None configured.
PIM domain border	Disabled.
PIM multicast boundary	None.
Candidate BSRs	Disabled.
Candidate RPs	Disabled.
Shortest-path tree threshold rate	0 kb/s.
PIM router query message interval	30 seconds.

Configure basic IP multicast routing

This section provides information about configuring basic IP multicast routing.

Configure basic IP multicast routing

Before you begin

Note By default, multicast routing is disabled, and there is no default mode setting. To enable multicast routing, use the **ip multicast-routing** command.

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

In populating the multicast routing table, dense-mode interfaces are always added to the table. Sparse-mode interfaces are added to the table only when periodic join messages are received from downstream devices or when there is a directly connected member on the interface. Sparse-mode operation occurs if there is an RP known for the group when forwarding from a LAN. Packets are then encapsulated and sent toward the RP. Without a known RP, the packet is flooded in a dense-mode fashion. For both PIM dense mode and PIM any-source multicast mode, the multicast source address must be on the directly connected incoming interface within the same subnet of the first-hop router. If the multicast traffic from a specific source is sufficient, the receiver's first-hop router will send join messages toward the source to build a source-based distribution tree.

Procedure

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

	Command or Action	Purpose
	Example: `Device(config-if)# ip pim sparse-mode`	By default, no mode is configured. The keywords have these meanings: • **dense-mode**: Enables dense mode of operation. • **sparse-mode**: Enables sparse mode of operation. If you configure sparse mode, you must also configure an RP. • **sparse-dense-mode**: Causes the interface to be treated in the mode in which the group belongs. Sparse-dense mode is the recommended setting. **Note** To disable PIM on an interface, use the **no ip pim** interface configuration command.
Step 6	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 7	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 8	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configure static mroute

Procedure

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip mroute** [**vrf** *vrf-name*] *source-address mask* { **fallback-lookup** {**global** | **vrf** *vrf-name* }[*protocol*] {*rpf-address* | *interface-type interface-number*}} [**distance**] **Example:** `Device(config)# ip mroute 10.1.1.1 255.255.255.255 10.2.2.2`	The source IP address 10.1.1.1 is accessible via the interface linked to IP address 10.2.2.2.
Step 4	**exit** **Example:** `Device(config)# exit`	Returns to privileged EXEC mode.
Step 5	**show running-config** **Example:** `Device# show running-config`	(Optional) Verifies your entries.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Enable sdr listener support

By default, the device does not listen to session directory advertisements. This procedure is optional.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet 1/0/1`	Specifies the interface to be enabled for sdr, and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**ip sap listen** **Example:** `Device(config-if)# ip sap listen`	Enables the device software to listen to session directory announcements.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 6	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Limit sdr cache entry

Entries are retained indefinitely in the sdr cache by default. You can limit how long the entry remains active so that if a source stops advertising SAP information, old advertisements are not unnecessarily kept.

This procedure is optional.

Procedure

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

	Command or Action	Purpose
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 5	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 6	**show ip sap** **Example:** `Device# show ip sap`	Displays the SAP cache.
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Monitor and maintain basic IP multicast routing

The commands in this section can be used to monitor and maintain basic IP multicast routing.

Clear caches, tables, and databases

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

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

Table 3: Commands for clearing caches, tables, and databases

Command	Purpose
clear ip igmp group {group [*hostname* | *IP address*] | vrf *name* group [*hostname* | *IP address*] }	Deletes entries from the IGMP cache.
clear ip mfib { counters [*group* | *source*] | global counters [*group* | *source*] | vrf * }	Clears all active IPv4 Multicast Forwarding Information Base (MFIB) traffic counters.
clear ip mrm {status-report [*source*] }	Clears IP multicast routing monitor status reports.
clear ip mroute { * | [*hostname* | *IP address*] | vrf *name* group [*hostname* | *IP address*] }	Deletes entries from the IP multicast routing table.
clear ip msdp { peer | sa-cache | statistics | vrf }	Clears the Multicast Source Discovery Protocol (MSDP) cache.

Command	Purpose
clear ip multicast { limit | redundancy statistics }	Clears the IP multicast information.
clear ip pim { df [int | rp *rp address*] | interface | rp-mapping [*rp address*] | vrf *vpn name* { df | interface | rp-mapping }	Clears the PIM cache.
clear ip sap [*group-address* | *"session-name"*]	Deletes the Session Directory Protocol Version 2 cache or an sdr cache entry.

Display system and network statistics

You can view specific statistics like the IP routing tables, caches, and databases to learn resource usage and solve network problems.

You can also view information about node reachability and discover the routing path that packets of your device are taking through the network.

Note This release does not support per-route statistics.

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

Table 4: Commands for displaying system and network statistics

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

Command	Purpose
show ip mfib [*type number*]	Displays the IP multicast forwarding information base.
show ip mrib {**client** | **route** | **vrf**}	Displays the multicast routing information base.
show ip mrm {**interface** | **manager** | **status-report**}	Displays the IP multicast routing monitor information.
show ip mroute [*group-name* | *group-address*] [*source*] [**count** | **interface** | **proxy** | **pruned** | **summary** | **verbose**]	Displays the contents of the IP multicast routing table.
show ip msdp {**count** | **peer** | **rpf-peer** | **sa-cache** | **summary** | **vrf**}	Displays the Multicast Source Discovery Protocol (MSDP) information.
show ip multicast [**interface** | **limit** | **mpls** | **redundancy** | **vrf**]	Displays global multicast information.
show ip pim all-vrfs {**tunnel**}	Display all VRFs.
show ip pim autorp	Display global auto-RP information.
show ip pim boundary [*type number*]	Displays boundary information.
show ip pim bsr-router	Display bootstrap router information (version 2).
show ip pim interface [*type number*] [**count** | **detail** | **df** | **stats**]	Displays information about interfaces configured for PIM. This command is available in all software images.
show ip pim neighbor [*type number*]	Lists the PIM neighbors discovered by the device. This command is available in all software images.
show ip pim mdt [**bgp**]	Displays multicast tunnel information.
show ip pim rp [*group-name* | *group-address*]	Displays the RP routers associated with a sparse-mode multicast group. This command is available in all software images.
show ip pim rp-hash [*group-name* | *group-address*]	Displays the RP to be chosen based upon the group selected.
show ip pim tunnel [*tunnel* | *verbose*]	Displays the registered tunnels.
show ip pim vrf *name*	Displays VPN routing and forwarding instances.

Command	Purpose
show ip rpf {*source-address* | *name*}	Displays how the device is doing Reverse-Path Forwarding (that is, from the unicast routing table, DVMRP routing table, or static mroutes). Command parameters include: • *Host name* or *IP address*: IP name or group address. • **Select**: Group-based VRF select information. • **vrf**: Selects VPN Routing/Forwarding instance.
show ip sap [*group* | **"***session-name***"** | **detail**]	Displays the Session Announcement Protocol (SAP) Version 2 cache. Command parameters include: • *A.B.C.D*: IP group address. • *WORD*: Session name (in double quotes). • **detail**: Session details.

Configuration examples

This section provides configuration examples for basic IP multicast routing.

Example: Configure an IP multicast boundary

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

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

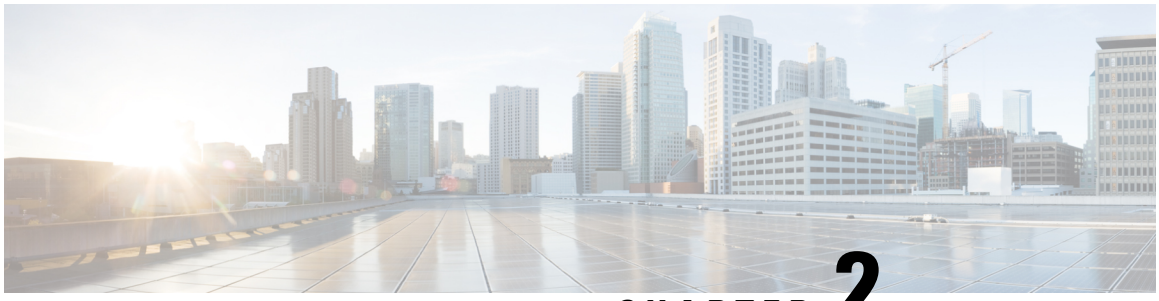
Example: Respond to mrinfo requests

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

```
Device# mrinfo

  171.69.214.27 (mm1-7kd.cisco.com) [version cisco 11.1] [flags: PMS]:
  171.69.214.27 -> 171.69.214.26 (mm1-r7kb.cisco.com) [1/0/pim/querier]
  171.69.214.27 -> 171.69.214.25 (mm1-45a.cisco.com) [1/0/pim/querier]
  171.69.214.33 -> 171.69.214.34 (mm1-45c.cisco.com) [1/0/pim]
  171.69.214.137 -> 0.0.0.0 [1/0/pim/querier/down/leaf]
  171.69.214.203 -> 0.0.0.0 [1/0/pim/querier/down/leaf]
```

```
171.69.214.18 -> 171.69.214.20 (mm1-45e.cisco.com) [1/0/pim]
171.69.214.18 -> 171.69.214.19 (mm1-45c.cisco.com) [1/0/pim]
171.69.214.18 -> 171.69.214.17 (mm1-45a.cisco.com) [1/0/pim]
```



CHAPTER 2

IGMP

Feature history for IGMP

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature name and description	Supported platform
Cisco IOS XE 17.18.1	IGMP: IGMP is a communication protocol used between hosts on a LAN and network devices to monitor IP multicast group memberships.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

Understand IGMP

This section describes about Internet Group Management Protocol (IGMP) and its features.

IGMP

IGMP is a communication protocol used between hosts on a LAN and network devices to monitor IP multicast group memberships. To participate in IP multicasting, multicast hosts, routers, and multilayer devices must have IGMP operating.

Role of IGMP

IGMP is used to dynamically register individual hosts in a multicast group on a particular LAN, automatically controlling and limiting multicast traffic using special multicast queriers and hosts. Enabling PIM on an interface also enables IGMP.

- A querier is a network device, such as a router, that sends query messages to identify the network devices belonging to a specific multicast group.
- A host is a receiver, including routers, that sends report messages (in response to query messages) to inform the querier of a host membership. Hosts use IGMP messages to join and leave multicast groups.

Hosts identify group memberships by sending IGMP messages to their local multicast device. Under IGMP, devices process IGMP messages and periodically send queries to determine which groups are active or inactive on a particular subnet.

IGMP multicast addresses

IP multicast traffic uses group addresses, which are Class D IP addresses. The high-order four bits of a Class D address are 1110. Therefore, host group addresses can be in the range 224.0.0.0 to 239.255.255.255.

Multicast addresses ranging from 224.0.0.0 to 224.0.0.255 are reserved for routing protocols and other network control traffic. The address 224.0.0.0 is guaranteed not to be assigned to any group.

IGMP packets use IP multicast group addresses for transmission:

- IGMP general queries are destined to the address 224.0.0.1 (all systems on a subnet).
- IGMP group-specific queries are destined to the group IP address for which the device is querying.
- IGMP group membership reports are sent to the group IP address for which the device is reporting.
- IGMPv2 leave-group messages are destined to the address 224.0.0.2 (all devices on a subnet).
- IGMPv3 membership reports go to 224.0.0.22. All IGMPv3-capable devices must listen to this address.

IGMP versions

The device supports IGMP versions 1, 2, and 3. The device interoperates with these versions. For example, if IGMP snooping is enabled and the querier's version is IGMPv2, the device forwards the IGMPv3 report to the multicast router upon receiving it from a host.

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

IGMP version 1

IGMP version 1 (IGMPv1) uses a query-response model, allowing the multicast router and multilayer device to identify active multicast groups on the local subnet, characterized by having one or more hosts interested in a multicast group. For more information, see RFC 1112.

IGMP version 2

IGMP version 2 (IGMPv2) extends IGMP functionality by providing features like the IGMP leave process to reduce leave latency, group-specific queries, and a defined maximum query response time. IGMPv2 enables routers to elect the IGMP querier independently of the multicast protocol. For more information, see RFC 2236.

Note IGMP version 2 is the default version.

IGMP version 3

An IGMP version 3 (IGMPv3) device supports Basic IGMPv3 Snooping Support (BISS), which includes snooping features for IGMPv1 and IGMPv2 switches, as well as IGMPv3 membership report messages. BISS constrains the flooding of multicast traffic when your network includes IGMPv3 hosts. It constrains traffic to approximately the same set of ports as the IGMP snooping feature on IGMPv2 or IGMPv1 hosts.

IGMPv3 devices can both receive and forward messages with devices using the Source Specific Multicast (SSM) feature.

IGMPv3 host signaling

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

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

IGMP version differences

There are three versions of IGMP, as defined by Request for Comments (RFC) documents of the Internet Engineering Task Force (IETF). IGMPv2 enhances IGMPv1 by allowing hosts to signal their desire to leave a multicast group. IGMPv3 further improves IGMPv2 by offering the capability to listen to multicast traffic originating from specific source IP addresses.

Table 5: IGMP versions

IGMP Version	Description
IGMPv1	Provides the basic query-response mechanism that allows the multicast device to determine which multicast groups are active and other processes that enable hosts to join and leave a multicast group. RFC 1112 defines the IGMPv1 host extensions for IP multicasting.
IGMPv2	Extends IGMP, allowing such capabilities as the IGMP leave process, group-specific queries, and an explicit maximum response time field. IGMPv2 also adds the capability for devices to elect the IGMP querier without dependence on the multicast protocol to perform this task. RFC 2236 defines IGMPv2.

IGMP Version	Description
IGMPv3	Provides for source filtering, which enables a multicast receiver host to signal to a device which groups it wants to receive multicast traffic from, and from which sources this traffic is expected. In addition, IGMPv3 supports the link local address 224.0.0.22, which is the destination IP address for IGMPv3 membership reports; all IGMPv3-capable multicast devices must listen to this address. RFC 3376 defines IGMPv3.

Note By default, enabling a PIM on an interface enables IGMPv2 on that device. IGMPv2 was designed to be backward compatible with IGMPv1. To accomplish this backward compatibility, RFC 2236 defined special interoperability rules. If your network contains legacy IGMPv1 hosts, you should be familiar with these operability rules. For more information about IGMPv1 and IGMPv2 interoperability, see RFC 2236, Internet Group Management Protocol, Version 2 .

Devices running IGMPv1

IGMPv1 devices send IGMP queries to the “all-hosts” multicast address of 224.0.0.1 to solicit multicast groups with active multicast receivers. The multicast receivers can also send IGMP reports to the device to notify it that they are interested in receiving a particular multicast stream. Hosts can send the report asynchronously or in response to the IGMP queries sent by the device. If more than one multicast receiver exists for the same multicast group, only one of these hosts sends an IGMP report message; the other hosts suppress their report messages.

In IGMPv1, there is no election of an IGMP querier. If more than one device on the segment exists, all the devices send periodic IGMP queries. IGMPv1 has no special mechanism by which the hosts can leave the group. If the hosts are no longer interested in receiving multicast packets for a particular group, they simply do not reply to the IGMP query packets sent from the device. The device continues sending query packets. If the device does not hear a response in three IGMP queries, the group times out and the device stops sending multicast packets on the segment for the group. If you want to receive multicast packets after the timeout period, just send a new IGMP join to the device, and the device begins to forward the multicast packet again.

If multiple devices are on a LAN, elect a designated router (DR) to avoid duplicating multicast traffic. PIM devices use an election process to select a DR—the device with the highest IP address becomes the DR.

The DR is responsible for these tasks:

- Sending PIM register and PIM Join and Prune messages toward the rendezvous point (RP) to inform it about host group membership.
- Sending IGMP host-query messages.
- Sending host-query messages by default every 60 seconds in order to keep the IGMP overhead on hosts and networks very low.

Devices running IGMPv2

IGMPv2 improves the query messaging capabilities of IGMPv1.

The query and membership report messages in IGMPv2 are identical to the IGMPv1 messages with two exceptions:

- IGMPv2 query messages are broken into two categories: general queries (identical to IGMPv1 queries) and group-specific queries.
- IGMPv1 membership reports and IGMPv2 membership reports have different IGMP type codes.

IGMPv2 also enhances IGMP by providing support for the following capabilities:

- Querier election process: Provides the capability for IGMPv2 devices to elect the IGMP querier without having to rely on the multicast routing protocol to perform the process.
- Maximum Response Time field: A new field in query messages permits the IGMP querier to specify the maximum query-response time. This field permits the tuning of the query-response process to control response burstiness and to fine-tune leave latencies.
- Group-Specific Query messages: Permits the IGMP querier to perform the query operation on a specific group instead of all groups.
- Leave-Group messages: Provides hosts with a method of notifying devices on the network that they wish to leave the group.

Unlike IGMPv1, in which the DR and the IGMP querier are typically the same device, in IGMPv2 the two functions are decoupled. The DR and the IGMP querier are selected based on different criteria and may be different devices on the same subnet. The DR is the device with the highest IP address on the subnet, whereas the IGMP querier is the device with the lowest IP address.

Query messages are used to elect the IGMP querier as follows:

1. When IGMPv2 devices start, they each multicast a general query message to the all-systems group address of 224.0.0.1 with their interface address in the source IP address field of the message.
2. When an IGMPv2 device receives a general query message, the device compares the source IP address in the message with its own interface address. The device with the lowest IP address on the subnet is elected the IGMP querier.
3. All devices (excluding the querier) start the query timer, which is reset whenever a general query message is received from the IGMP querier. If the query timer expires, it is assumed that the IGMP querier has gone down, and the election process is performed again to elect a new IGMP querier.

By default, the timer is two times the query interval.

Devices running IGMPv3

IGMPv3 supports source filtering, enabling multicast receiver hosts to signal desired multicast group memberships and source IP addresses from which traffic is expected. This information allows software to forward traffic exclusively from requested sources.

IGMPv3 supports applications that explicitly signal sources for traffic receipt. Receivers using IGMPv3 can signal membership to a multicast group in two primary modes:

- INCLUDE mode: In this mode, the receiver announces membership to a group and provides a list of IP addresses (the INCLUDE list) from which it wants to receive traffic.
- EXCLUDE mode: In this mode, the receiver announces membership to a group and provides a list of IP addresses (the EXCLUDE list) from which it does not want to receive traffic. In other words, the host wants to receive traffic only from sources whose IP addresses are not listed in the EXCLUDE list. To receive traffic from all sources, like in the case of the Internet Standard Multicast (ISM) service model, a host expresses EXCLUDE mode membership with an empty EXCLUDE list.

IGMPv3 is the industry-designated standard protocol for hosts to signal channel subscriptions in an SSM network environment. For SSM to rely on IGMPv3, IGMPv3 must be available in the network stack portion of the operating systems running on the last hop devices and hosts and be used by the applications running on those hosts.

In IGMPv3, hosts send their membership reports to 224.0.0.22; all IGMPv3 devices, therefore, must listen to this address. Hosts, however, do not listen or respond to 224.0.0.22; they only send their reports to that address. In addition, in IGMPv3, there is no membership report suppression because IGMPv3 hosts do not listen to the reports sent by other hosts. Therefore, when a general query is sent out, all hosts on the wire respond.

IGMP join process

When a host wants to join a multicast group, it sends one or more unsolicited membership reports for the multicast group it wants to join. The IGMP join process is the same for IGMPv1 and IGMPv2 hosts.

In IGMPv3, the join process for hosts includes:

- When a host wants to join a group, it sends an IGMPv3 membership report to 224.0.0.22 with an empty EXCLUDE list.
- When a host wants to join a specific channel, it sends an IGMPv3 membership report to 224.0.0.22 with the address of the specific source included in the INCLUDE list.
- When a host wants to join a group but exclude particular sources, it sends an IGMPv3 membership report to 224.0.0.22, listing excluded sources in the EXCLUDE list.

Note When some IGMPv3 hosts on a LAN wish to exclude a source while others want to include it, the device opts to send traffic for the source on the LAN because inclusion takes precedence over exclusion in this situation.

IGMP leave process

The way you leave a group depends on which version of IGMP you are using.

IGMPv1 leave process

There is no leave-group message to notify devices on the subnet when a host no longer wants to receive multicast traffic from a specific group. The host simply stops processing traffic for the multicast group and ceases responding to IGMP queries with IGMP membership reports for the group. As a result, the only way IGMPv1 devices know that there are no longer any active receivers for a particular multicast group on a subnet is when the devices stop receiving membership reports.

To facilitate this process, IGMPv1 devices associate a countdown timer with an IGMP group on a subnet. When a membership report is received for the group on the subnet, the timer is reset. For IGMPv1 devices, this timeout interval is typically three times the query interval (3 minutes). This timeout interval means that the device may continue to forward multicast traffic onto the subnet for up to 3 minutes after all hosts have left the multicast group.

IGMPv2 leave process

IGMPv2 incorporates a leave-group message that provides the means for a host to indicate that it wishes to stop receiving multicast traffic for a specific group. When an IGMPv2 host leaves a multicast group, if it was

the last host to respond to a query with a membership report for that group, it sends a leave-group message to the all-devices multicast group (224.0.0.2).

IGMPv3 leave process

IGMPv3 enhances the leave process by introducing the capability for a host to stop receiving traffic from a particular group, source, or channel in IGMPv3 membership reports by including or excluding originating sources, target groups, or specific channels.

IGMP snooping

IGMP snooping is used for multicasting in Layer 2 setups by configuring interfaces to forward traffic to relevant IP multicast devices. As the name implies, IGMP snooping requires the LAN device to snoop on the IGMP transmissions between the host and the router and to keep track of multicast groups and member ports. The device manages host port numbers based on IGMP activity. It adds numbers upon receiving IGMP reports and removes them upon receiving Leave Group messages. It also periodically deletes entries if it does not receive IGMP membership reports from the multicast clients.

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

The multicast router sends queries to all VLANs, and interested hosts send join requests which are added to the forwarding table. The device creates one entry per VLAN in the IGMP snooping IP multicast forwarding table for each group from which it receives an IGMP join request.

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

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

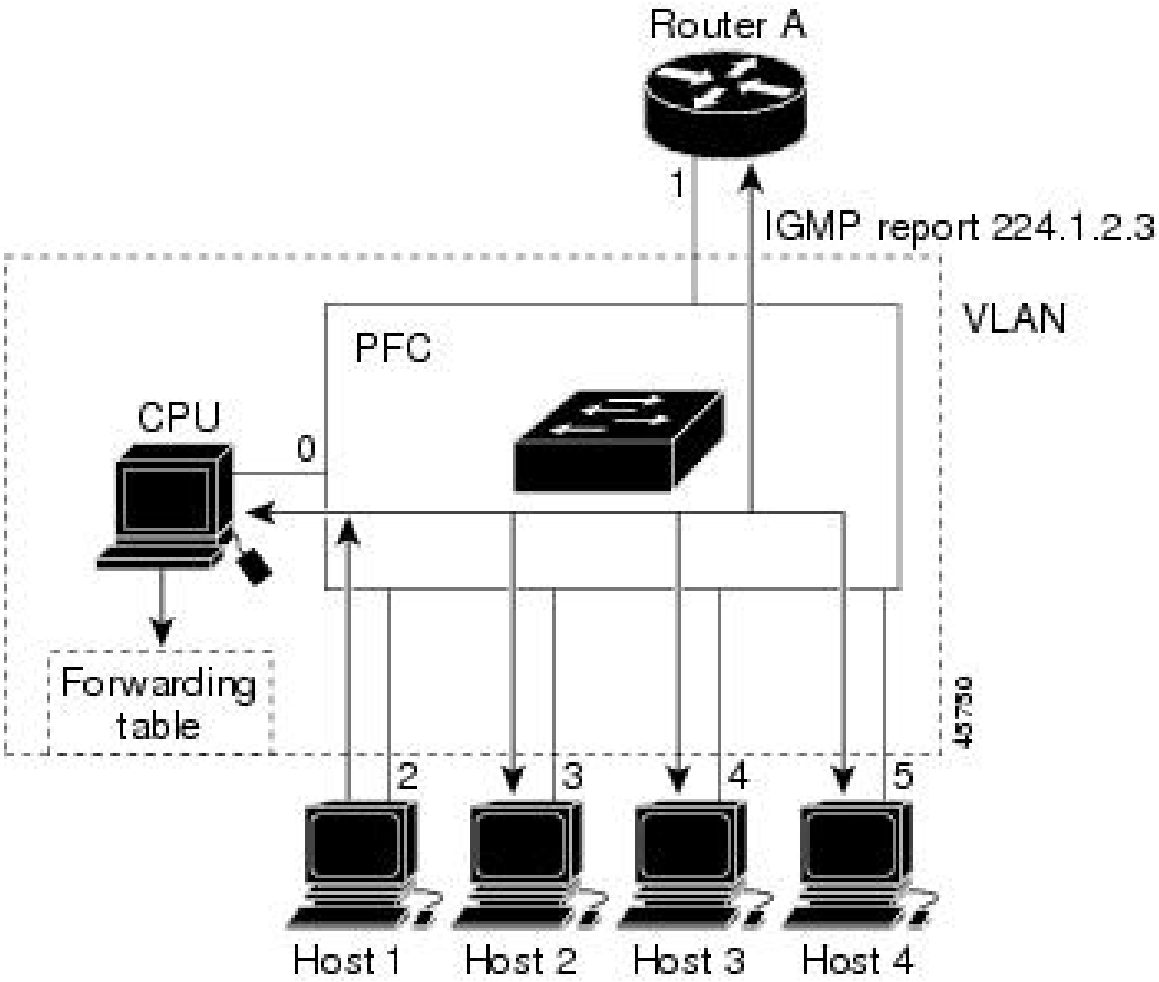
You can configure an IGMP snooping querier to support IGMP snooping in subnets without multicast interfaces because the multicast traffic does not need to be routed.

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

Join a multicast group

Figure 7: Initial IGMP join message

A host connected to the device sends an unsolicited IGMP join message specifying the IP multicast group it wants to join if it is an IGMP version 2 client. Alternatively, when the device receives a general query from the router, it forwards the query to all ports in the VLAN. IGMP version 1 or version 2 hosts wanting to join the multicast group respond by sending a join message to the device. The device CPU creates a multicast forwarding-table entry for the group if it is not already present. The CPU also adds the interface where the join message was received to the forwarding-table entry. The host associated with that interface receives multicast traffic for that multicast group.



Router A sends a general query to the device. The device then forwards the query to ports 2 through 5, all members of the same VLAN. Host 1 wants to join multicast group 224.1.2.3 and multicasts an IGMP membership report (IGMP join message) to the group. The device CPU uses the information in the IGMP report to set up a forwarding-table entry that includes the port numbers connected to Host 1 and to the router.

Table 6: IGMP snooping forwarding table

Destination address	Type of packet	Ports
224.1.2.3	IGMP	1, 2

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

Figure 8: Second host joining a multicast group

If another host (for example, Host 4) sends an unsolicited IGMP join message for the same group, the CPU receives that message. It adds the port number of Host 4 to the forwarding table. The forwarding table only directs IGMP messages to the CPU, preventing message flooding to other device ports. Any known multicast

traffic is forwarded to the group, not the CPU.

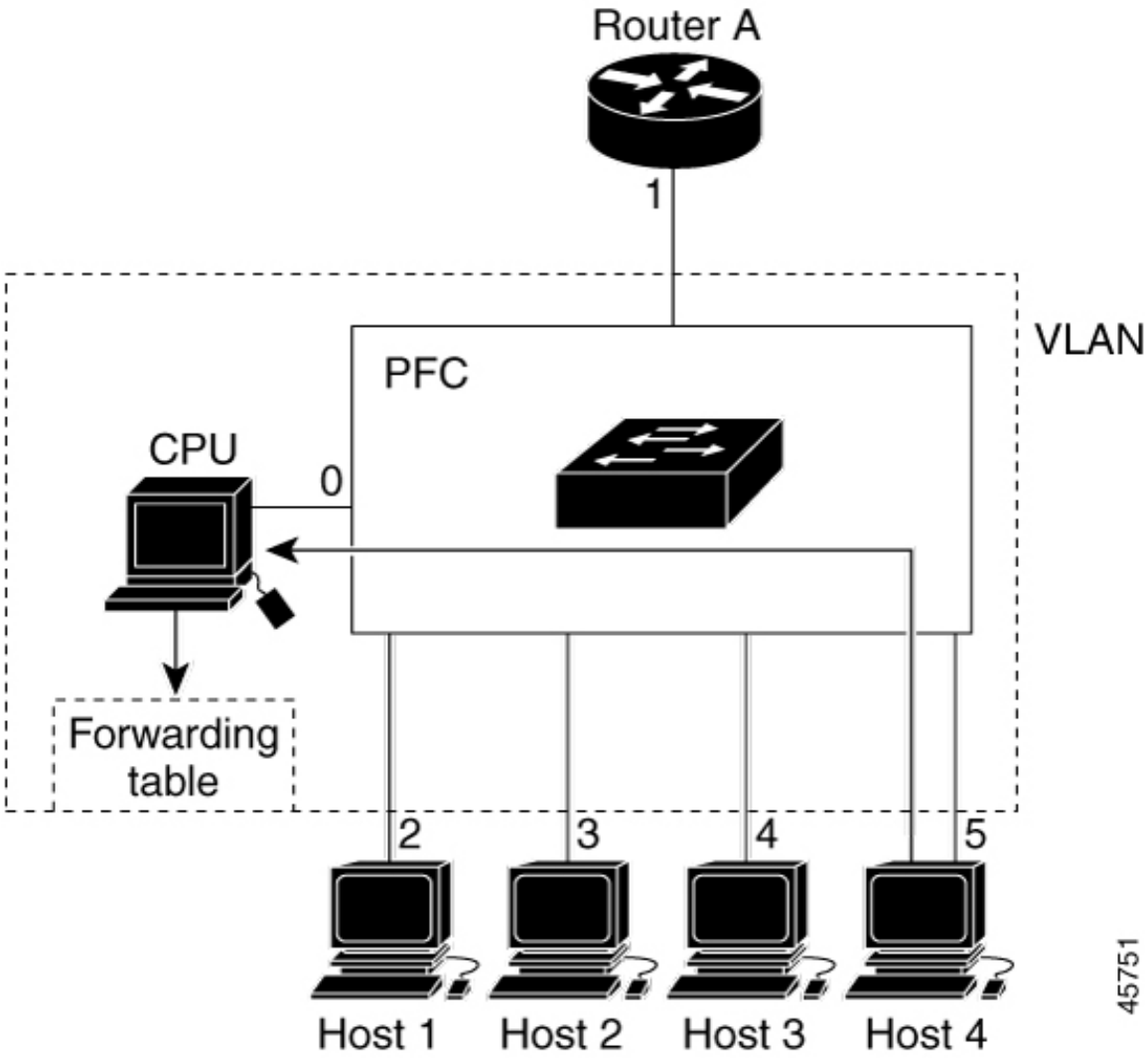


Table 7: Updated IGMP snooping forwarding table

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

Leave a multicast group

The router sends multicast general queries, which the device forwards through the VLAN ports. Interested hosts respond to the queries. If any host in the VLAN opts to receive multicast traffic, the router continues forwarding multicast traffic to the VLAN. IGMP snooping maintains the forwarding table, and the device forwards multicast group traffic only to listed hosts.

Hosts can silently leave a multicast group or send a leave message. When the device receives a leave message from a host, it sends a group-specific query to check if other connected devices on that interface are interested in the specific multicast group traffic. The device then updates the forwarding table for that MAC group so

that only those hosts interested in receiving multicast traffic for the group are listed in the forwarding table. If the router does not get reports from a VLAN, it deletes the group from its IGMP cache.

IGMP leave timer

Configure the device wait time after a group-specific query to determine whether any hosts remain interested in a specific multicast group. The IGMP leave response time can be configured from 100 to 32767 milliseconds.

IGMP report suppression

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

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

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

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

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

IGMP snooping and device stacks

IGMP snooping functions across the device stack; that is, IGMP control information from one device is distributed to all devices in the stack. Regardless of the stack member through which IGMP multicast data enters the stack, the data reaches the hosts that have registered for that group.

If a device in the stack fails or is removed, only the multicast group members on that device will not receive the multicast data. All other members of a multicast group on other devices in the stack continue to receive multicast data streams. However, multicast groups that are common for both Layer 2 and Layer 3 (IP multicast routing) might take longer to converge if the active device is removed.

IGMP filtering and throttling

In some settings, such as metropolitan or multiple-dwelling units (MDUs), you may want to control the multicast groups a user can join on a switch port. You can control the distribution of multicast services, such as IP/TV, based on some type of subscription or service plan. You might also want to limit the number of multicast groups a user on a switch port can join.

With the IGMP filtering feature, you can filter multicast joins on a per-port basis by configuring IP multicast profiles and associating them with individual switch ports. An IGMP profile can contain one or more multicast groups and specifies whether access to the group is permitted or denied. If you apply an IGMP profile denying access to a multicast group on a switch port, the system drops the IGMP join report, and the port cannot receive IP multicast traffic from that group. If access to the multicast group is permitted, the IGMP report from the port will be forwarded for normal processing. You can also set the maximum number of IGMP groups that a Layer 2 interface can join.

IGMP filtering controls only group-specific query and membership reports, such as join and leave reports, but it does not control general IGMP queries. IGMP filtering has no relationship with the function that directs

the forwarding of IP multicast traffic. The filtering feature operates in the same manner whether CGMP or MVR is used to forward the multicast traffic.

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

The IGMP throttling feature lets you set the maximum IGMP groups a Layer 2 interface can join. If the maximum number of IGMP groups is reached, the snooping table contains the maximum entries. When the interface receives an IGMP join report, configure the interface to either drop the report or replace a random multicast entry with it.

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

IGMP explicit tracking

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

The tracking of hosts, groups, and channels allows the device to monitor each host joined to a group or channel. The main benefits of this feature are that it provides minimal leave latencies, faster channel changes, and improved diagnostic capabilities for IGMP.

Minimal leave latencies

Explicit tracking of hosts, groups, and channels in IGMP allows minimal leave latency when a host leaves a multicast group or channel. IGMP leave latency is the time it takes for a device to stop forwarding traffic after a host wants to leave a multicast group. With IGMP Version 3 (IGMPv3) and explicit tracking, the device immediately stops forwarding traffic when the last host indicates it no longer wants to receive traffic. The leave latency is thus bound only by the packet transmission latencies in the multiaccess network and the processing time in the device.

In IGMP Version 2, a device sends an IGMP group-specific query upon receiving a leave message to check if other hosts still request traffic. If no host replies within approximately 3 seconds, the device stops forwarding traffic. This query process is required because, in IGMP Version 1 and 2, IGMP membership reports are suppressed if the same report is already sent by another host in the network. Therefore, it is impossible for the device to reliably know how many hosts on a multiaccess network are requesting to receive traffic.

Faster channel changing

In networks such as xDSL deployments, bandwidth constraints often limit the number of multicast streams that can be received in parallel, typically to N streams. In these deployments, joining only one multicast stream is possible due to bandwidth limitations. The speed at which channels can be changed is determined by the effective leave latency in these environments. You cannot receive the new multicast stream until the old stream has stopped forwarding. If you try to change the channel faster than the leave latency, the application will overload the bandwidth of the access network, and degrade the traffic flow temporarily for all hosts. Explicit tracking in IGMP allows for fast channel changes by enabling minimal leave latencies.

Default IGMP configuration

This table displays the default IGMP configuration for the device.

Table 8: Default IGMP configuration

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

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

Table 9: Default IGMP snooping configuration

Feature	Default Setting
IGMP snooping	Enabled globally and per VLAN.
Multicast routers	None configured.
Static groups	None configured.
TCN[1] flood query count	2
TCN query solicitation	Disabled.
IGMP snooping querier	Disabled.
IGMP report suppression	Enabled.

[1] (1) TCN = Topology Change Notification

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

Table 10: Default IGMP filtering configuration

Feature	Default Setting
IGMP filters	None applied.

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

Prerequisites for IGMP

Follow these guidelines to configure the IGMP snooping querier:

- Configure the VLAN in global configuration mode.
- Configure an IP address on the VLAN interface. When enabled, the IGMP snooping querier uses the IP address as the query source address.
- If there is no IP address configured on the VLAN interface, the IGMP snooping querier uses the configured global IP address. If there is no global IP address specified, the IGMP querier tries to use the VLAN device virtual interface (SVI) IP address (if one exists). If there is no SVI IP address, the device selects the first available IP address. This IP address appears in the **show ip interface** privileged EXEC command output. The IGMP snooping querier does not initiate an IGMP general query if there is no available IP address on the device.
- The IGMP snooping querier supports IGMP Versions 1 and 2.
- When administratively enabled, the IGMP snooping querier moves to the nonquerier state if it detects the presence of a multicast router in the network.
- When it is administratively enabled, the IGMP snooping querier moves to the operationally disabled state under these conditions:
  - IGMP snooping is disabled in the VLAN.
  - PIM is enabled on the SVI of the corresponding VLAN.

Restrictions for IGMP

The restrictions for configuring IGMP include:

- For IGMP Version 3, only IGMP Version 3 BISS (Basic IGMPv3 Snooping Support) is supported.
- IGMPv3 uses new membership report messages that might not be correctly recognized by older IGMP snooping devices.

- IGMPv3 can operate with both ISM and SSM. In ISM, both EXCLUDE and INCLUDE mode reports are applicable. In SSM, the last-hop router accepts only include mode reports and ignores exclude mode reports.
- Using ACLs, designate a specified port as a multicast host port instead of a multicast router port. Multicast router control-packets received on this port are dropped by the system.

The restrictions for configuring IGMP snooping include:

- The device supports IGMPv3 snooping based only on the destination multicast IP address. It does not support snooping based on a source IP address or proxy report.
- Devices running IGMP filtering or Multicast VLAN registration (MVR) do not support IGMPv3 join and leave messages.
- IGMP report suppression is supported only when the multicast query has IGMPv1 and IGMPv2 reports. This feature is not supported when the query includes IGMPv3 reports.
- The IGMP configurable leave time is only supported on hosts running IGMP Version 2. IGMP version 2 is the default version for the device.

  Network leave latency is usually the configured leave time. Variations can occur due to real-time CPU load, network delays, and traffic levels.
- Apply IGMP throttling action restriction only to Layer 2 ports. You can use **ip igmp max-groups action replace** interface configuration command on a logical EtherChannel interface but cannot use it on ports that belong to an EtherChannel port group.

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

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

The restrictions for configuring IGMP explicit tracking include:

- When hosts supporting only IGMP Version 1 or 2 are present, multicast group leave latencies revert to 3 seconds for IGMP Version 2 and up to 180 seconds for IGMP Version 1. This condition affects only the multicast groups that these legacy hosts join. In addition, the membership reports for these multicast groups sent by IGMPv3 hosts may revert to IGMP Version 1 or 2 reports, disabling explicit tracking of those memberships.
- IGMP Version 3 lite (IGMP v3lite) or URL Rendezvous Directory (URD) channel membership reports are not eligible for explicit tracking. Therefore, the leave latency for multicast groups sending traffic to hosts using IGMPv3 lite or URD will be determined by the leave latency of the version of IGMP configured on the hosts (for IGMPv3, the leave latency is typically 3 seconds when explicit tracking is not configured).

Configure IGMP

This section provides configuration information about IGMP.

Configure the device as a member of a group

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

Caution Performing this procedure might impact CPU performance, as the CPU receives all data traffic for the group address.

This procedure is optional.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enabled privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface GigabitEthernet 1/0/1`	Specifies the interface on which you want to enable multicast routing, and enters interface configuration mode.
Step 4	**ip igmp join-group** *group-address* **Example:** `Device(config-if)# ip igmp join-group 225.2.2.2`	Configures the device to join a multicast group. No group memberships are defined by default. For *group-address*, specify the multicast IP address in dotted decimal notation.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 6	**show ip igmp interface** [*interface-id*] **Example:** `Device# show ip igmp interface GigabitEthernet 1/0/1`	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:**	(Optional) Saves your entries in the configuration file.

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

Change the IGMP version

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

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

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

This procedure is optional.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enabled privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet 1/0/1`	Specifies the interface to be configured, and enters the interface configuration mode.
Step 4	**ip igmp version** {**1** | **2** | **3** } **Example:** `Device(config-if)# ip igmp version 2`	Specifies the IGMP version that the switch uses. **Note** If you change to Version 1, you cannot configure the **ip igmp query-interval** or the **ip igmp query-max-response-time** interface configuration commands. To return to the default setting, use the **no ip igmp version** interface configuration command.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

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

Modify the IGMP host-query message interval

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

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

This procedure is optional.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet 1/0/1`	Specifies the interface on which you want to enable multicast routing, and enters interface configuration mode.
Step 4	**ip igmp query-interval** *seconds* **Example:** `Device(config-if)# ip igmp query-interval 75`	Configures the frequency at which the designated router sends IGMP host-query messages.

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

Change the maximum query response time for IGMPv2

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

This procedure is optional.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enabled privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface GigabitEthernet 1/0/1`	Specifies the interface on which you want to enable multicast routing, and enters interface configuration mode.
Step 4	**ip igmp query-max-response-time** *seconds* **Example:**	Changes the maximum query response time advertised in IGMP queries. The default is 10 seconds. The range is 1 to 25.

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

Configure the device as a statically connected member

At various times, a network segment may lack a group member, or a host may be unable to report its group membership using IGMP. You may wish to send multicast traffic to that network segment despite these conditions. The following commands are used to pull multicast traffic down to a network segment:

- **ip igmp join-group**: The device accepts the multicast packets in addition to forwarding them. Accepting the multicast packets prevents the device from fast switching.
- **ip igmp static-group**: The device does not accept the packets itself, but only forwards them. This method enables fast switching. The outgoing interface appears in the IGMP cache, but the device itself is not a member, as evidenced by lack of an L (local) flag in the multicast route entry.

This procedure is optional.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enabled privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface GigabitEthernet`	Specifies the interface on which you want to enable multicast routing, and enters interface configuration mode.

	Command or Action	Purpose
	`1/0/1`	
Step 4	**ip igmp static-group** *group-address* **Example:** `Device(config-if)# ip igmp static-group 239.100.100.101`	Configures the device as a statically connected member of a group. By default, this feature is disabled.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 6	**show ip igmp interface** [*interface-id*] **Example:** `Device# show ip igmp interface GigabitEthernet 1/0/1`	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configure IGMP profiles

Follow these steps to create an IGMP profile:

This task is optional.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enabled privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip igmp profile** *profile number* **Example:** `Device(config)# ip igmp profile 3`	Assigns a number to the profile you are configuring, and enters IGMP profile configuration mode. The profile number range is 1 to 4294967295. When you are in IGMP profile configuration mode, you can create the profile by using these commands:

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

	Command or Action	Purpose
Step 8	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 9	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Apply IGMP profiles

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

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enabled privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface GigabitEthernet 1/0/1`	Specifies the physical interface, and enters interface configuration mode. The interface must be a Layer 2 port that does not belong to an EtherChannel port group.
Step 4	**ip igmp filter** *profile number* **Example:** `Device(config-if)# ip igmp filter 321`	Applies the specified IGMP profile to the interface. The range is 1 to 4294967295. **Note** To remove a profile from an interface, use the **no ip igmp filter** *profile number* interface configuration command.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

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

Set the maximum number of IGMP groups

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

Before you begin

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet1/0/2`	Specifies the interface to be configured, and enters interface configuration mode. The interface can be a Layer 2 port that does not belong to an EtherChannel group or a EtherChannel interface.
Step 4	**ip igmp max-groups** *number* **Example:** `Device(config-if)# ip igmp max-groups 20`	Sets the maximum number of IGMP groups that the interface can join. The range is 0 to 4294967294. The default is to have no maximum set.
Step 5	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 6	**show running-config interface** *interface-id* **Example:** `Device# show running-config interface gigabitethernet1/0/1`	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configure the IGMP throttling action

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

To configure the throttling action when the maximum number of entries is in the forwarding table, follow these steps:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enabled privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet1/0/1`	Specifies the physical interface to be configured, and enters interface configuration mode. The interface can be a Layer 2 port that does not belong to an EtherChannel group or an EtherChannel interface. The interface cannot be a trunk port.
Step 4	**ip igmp max-groups action** {**deny** | **replace**} **Example:** `Device(config-if)# ip igmp max-groups action replace`	When an interface receives an IGMP report and the maximum number of entries is in the forwarding table, the interface specifies the action it takes: • **deny**: Drops the report. If you configure this throttling action, the entries that were previously in the forwarding table are not removed but are aged out. After these entries are aged out and the maximum number of entries is in the forwarding

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

Configure multicast forwarding in absence of directly connected IGMP hosts

Perform this optional task to configure the device to forward multicast traffic in the absence of directly connected IGMP hosts.

Procedure

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

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 1/0/1`	Enters interface configuration mode. • For the *type* and *number* arguments, specify an interface that is connected to hosts.

Step 4	Do one of the following:
	• **ip igmp join-group** *group-address*
	• **ip igmp static-group** {***** | *group-address* [**source** *source-address*]}
	Example:
	`Device(config-if)# ip igmp join-group 225.2.2.2`
	Example:
	`Device(config-if)# ip igmp static-group 225.2.2.2`
		The first sample shows how to configure an interface on the device to join the specified group.
		With this method, the device accepts the multicast packets in addition to forwarding them. Accepting the multicast packets prevents the device from fast switching.
		The second example shows how to configure static group membership entries on an interface. With this method, the device does not accept the packets itself, but only forwards them. Hence, this method allows fast switching. The outgoing interface appears in the IGMP cache, but the device itself is not a member, as evidenced by lack of an "L" (local) flag in the multicast route entry

Step 5	**end** **Example:** `Device#(config-if)# end`	Returns to privileged EXEC mode.
Step 6	**show ip igmp interface** [*interface-type interface-number*] **Example:** `Device# show ip igmp interface`	(Optional) Displays multicast-related information about an interface.

Control access to an SSM network using IGMP extended access lists

Perform this optional task to control access to an SSM network by using an IGMP extended access list that filters SSM traffic based on source address, group address, or both.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast-routing** **Example:** `Device(config)# ip multicast-routing distributed`	Enables IP multicast routing.
Step 4	**ip pim ssm** {**default** | **range** *access-list*} **Example:** `Device(config)# ip pim ssm default`	Configures SSM service. • The **default** keyword defines the SSM range access list as 232/8. • The **range** keyword specifies the standard IP access list number or name that defines the SSM range.
Step 5	**ip access-list extended** *access-list* -name **Example:** `Device(config)# ip access-list extended mygroup`	Specifies an extended named IP access list.
Step 6	**deny igmp** *source source-wildcard destination destination-wildcard* [*igmp-type*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Device(config-ext-nacl)# deny igmp host 10.1.2.3 any`	(Optional) Filters the specified source address or group address from the IGMP report, thereby restricting hosts on a subnet from membership to the (S, G) channel. • Repeat this step to restrict hosts on a subnet membership to other (S, G) channels. (These sources should be more specific than a subsequent **permit** statement because any sources or groups not specifically permitted are denied.) • Remember that the access list ends in an implicit **deny** statement. • This example shows how to create a deny statement that filters all groups for source 10.1.2.3, which effectively denies the source.

	Command or Action	Purpose
Step 7	**permit igmp** *source source-wildcard destination destination-wildcard* [*igmp-type*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Device(config-ext-nacl)# permit igmp any any`	Allows a source address or group address in an IGMP report to pass the IP access list. • You must have at least one **permit** statement in an access list. • Repeat this step to allow other sources to pass the IP access list. • This example shows how to allow group membership to sources and groups not denied by prior **deny** statements.
Step 8	**exit** **Example:** `Device(config-ext-nacl)# exit`	Exits the current configuration session and returns to global configuration mode.
Step 9	interface type number **Example:** `Device(config)# interface ethernet 0`	Selects an interface that is connected to hosts on which IGMPv3 can be enabled.
Step 10	**ip igmp access-group** *access-list* **Example:** `Device(config-if)# ip igmp access-group mygroup`	Applies the specified access list to IGMP reports.
Step 11	**ip pim sparse-mode** **Example:** `Device(config-if)# ip pim sparse-mode`	Enables PIM-SM on the interface. **Note** You must use sparse mode.
Step 12	Repeat Steps 1 through 11 on all interfaces that require access control of SSM channel membership.	--
Step 13	**ip igmp version 3** **Example:** `Device(config-if)# ip igmp version 3`	Enables IGMPv3 on this interface. The default version of IGMP is IGMP version 2. Version 3 is required by SSM.
Step 14	Repeat Step 13 on all host-facing interfaces.	--
Step 15	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Configure IGMP snooping

This section provides configuration information about IGMP snooping.

Enable IGMP snooping

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip igmp snooping** **Example:** `Device(config)# ip igmp snooping`	Globally enables IGMP snooping after it has been disabled.
Step 4	**bridge-domain** *bridge-id* **Example:** `Device(config)# bridge-domain 100`	(Optional) Enters bridge domain configuration mode.
Step 5	**ip igmp snooping** **Example:** `Device(config-bdomain)# ip igmp snooping`	(Optional) Enables IGMP snooping on the bridge domain interface being configured. • Required only if IGMP snooping was previously explicitly disabled on the specified bridge domain.
Step 6	**end** **Example:** `Device(config-bdomain)# end`	Returns to privileged EXEC mode.

Enable or disable IGMP snooping on a VLAN interface

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip igmp snooping vlan** *vlan-id* **Example:** `Device(config)# ip igmp snooping vlan 7`	Enables IGMP snooping on the VLAN interface. The VLAN ID range is 1 to 1001 and 1006 to 4094. IGMP snooping must be globally enabled before you can enable VLAN snooping. **Note** To disable IGMP snooping on a VLAN interface, use the **no ip igmp snooping vlan** *vlan-id* global configuration command for the specified VLAN number.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 5	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Set the snooping method

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

- Snooping on IGMP queries and Protocol-Independent Multicast (PIM) packets.
- Statically connecting to a multicast router port using the **ip igmp snooping mrouter** global configuration command.

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

Procedure

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

Configure a multicast router port

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

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enabled privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip igmp snooping vlan** *vlan-id* **mrouter interface** *interface-id* **Example:** `Device(config)# ip igmp snooping vlan 5 mrouter interface GigabitEthernet 1/0/1`	Specifies the multicast router VLAN ID and the interface to the multicast router. • The VLAN ID range is 1 to 1001 and 1006 to 4094. • The interface can be a physical interface or a port channel. The port-channel range is 1 to 128. **Note** To remove a multicast router port from the VLAN, use the **no ip igmp snooping vlan** *vlan-id* **mrouter interface** *interface-id* global configuration command.
Step 4	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 5	**show ip igmp snooping mrouter** [**vlan** *vlan-id*] **Example:** `Device# show ip igmp snooping mrouter vlan 5`	Verifies that IGMP snooping is enabled on the VLAN interface.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configure a host statically to join a group

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

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enabled privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip igmp snooping vlan** *vlan-id* **static** *ip_address* **interface** *interface-id* **Example:** `Device(config)# ip igmp snooping vlan 105 static 230.0.0.1 interface gigabitethernet1/0/1`	Statically configures a Layer 2 port as a member of a multicast group: • *vlan-id* is the multicast group VLAN ID. The range is 1 to 1001 and 1006 to 4094. • *ip-address* is the group IP address. • *interface-id* is the member port. It can be a physical interface or a port channel (1 to 128). **Note** To remove the Layer 2 port from the multicast group, use the **no ip igmp snooping vlan** *vlan-id* **static** *mac-address* **interface** *interface-id* global configuration command.
Step 4	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 5	**show ip igmp snooping groups** **Example:** `Device# show ip igmp snooping groups`	Verifies the member port and the IP address.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configure the IGMP leave timer

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enabled privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip igmp snooping last-member-query-interval** *time* **Example:** `Device(config)# ip igmp snooping last-member-query-interval 1000`	Configures the IGMP leave timer globally. The range is 100 to 32767 milliseconds. The default leave time is 1000 milliseconds. **Note** To globally reset the IGMP leave timer to the default setting, use the **no ip igmp snooping last-member-query-interval** global configuration command.
Step 4	**ip igmp snooping vlan** *vlan-id* **last-member-query-interval** *time* **Example:** `Device(config)# ip igmp snooping vlan 210 last-member-query-interval 1000`	(Optional) Configures the IGMP leave time on the VLAN interface. The range is 100 to 32767 milliseconds. **Note** Configuring the leave time on a VLAN overrides the globally configured timer. **Note** To remove the configured IGMP leave-time setting from the specified VLAN, use the **no ip igmp snooping vlan** *vlan-id* **last-member-query-interval** global configuration command.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 6	**show ip igmp snooping** **Example:** `Device# show ip igmp snooping`	(Optional) Displays the configured IGMP leave time.
Step 7	**copy running-config startup-config** **Example:**	(Optional) Saves your entries in the configuration file.

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

Configure the IGMP robustness-variable

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

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enabled privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip igmp snooping robustness-variable** *count* **Example:** `Device(config)# ip igmp snooping robustness-variable 3`	Configures the IGMP robustness variable. The range is 1 to 3 times. The recommended value for the robustness variable is 2. Use this command to change the value of the robustness variable for IGMP snooping from the default (2) to a specified value.
Step 4	**ip igmp snooping vlan** *vlan-id* **robustness-variable** *count* **Example:** `Device(config)#ip igmp snooping vlan 100 robustness-variable 3`	(Optional) Configures the IGMP robustness variable on the VLAN interface. The range is 1 to 3 times. The recommended value for the robustness variable is 2. **Note** Configuring the robustness variable count on a VLAN overrides the globally configured value.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 6	**show ip igmp snooping** **Example:** `Device# show ip igmp snooping`	(Optional) Displays the configured IGMP robustness variable count.

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

Configure the IGMP last member query count

Use this procedure to set how many times the device should send IGMP group-specific or group-source-specific query messages when it receives a leave message.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enabled privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip igmp snooping last-member-query-count** *count* **Example:** `Device(config)# ip igmp snooping last-member-query-count 3`	Configures the IGMP last member query count. The range is 1 to 7 messages. The default is 2 messages.
Step 4	**ip igmp snooping vlan** *vlan-id* **last-member-query-count** *count* **Example:** `Device(config)#ip igmp snooping vlan 100 last-member-query-count 3`	(Optional) Configures the IGMP last member query count on the VLAN interface. The range is 1 to 7 messages. **Note** Configuring the last member query count on a VLAN overrides the globally configured timer.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 6	**show ip igmp snooping** **Example:** `Device# show ip igmp snooping`	(Optional) Displays the configured IGMP last member query count.

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

Configure TCN-related commands

This section provides configuration information about TCN.

Control the multicast flood time after a TCN event

You can configure the number of general queries by which multicast data traffic is flooded after a topology change notification (TCN) event. If you set the TCN flood query count to 1 the flooding stops after receiving 1 general query. If you set the count to 7, the flooding continues until 7 general queries are received. Groups are relearned based on the general queries received during the TCN event.

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

Follow these steps to configure the TCN flood query count:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enabled privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip igmp snooping tcn flood query count** *count* **Example:** `Device(config)# ip igmp snooping tcn flood query count 3`	Specifies the number of IGMP general queries for which the multicast traffic is flooded. The range is 1 to 10. The default, the flooding query count is 2. **Note** To return to the default flooding query count, use the **no ip igmp snooping tcn flood query count** global configuration command.
Step 4	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

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

Recover from flood mode

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

Follow these steps to enable sending of leave messages:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip igmp snooping tcn query solicit** **Example:** `Device(config)# ip igmp snooping tcn query solicit`	Sends an IGMP leave message (global leave) to speed the process of recovering from the flood mode caused during a TCN event. By default, query solicitation is disabled. **Note** To return to the default query solicitation, use the **no ip igmp snooping tcn query solicit** global configuration command.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 5	**show ip igmp snooping** **Example:**	Verifies the TCN settings.

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

Disable multicast flood during a TCN event

When the device receives a TCN, multicast traffic is flooded to all STP non-edge ports until 2 general queries are received. The device does not flood multicast traffic to STP edge ports after STP TCN events. If the device has many ports with attached hosts that are subscribed to different multicast groups, this flooding might exceed the capacity of the link and cause packet loss. You can use the **no** form of **ip igmp snooping tcn flood** interface configuration command to control this behavior.

Follow these steps to disable multicast flooding on an interface:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enabled privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface GigabitEthernet 1/0/1`	Specifies the interface to be configured, and enters interface configuration mode.
Step 4	**no ip igmp snooping tcn flood** **Example:** `Device(config-if)# no ip igmp snooping tcn flood`	Disables the flooding of multicast traffic during a spanning-tree TCN event. By default, multicast flooding is enabled on an interface. **Note** To re-enable multicast flooding on an interface, use the **ip igmp snooping tcn flood** interface configuration command.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 6	**show ip igmp snooping** **Example:** `Device# show ip igmp snooping`	Verifies the TCN settings.
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configure the IGMP snooping querier

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enabled privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip igmp snooping querier** **Example:** `Device(config)# ip igmp snooping querier`	Enables the IGMP snooping querier.
Step 4	**ip igmp snooping querier address** *ip_address* **Example:** `Device(config)# ip igmp snooping querier address 172.16.24.1`	(Optional) Specifies an IP address for the IGMP snooping querier. If you do not specify an IP address, the querier tries to use the global IP address configured for the IGMP querier. **Note** The IGMP snooping querier does not generate an IGMP general query if it cannot find an IP address on the device.
Step 5	**ip igmp snooping querier query-interval** *interval-count* **Example:** `Device(config)# ip igmp snooping querier query-interval 30`	(Optional) Sets the interval between IGMP queriers. The range is 1 to 18000 seconds.

	Command or Action	Purpose
Step 6	**ip igmp snooping querier tcn query** [**count** *count* | **interval** *interval*] **Example:** `Device(config)# ip igmp snooping querier tcn query interval 20`	(Optional) Sets the time between Topology Change Notification (TCN) queries. The count range is 1 to 10. The interval range is 1 to 255 seconds.
Step 7	**ip igmp snooping querier timer expiry** *timeout* **Example:** `Device(config)# ip igmp snooping querier timer expiry 180`	(Optional) Sets the length of time until the IGMP querier expires. The range is 60 to 300 seconds.
Step 8	**ip igmp snooping querier version** *version* **Example:** `Device(config)# ip igmp snooping querier version 2`	(Optional) Selects the IGMP version number that the querier feature uses. Select 1 or 2.
Step 9	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 10	**show ip igmp snooping vlan** *vlan-id* **Example:** `Device# show ip igmp snooping vlan 30`	(Optional) Verifies that the IGMP snooping querier is enabled on the VLAN interface. The VLAN ID range is 1 to 1001 and 1006 to 4094.
Step 11	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Disable IGMP report suppression

Follow these steps to disable IGMP report suppression:

Procedure

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

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

Configure IGMP explicit tracking

This section provides configuration information about IGMP explicit tracking.

Enable explicit tracking globally

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

Procedure

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

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

Enable explicit tracking on Layer 3 interfaces

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface vlan 77`	Configures an interface and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.1.1.1 255.255.255.254`	Sets a primary or secondary IP address for an interface.
Step 5	**ip pim sparse-mode** **Example:** `Device(config-if)# ip pim sparse-mode`	Enables Protocol Independent Multicast (PIM) sparse mode on an interface.
Step 6	**ip igmp version 3** **Example:** `Device(config-if)# ip igmp version 3`	Configure Internet Group Management Protocol (IGMP) Version 3 (IGMPv3) on the device.

	Command or Action	Purpose
Step 7	**ip igmp explicit-tracking** **Example:** `Device(config-if)# ip igmp explicit-tracking`	Enables IGMP explicit host tracking.
Step 8	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuration examples

Refer this section for configuration examples of IGMP and IGMP snooping.

Example: Configure the device as a member of a multicast group

This example shows how to enable the device to join multicast group 10.11.1.1:

```
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# ip igmp join-group 10.11.1.1
Device(config-if)#
```

Example: Control access to multicast groups

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

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

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

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

Example: Configure IGMP snooping

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

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

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

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

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

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

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

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

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

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

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

```
Device# configure terminal
Device(config)# ip igmp snooping querier version 2
Device(config)# end
```

Example: Configure IGMP profiles

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

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

IGMP Profile 4
    permit
    range 229.9.9.0 229.9.9.0
```

Example: Apply IGMP profile

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

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

Example: Set the maximum number of IGMP groups

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

```
Device(config)# interface Gigabitethernet1/0/2
Device(config-if)# ip igmp max-groups 25
Device(config-if)# end
```

Example: Interface configuration as a routed port

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

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

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

Example: Interface configuration as an SVI

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

```
Device(config)# interface vlan 150
Device(config-if)# ip address 10.20.20.1 255.255.255.0
Device(config-if)# ip pim sparse-mode
Device(config-if)#  ip igmp join-group 224.1.2.3 source 15.15.15.2
Device(config-if)# end
Device# configure terminal
Device(config)# ip igmp snooping vlan 20 static 224.1.2.3 interface gigabitEthernet 1/0/9
Device# show run interface vlan 150

Current configuration : 137 bytes
!
interface vlan 150
 ip address 10.20.20.1 255.255.255.0
 ip pim sparse-mode
 ip igmp static-group 224.1.2.3 source 15.15.15.2
end
```

Example: Configure multicast forwarding in absence of directly connected IGMP hosts

This example shows how to configure a device to forward multicast traffic in the absence of directly connected IGMP hosts using the **ip igmp join-group** command. With this method, the device accepts the multicast packets in addition to forwarding them. Accepting the multicast packets prevents the device from fast switching.

In this example, GigabitEthernet interface 1/0/1 on the device is configured to join the group 225.2.2.2:

```
interface GigabitEthernet1/0/1
 ip igmp join-group 225.2.2.2
```

This example shows how to configure a device to forward multicast traffic in the absence of directly connected IGMP hosts using the **ip igmp static-group** command. With this method, the device does not accept the packets itself, but only forwards them. Hence, this method allows fast switching. The outgoing interface appears in the IGMP cache, but the device itself is not a member, as evidenced by lack of an "L" (local) flag in the multicast route entry.

In this example, static group membership entries for group 225.2.2.2 are configured on Fast Ethernet interface 0/1/0:

```
interface GigabitEthernet1/0/1
 ip igmp static-group 225.2.2.2
```

Example: Control access to an SSM network using IGMP extended access lists

This section contains configuration examples for controlling access to an SSM network using IGMP extended access lists:

Note Access lists offer flexibility with numerous combinations of permit and deny statements to filter multicast traffic. This section includes examples of how to implement these configurations.

Example: Deny all states for a group G

This example shows how to deny all states for a group G. In this example, Fast Ethernet interface 0/0/0 is configured to filter all sources for SSM group 232.2.2.2 in IGMPv3 reports, which effectively denies this group.

```
ip access-list extended test1
 deny igmp any host 232.2.2.2
 permit igmp any any
!
interface GigabitEthernet 1/0/1
 ip igmp access-group test1
```

Example: Deny all states for a source S

This example shows how to deny all states for a source S. In this example, Gigabit Ethernet interface 1/1/0 is configured to filter all groups for source 10.2.1.32 in IGMPv3 reports, which effectively denies this source.

```
ip access-list extended test2
 deny igmp host 10.2.1.32 any
 permit igmp any any
!
interface GigabitEthernet1/0/1
 ip igmp access-group test2
```

Example: Permit all states for a group G

This example shows how to permit all states for a group G. In this example, Gigabit Ethernet interface 1/2/0 is configured to accept all sources for SSM group 232.1.1.10 in IGMPv3 reports, which effectively accepts this group altogether.

```
ip access-list extended test3
 permit igmp any host 232.1.1.10
!
interface GigabitEthernet 1/2/0
 ip igmp access-group test3
```

Example: Permit all states for a source S

This example shows how to permit all states for a source S. In this example, Gigabit Ethernet interface 1/2 is configured to accept all groups for source 10.6.23.32 in IGMPv3 reports, which effectively accepts this source altogether.

```
ip access-list extended test4
 permit igmp host 10.6.23.32 any
!
interface GigabitEthernet1/2/0
 ip igmp access-group test4
```

Example: Filter a source S for a group G

This example shows how to filter a particular source S for a group G. In this example, Gigabit Ethernet interface 0/3/0 is configured to filter source 232.2.2.2 for SSM group 232.2.30.30 in IGMPv3 reports.

```
ip access-list extended test5
 deny igmp host 10.4.4.4 host 232.2.30.30
 permit igmp any any
!
interface GigabitEthernet0/3/0
 ip igmp access-group test5
```

Monitor IGMP

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

Note Per-route statistics are not supported.

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

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

Table 11: Commands for displaying system and network statistics

Command	Purpose
show ip igmp filter	Displays IGMP filter information.
show ip igmp groups [*type-number* | *detail*]	Displays the multicast groups that are directly connected to the device and that were learned through IGMP.
show ip igmp interface [*type number*]	Displays multicast-related information about an interface.
show ip igmp membership [*name/group address* | **all** | **tracked**]	Displays IGMP membership information for forwarding.
show ip igmp profile [*profile_number*]	Displays IGMP profile information.
show ip igmp ssm-mapping [*hostname/IP address*]	Displays IGMP SSM mapping information.
show ip igmp static-group {**class-map** [**interface** [*type*]]	Displays static group information.
show ip igmp vrf	Displays the selected VPN routing/forwarding instance by name. **Note** The **show ip igmp vrf** *vrf-name* **snooping groups** command ignores the **vrf** keyword and displays the snooping information for the VLANs. Use the **show ip igmp snooping groups** command to see the IGMP snooping information for the VLANs.

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

Table 12: Commands for displaying IGMP snooping information

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

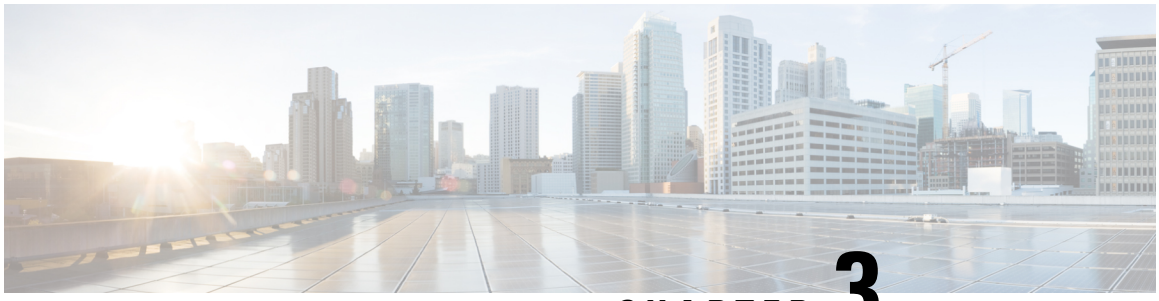
Command	Purpose
show ip igmp snooping groups [**count** |**dynamic** [**count**] | **user** [**count**]]	Displays multicast table information for the device or about a specific parameter: • **count**: Displays the total number of entries for the specified command options instead of the actual entries. • **dynamic**: Displays entries learned through IGMP snooping. • **user**: Displays only the user-configured multicast entries.
show ip igmp snooping groups [**count** | [**vlan** *vlan-id* [*A.B.C.D* | **count**]]	Displays multicast table information for the device or about a specific parameter: • **count**: Displays the total number of groups. • **vlan**: Displays group information by VLAN ID.
show ip igmp snooping groups vlan *vlan-id* [*ip_address* | **count** | **dynamic** [**count**] | **user**[**count**]]	Displays multicast table information for a multicast VLAN or about a specific parameter for the VLAN: • *vlan-id*: The VLAN ID range is 1 to 1001 and 1006 to 4094. • **count**: Displays the total number of entries for the specified command options instead of the actual entries. • **dynamic**: Displays entries learned through IGMP snooping. • *ip_address*: Displays characteristics of the multicast group with the specified group IP address. • **user**: Displays only the user-configured multicast entries.
show ip igmp snooping mrouter [**vlan** *vlan-id*]	Displays information on dynamically learned and manually configured multicast router interfaces. **Note** When you enable IGMP snooping, the device automatically learns the interface to which a multicast router is connected. These are dynamically learned interfaces. (Optional) Enter **vlan** *vlan-id* to display information for a single VLAN.

Command	Purpose
show ip igmp snooping querier [**detail** | **vlan** *vlan-id*]	Displays information about the IP address and receiving port for the most-recently received IGMP query messages in the VLAN. (Optional) Enter **detail** to display the detailed IGMP querier information in a VLAN. (Optional) Enter **vlan** *vlan-id* to display information for a single VLAN.
show ip igmp snooping querier [**vlan** *vlan-id*] **detail**	Displays information about the IP address and receiving port of the most-recently received IGMP query message in the VLAN and the configuration and operational state of the IGMP snooping querier in the VLAN.
show ip igmp snooping [**vlan** *vlan-id* [**detail**]]	Displays the snooping configuration information for all VLANs on the device or for a specified VLAN. (Optional) Enter **vlan** *vlan-id* to display information for a single VLAN. The VLAN ID range is 1 to 1001 and 1006 to 4094.

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

Table 13: Commands for displaying IGMP filtering and throttling configuration

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



CHAPTER 3

PIM

Feature history for PIM

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature name and description	Supported platform
Cisco IOS XE 17.18.1	PIM: PIM is IP routing protocol-independent and operates independently of any specific unicast routing protocol.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

Understand PIM

The Protocol Independent Multicast (PIM) protocol maintains the current IP multicast service mode of receiver-initiated membership. PIM operates independently of any specific unicast routing protocol. It is IP routing protocol-independent. PIM uses available unicast routing protocols to populate the unicast routing table, including Enhanced Interior Gateway Routing Protocol (EIGRP), Open Shortest Path First (OSPF), ECMP, 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. PIM differs from other routing protocols as it does not send or receive routing updates between routers.

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

PIM versions

PIMv2 includes these improvements over PIMv1:

- A single, active rendezvous point (RP) exists per multicast group, with multiple backup RPs. This single RP compares to multiple active RPs for the same group in PIMv1.
- A bootstrap router (BSR) provides a fault-tolerant, automated RP discovery and distribution function that enables routers and multilayer switches to dynamically learn the group-to-RP mappings.
- PIM join and prune messages have more flexible encoding for multiple address families.
- A more flexible hello packet format replaces the query packet to encode current and future capability options.
- Register messages sent to an RP specify whether they are sent by a border router or a designated router.
- PIM packets are no longer inside IGMP packets; they are standalone packets.

Multicast source discovery protocol

Multicast Source Discovery Protocol (MSDP) is used for inter-domain source discovery when PIM SM is used. Each PIM administrative domain has its own RP. MSDP signals new sources between RPs across different domains.

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

If RPs in other domains have join requests for the group's SA message (shown by a (*,G) entry with a non-empty outgoing interface list), the domain is interested in the group. The RP then triggers an (S,G) join toward the source.

PIM sparse mode

PIM sparse mode (PIM-SM) is a multicast routing protocol designed to efficiently route IP multicast traffic in networks where receivers are sparsely distributed. Unlike dense mode protocols that flood multicast traffic to all parts of the network initially, PIM Sparse Mode uses a more controlled approach to conserve bandwidth and resources.

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

Sparse mode interfaces are added to the multicast routing table when Join messages arrive from downstream routers or when a connected member is on the interface. When forwarding from a LAN, sparse mode operation happens only if an RP is recognized for the group. If so, the packets are encapsulated and sent toward the RP. When there is sufficient multicast traffic from a source, the receiver's first hop router may send Join messages toward the source to create 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). You must configure the RP in the network.

In sparse mode, routers do not forward multicast packets for a group unless an explicit request for traffic is received. When hosts join a multicast group, the directly connected routers send PIM Join messages toward the RP. The RP keeps track of multicast groups. Hosts that send multicast packets are registered with the RP by the first hop router of that host. The RP then sends Join messages toward the source, and 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. Edge routers learn about a particular source when they receive data packets that travel on the shared tree from that source through the RP. The edge router then sends PIM (S,G) Join messages toward that source. Each router on the reverse path compares the RP address's unicast metric to the source address's metric. If the metric for the source address is better, it will forward a PIM (S,G) Join message toward the source. If the metric for the RP is the same or better, then the PIM (S,G) Join message will be sent in the same direction as the RP. In this case, the shared tree and the source tree would be considered congruent.

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

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

PIM stub routing

The PIM stub routing feature moves routed traffic closer to the end user to reduce resource usage. This feature supports multicast routing between the distribution layer and the access layer. It supports two types of PIM interfaces: uplink PIM interfaces and PIM passive interfaces. A routed interface configured with the PIM passive mode does not pass or forward PIM control traffic; it only passes and forwards IGMP traffic.

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

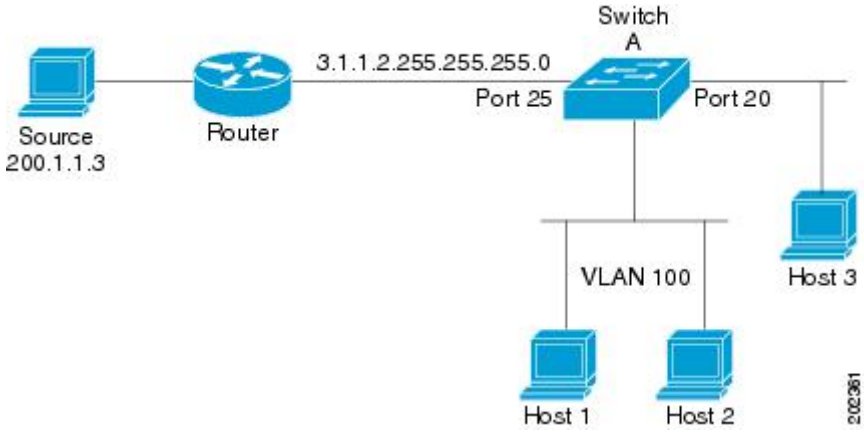
When using PIM stub routing, you should configure the distribution and remote routers to use IP multicast routing. Configure only the device as a PIM stub router. The device does not route transit traffic between distribution routers. You also need to configure a routed uplink port on the device. The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

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

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

Figure 9: PIM stub router configuration

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



Rendezvous points

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

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. This occurs because sources need only periodically register with the RP to create state.

Auto-RP

Auto-RP is a PIM-SM feature that:

- automates the distribution of group-to-RP mappings in a PIM network,
- allows easy configuration of multiple RPs to serve different groups and enables load splitting, and
- prevents inconsistent, manual RP configurations that may cause connectivity issues.

In the initial version of PIM-SM, static RP configuration required manual RP address entry on leaf routers. Auto-RP simplifies this by allowing designated RP-mapping agents to manage announcements and resolve conflicts, enabling automatic group-to-RP discovery across the network.

The RP-mapping agent receives RP-announcements, adjudicates any discrepancies, and sends consistent mappings to other routers. This automated process is crucial for large, complex networks where manual configuration is tedious.

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 use Auto-RP, designate a router as an RP mapping agent to receive RP announcements and arbitrate conflicts. Thus, all routers automatically discover which RP to use for the groups. 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 strengths, weaknesses, and complexity. In conventional IP multicast network scenarios, use Auto-RP to configure RPs because it is easy to set up, well-tested, and stable. The alternative ways to configure an RP are static RP, Auto-RP, and bootstrap router.

Auto-RP in a PIM network

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

You can automatically discover the RP to use for the groups you support. The 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.

Benefits of Auto-RP in a PIM network

- Auto-RP enables changes to the RP designation to be configured exclusively on RP devices and not on leaf routers.
- Auto-RP allows the scoping of the RP address within a domain.

Auto-RP sparse-dense mode

An interface configured in sparse-dense mode operates in sparse or dense mode, based on the multicast group's mode. 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. Configure all interfaces in sparse-dense mode using the **ip pim sparse-dense-mode** interface configuration command for Auto-RP.

We recommend configuring a sink RP (also known as RP of last resort) 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. 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 an unknown or unexpected source can 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 boundaries

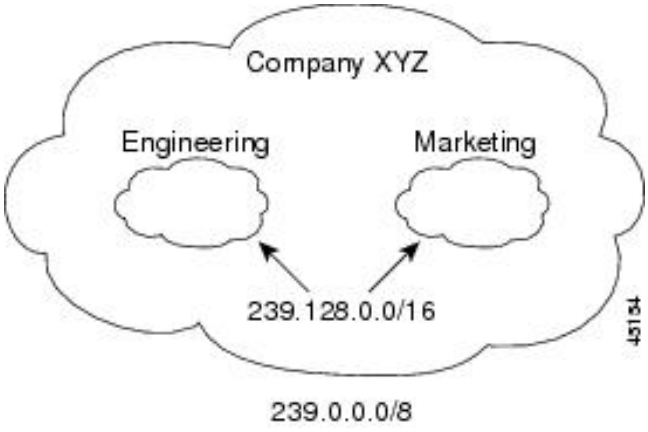
Multicast boundaries are used in multicast routing to control and limit the scope of multicast traffic within a network. They act as filters or boundaries that prevent multicast traffic from crossing certain points in the network, thereby containing multicast traffic to specific areas and reducing unnecessary load on routers and links outside those areas.

Use administratively-scoped boundaries to limit multicast traffic forwarding outside a domain or subdomain. This method uses a specific range of multicast addresses, named administratively-scoped addresses, to create the boundary mechanism. If you configure an administratively-scoped boundary on a routed interface, multicast traffic whose multicast group addresses fall in this range cannot enter or exit this interface, which provides a firewall for multicast traffic in this address range.

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

Figure 10: Administratively-scoped boundaries

This figure shows that Company XYZ has an administratively-scoped boundary set for the multicast address range 239.0.0.0/8 on all routed interfaces at the perimeter of its network. This boundary prevents any multicast traffic in the range 239.0.0.0 through 239.255.255.255 from entering or leaving the network. Similarly, the engineering and marketing departments have an administratively-scoped boundary of 239.128.0.0/16 around the perimeter of their networks. This boundary prevents multicast traffic in the range of 239.128.0.0 through 239.128.255.255 from entering or leaving their respective networks.



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

The IANA has designated the multicast address range 239.0.0.0 to 239.255.255.255 as the administratively-scoped addresses. The addresses can be reused in domains managed by different organizations. The addresses would be considered local, not globally unique.

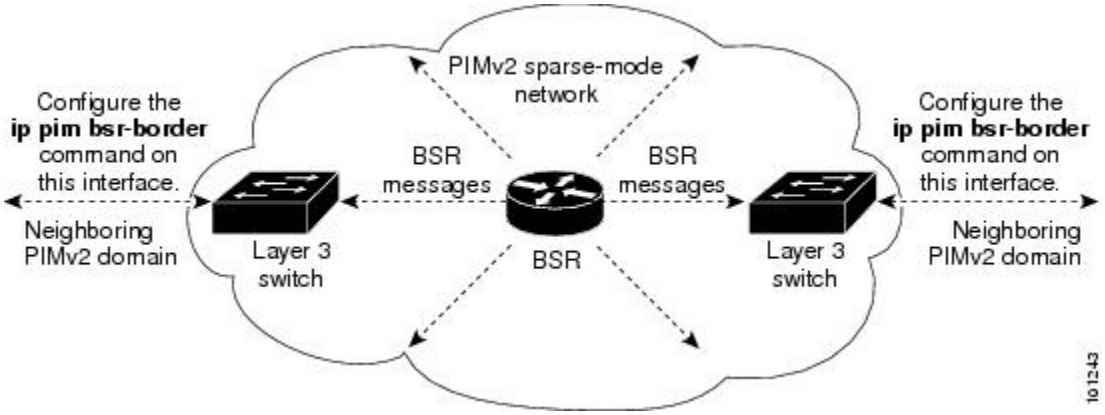
You can configure the **filter-autorp** keyword to examine and filter Auto-RP discovery and announcement messages at the administratively scoped boundary. Any Auto-RP group range announcements from the Auto-RP packets that are denied by the boundary access control list (ACL) are removed. The boundary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

and passes an Auto-RP group range announcement only if all addresses in the Auto-RP group range are allowed 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.

PIM domain border

As IP multicast becomes more widespread, the chance of one PIMv2 domain bordering another PIMv2 domain increases. Because two domains probably do not share the same set of RPs, BSR, candidate RPs, and candidate BSRs, you need to constrain PIMv2 BSR messages from flowing into or out of the domain. If messages leak across the domain borders, it could negatively impact the normal BSR election process by electing a single BSR for all bordering domains and mixing candidate RP advertisements, which may lead to the election of RPs in the incorrect domain.

This figure displays how you can configure the PIM domain border by using the **ip pim bsr-border** command.



PIMv2 bootstrap router

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

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

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

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

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. This tree connects all sources to all receivers in the group and may either be shared by all sources (a shared tree) or be built separately for each source (a source tree).

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 these:

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

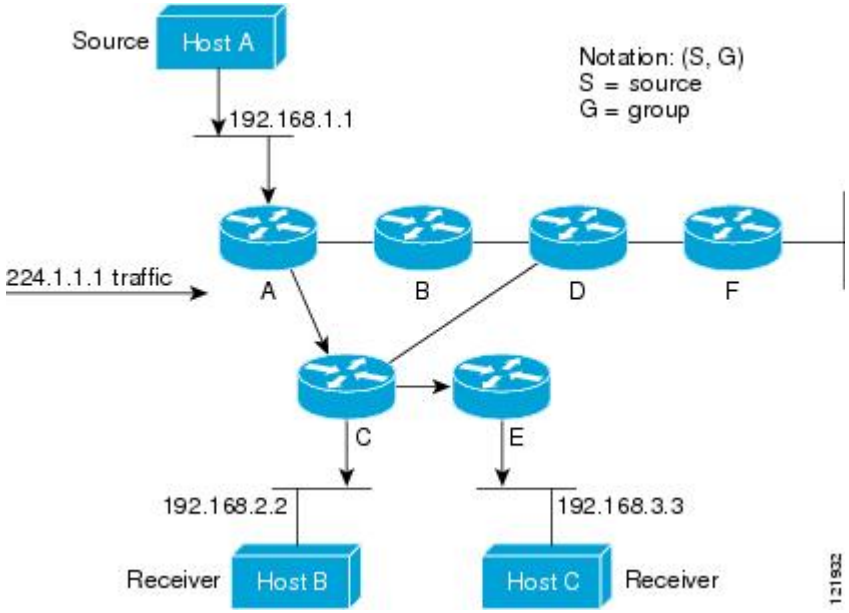
The notation of (S,G), pronounced “S comma G,” enumerates a shortest path tree where S is the IP address of the source and G is the multicast group address.

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. This tree is referred to as a shortest path tree (SPT) because it uses the shortest path through the network.

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



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

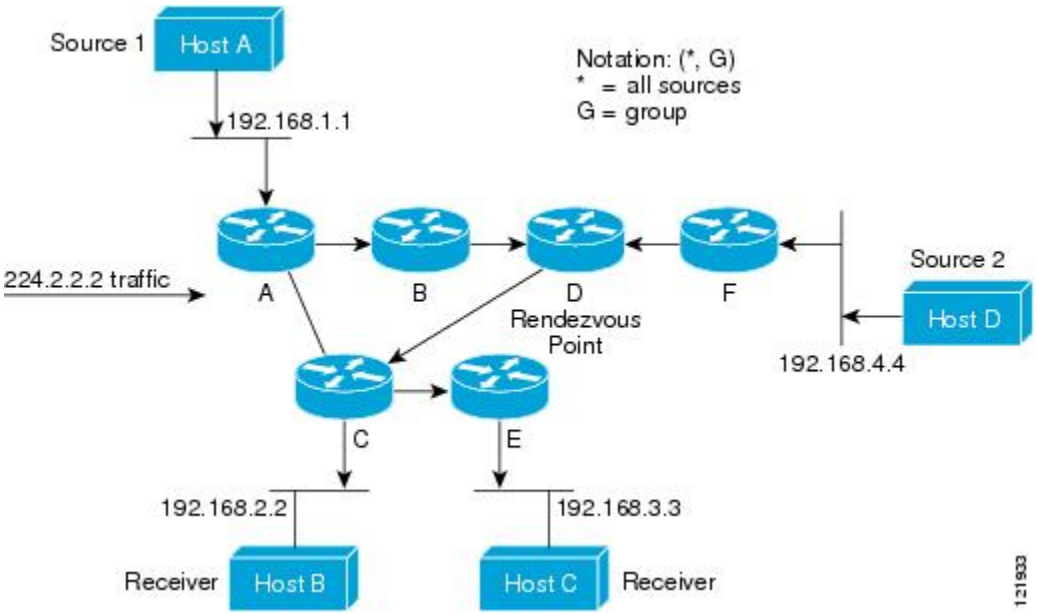
The (S,G) notation implies that a separate SPT exists for each individual source sending to each group--which is correct.

Multicast distribution shared tree

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

This 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 the receivers, except when the receiver is located between the source and the RP in which case it will be serviced directly.

Figure 11: Shared tree



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

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

Source tree advantage

Source trees have the advantage of creating the optimal path between the source and the receivers. This advantage reduces network latency for multicast traffic. However, this optimization comes at a cost. The routers must maintain path information for each source. In large networks consisting of many sources and groups, this overhead can quickly become a resource issue for routers. Network designers must consider how the size of the multicast routing table affects memory consumption.

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. Shared trees can have non-optimal paths between the source and receivers, which may introduce 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. Carefully consider where to place 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 its routing table for the destination address and forwards a single unicast packet toward the destination.

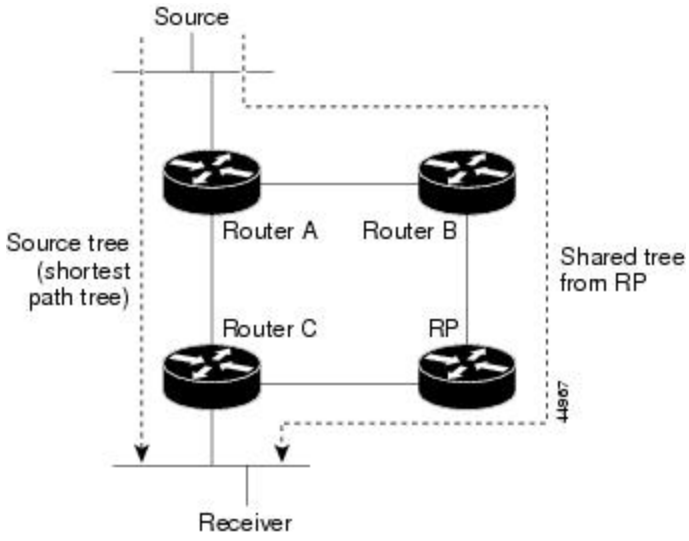
Multicast forwarding involves routing traffic to a group of hosts identified by a multicast group address. The multicast router must determine the upstream direction (toward the source) and the downstream direction (or directions) toward the receivers. If there are multiple downstream paths, the router 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).

PIM shared tree and source tree

By default, you receive data from senders routed through a single data-distribution tree rooted at the RP.

Figure 12: Shared tree and source tree (shortest-path tree)

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



If the data rate warrants, leaf routers (routers without any downstream connections) on the shared tree can use the data distribution tree rooted at the source. This type of distribution tree is called a shortest-path tree or source tree. By default, the software transitions to a source tree upon receiving the first data packet from a source.

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

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

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

The shared tree is used by multiple sources sending to groups. You can configure the PIM device to stay on the shared tree.

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

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

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

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

Reverse path forwarding

Reverse Path Forwarding (RPF) forwards multicast traffic away from the source instead of to the receiver. RPF is an algorithm used for forwarding multicast datagrams.

PIM creates a distribution tree using unicast routing information along the reverse path from receivers to 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. Routers forward multicast packets only if they are received on the upstream interface. This RPF check helps to guarantee that the distribution tree will be loop-free.

RPF check

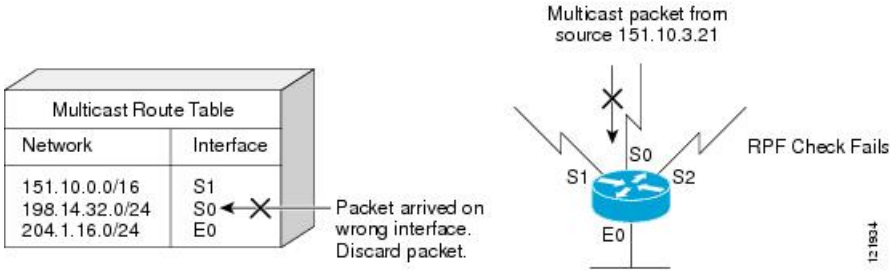
When a multicast packet arrives at a router, the router performs an RPF check on the packet. If the RPF check succeeds, the packet is forwarded. Otherwise, it is dropped.

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.

An example of an unsuccessful RPF check is shown in the figure.

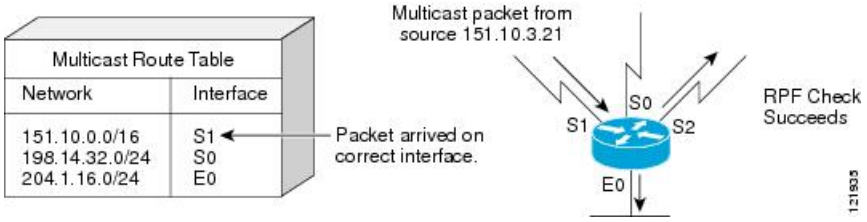
Figure 13: RPF check fails



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

The figure shows an example of a successful RPF check.

Figure 14: RPF check succeeds



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

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

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

Note DVMRP is not supported on the switch.

Sparse-mode PIM uses the RPF lookup function to decide where it needs to send joins and prunes:

- (S, G) joins (which are source-tree states) are sent toward the source.
- (*,G) joins (which are shared-tree states) are sent toward the RP.

High availability on PIM

Note This feature is applicable to Tenant Routed Multicast (TRM) deployments only.

High availability on PIM feature improves the multicast convergence time after Stateful Switchover (SSO) in a chassis with dual supervisors, StackWise Virtual Link (SVL) and Stacking devices.

After SSO, the PIM protocol generates a new ID and populates multicast route states using the PIM join/prune or IGMP report messages from downstream routers. The Join message populates the Source IP, Multicast Group address, and Outgoing Interface list. For incoming interface, PIM places a route watch request with Rendezvous Point (RP) address as the prefix for (*, G) entries, and source address as the prefix for (S,G) entries.

Once the unicast Routing Information Base (RIB) converges, the route watch update provides the Reverse Path Forward (RPF) interface and RPF neighbor address details. Reverse Path Forward (RPF) checking ensures that multicast traffic arrives on the expected router interface before further processing. If multicast packets fail the RPF check, they are discarded.

Now the PIM has complete multicast route information to program the forwarding table and to send a PIM join/prune message towards upstream PIM neighbor. In multicast deployments where unicast RIB convergence takes more than 3 minutes after SSO, PIM running in the new active device does not have incoming interface to program the forwarding table and RPF neighbor address to send join/prune messages towards upstream PIM neighbor. The time interval of 3 minutes is determined by the default value of the PIM join/prune interval.

If PIM join/prune message is not received, the upstream PIM neighbor removes outgoing interface from the multicast route entry. This affects multicast traffic depending on unicast protocol convergence time. The standby device stores the RPF interface and RPF neighbor address details to improve multicast traffic convergence. After SSO, the new active device uses this stored RPF information to program the forwarding table until the unicast RIB converges. It also sends a join/prune message to the upstream PIM neighbor.

Default PIM configuration

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

Table 14: Default PIM routing configuration

Feature	Default setting
Multicast routing	Disabled on all interfaces.
PIM version	Version 2.
PIM mode	No mode is defined.
PIM stub routing	None configured.
PIM RP address	None configured.
PIM domain border	Disabled.
PIM multicast boundary	None.
Candidate BSRs	Disabled.
Candidate RPs	Disabled.
Shortest-path tree threshold rate	0 kb/s.
PIM router query message interval	30 seconds.

Prerequisites for PIM

Decide which PIM mode you will use before starting the PIM configuration process. This is based on the applications you intend to support on your network. Use these guidelines:

- In general, if the application is one-to-many or many-to-many in nature, then PIM-SM can be used successfully.
- Use SSM for optimal one-to-many application performance if IGMP version 3 is supported.

Ensure you meet these conditions before configuring PIM stub routing:

- You must have IP multicast routing configured on both the stub router and the central router. You must also have PIM mode configured on the uplink interface of the stub router.
- You must also configure either Enhanced Interior Gateway Routing Protocol (EIGRP) stub routing or Open Shortest Path First (OSPF) stub routing on the device.
- The PIM stub router does not route the transit traffic between the distribution routers. Unicast (EIGRP) stub routing enforces this behavior. You must configure unicast stub routing to assist the PIM stub router behavior.

Restrictions for PIM

These are the PIM configuration restrictions:

- Use ACLs to designate a port as a multicast host port, not a multicast router port. Multicast router control packets received on this port are dropped.
- PIM nonbroadcast multiaccess (NBMA) mode is not supported on an ethernet interface.

PIMv1 and PIMv2 interoperability

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

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

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

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

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

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

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

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

Restrictions for PIM stub routing

- Only directly connected multicast (IGMP) receivers and sources are allowed in the Layer 2 access domains. Access domains do not support the PIM protocol.
- In a network using PIM stub routing, IP traffic to the user must pass through a device configured with PIM stub routing.

- The PIM stub feature supports only nonredundant access router topology; redundant PIM stub router topology is unsupported.

Restrictions for auto-RP and BSR

Consider your network configuration and these restrictions when configuring Auto-RP and BSR:

Restrictions for auto-RP

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

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

Restrictions for BSR

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

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

Restrictions and guidelines auto-RP and BSR

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

- If your network is all Cisco routers and multilayer switches, you can use either Auto-RP or BSR. The simultaneous deployment of Auto-RP and BSR is not supported.
- If you have non-Cisco routers in your network, you must use BSR.
- If you have Cisco PIMv1 and PIMv2 routers, multilayer switches, and non-Cisco routers, both Auto-RP and BSR are required. If your network includes routers from other vendors, configure the Auto-RP mapping agent and the BSR on a Cisco PIMv2 device. Ensure no PIMv1 device is on the path between the BSR and any non-Cisco PIMv2 device.

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

- Because bootstrap messages are sent hop-by-hop, a PIMv1 device prevents messages from reaching routers and multilayer switches across your network. If your network includes a PIMv1 device and Cisco routers and multilayer switches, use Auto-RP.

- If you have a network that includes non-Cisco routers, configure the Auto-RP mapping agent and the BSR on a Cisco PIMv2 router or multilayer switch. Ensure that no PIMv1 device is on the path between the BSR and a non-Cisco PIMv2 router.
- If you have non-Cisco PIMv2 routers that need to interoperate with Cisco PIMv1 routers and multilayer switches, both Auto-RP and a BSR are required. We recommend that a Cisco PIMv2 device be both the Auto-RP mapping agent and the BSR.

Configure PIM

This section provides information about the various tasks to configure PIM.

Enable PIM stub routing

This procedure is optional.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet 1/0/1`	Specifies the interface on which you want to enable PIM stub routing, and enters interface configuration mode. • A routed port: A physical port that has been configured as a Layer 3 port by entering the **no switchport** interface configuration command. • An SVI: A VLAN interface created by using the **interface vlan** *vlan-id* global configuration command.
Step 4	**ip pim passive** **Example:**	Configures the PIM stub feature on the interface.

	Command or Action	Purpose
	`Device(config-if)# ip pim passive`	
Step 5	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 6	**show ip pim interface** **Example:** `Device# show ip pim interface`	(Optional) Displays the PIM stub that is enabled on each interface.
Step 7	**show ip igmp groups detail** **Example:** `Device# show ip igmp groups detail`	(Optional) Displays the interested clients that have joined the specific multicast source group.
Step 8	**show ip mroute** **Example:** `Device# show ip mroute`	(Optional) Displays the IP multicast routing table.
Step 9	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 10	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configure a rendezvous point

A rendezvous point (RP) is required if the interface is in sparse-dense mode and if handling the group as sparse is desired. You can use these methods:

- Manually assign an RP to multicast groups.
- Use a standalone, Cisco-proprietary protocol separate from PIMv1.

- Utilize a standards track protocol in the Internet Engineering Task Force (IETF) by configuring PIMv2 BSR.

Note You can use Auto-RP, BSR, or a combination of both, depending on the PIM version that you are running and the types of routers in your network.

Manually assign an RP to multicast groups

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

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

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

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

This procedure is optional.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip pim rp-address** *ip-address* [*access-list-number*] [**override**] **Example:** `Device(config)# ip pim rp-address 10.1.1.1 20 override`	Configures the address of a PIM RP. By default, no PIM RP address is configured. You must configure the IP address of RPs on all routers and multilayer switches (including the RP). **Note**

	Command or Action	Purpose
		If there is no RP configured for a group, the device treats the group as dense, using the dense-mode PIM techniques. A PIM device can be an RP for more than one group. Only one RP address can be used at a time within a PIM domain. The access list conditions specify the groups for which the device is an RP. • For *ip-address*, enter the unicast address of the RP in dotted-decimal notation. • (Optional) For *access-list-number*, enter an IP standard access list number from 1 to 99. If no access list is configured, the RP is used for all groups. • (Optional) The **override** keyword indicates that if there is a conflict between the RP configured with this command and one learned by Auto-RP or BSR, the RP configured with this command prevails.
Step 4	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] **Example:** `Device(config)# access-list 25 permit 10.5.0.1 255.224.0.0`	Creates a standard access list, repeating the command as many times as necessary. • For *access-list-number*, enter the access list number specified in Step 2. • The **deny** keyword denies access if the conditions are matched. • The **permit** keyword permits access if the conditions are matched. • For *source*, enter the multicast group address for which the RP should be used. • (Optional) For *source-wildcard*, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore. The access list is always terminated by an implicit deny statement for everything.
Step 5	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

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

Set up Auto-RP in a new internetwork

Note If you want to configure a PIM router as the RP for the local group, omit step 3.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**show running-config** **Example:** `Device# show running-config`	Verifies that a default RP is already configured on all PIM devices and the RP in the sparse-mode network. It was previously configured with the **ip pim rp-address** global configuration command. **Note** This step is not required for spare-dense-mode environments. The selected RP should have good connectivity and be available across the network. Use this RP for the global groups (for example, 224.x.x.x and other global groups). Do not reconfigure the group address range that this RP serves. RPs dynamically discovered through Auto-RP take precedence over statically configured RPs. Assume that it is

	Command or Action	Purpose
		desirable to use a second RP for the local groups.
Step 3	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 4	**ip pim send-rp-announce** *interface-id* **scope** *ttl* **group-list** *access-list-number* **interval** *seconds* **Example:** `Device(config)# ip pim send-rp-announce gigabitethernet 1/0/5 scope 20 group-list 10 interval 120`	Configures another PIM device to be the candidate RP for local groups. • For *interface-id*, enter the interface type and number that identifies the RP address. Valid interfaces include physical ports, port channels, and VLANs. • For **scope** *ttl*, specify the time-to-live value in hops. Enter a hop count that is high enough so that the RP-announce messages reach all mapping agents in the network. There is no default setting. The range is 1 to 255. • For **group-list** *access-list-number*, enter an IP standard access list number from 1 to 99. If no access list is configured, the RP is used for all groups. • For **interval** *seconds*, specify how often the announcement messages must be sent. The default is 60 seconds. The range is 1 to 16383.
Step 5	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] **Example:** `Device(config)# access-list 10 permit 10.10.0.0`	Creates a standard access list, repeating the command as many times as necessary. • For *access-list-number*, enter the access list number specified in Step 3. • The **deny** keyword denies access if the conditions are matched. • The **permit** keyword permits access if the conditions are matched. • For *source*, enter the multicast group address range for which the RP should be used. • (Optional) For *source-wildcard*, enter the wildcard bits in dotted decimal notation

	Command or Action	Purpose
		to be applied to the source. Place ones in the bit positions that you want to ignore. **Note** Recall that the access list is always terminated by an implicit deny statement for everything.
Step 6	**ip pim send-rp-discovery scope** *ttl* **Example:** `Device(config)# ip pim send-rp-discovery scope 50`	Finds a device with stable connectivity and assigns it the role of RP-mapping agent. For **scope** *ttl*, specify the time-to-live value in hops to limit the RP discovery packets. All devices within the hop count from the source device receive the Auto-RP discovery messages. These messages tell other devices which group-to-RP mapping to use to avoid conflicts (such as overlapping group-to-RP ranges). There is no default setting. The range is 1 to 255.
Step 7	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 8	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 9	**show ip pim rp mapping** **Example:** `Device# show ip pim rp mapping`	Displays active RPs that are cached with associated multicast routing entries.
Step 10	**show ip pim rp** **Example:** `Device# show ip pim rp`	Displays the information cached in the routing table.
Step 11	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Add Auto-RP to an existing sparse-mode cloud

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

This procedure is optional.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**show running-config** **Example:** `Device# show running-config`	Verifies that a default RP is already configured on all PIM devices and the RP in the sparse-mode network. It was previously configured with the **ip pim rp-address** global configuration command. **Note** This step is not required for spare-dense-mode environments. The selected RP should have good connectivity and be available across the network. Use this RP for the global groups (for example, 224.x.x.x and other global groups). Do not reconfigure the group address range that this RP serves. RPs dynamically discovered through Auto-RP take precedence over statically configured RPs. Assume that it is desirable to use a second RP for the local groups.
Step 3	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 4	**ip pim send-rp-announce** *interface-id* **scope** *ttl* **group-list** *access-list-number* **interval** *seconds* **Example:** `Device(config)# ip pim send-rp-announce gigabitethernet 1/0/5 scope 20 group-list 10 interval`	Configures another PIM device to be the candidate RP for local groups. • For *interface-id*, enter the interface type and number that identifies the RP address. Valid interfaces include physical ports, port channels, and VLANs. • For **scope** *ttl*, specify the time-to-live value in hops. Enter a hop count that is

	Command or Action	Purpose
	`120`	high enough so that the RP-announce messages reach all mapping agents in the network. There is no default setting. The range is 1 to 255. • For **group-list** *access-list-number*, enter an IP standard access list number from 1 to 99. If no access list is configured, the RP is used for all groups. • For **interval** *seconds*, specify how often the announcement messages must be sent. The default is 60 seconds. The range is 1 to 16383.
Step 5	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] **Example:** `Device(config)# access-list 10 permit 224.0.0.0 15.255.255.255`	Creates a standard access list, repeating the command as many times as necessary. • For *access-list-number*, enter the access list number specified in Step 3. • The **deny** keyword denies access if the conditions are matched. • The **permit** keyword permits access if the conditions are matched. • For *source*, enter the multicast group address range for which the RP should be used. • (Optional) For *source-wildcard*, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore. Recall that the access list is always terminated by an implicit deny statement for everything.
Step 6	**ip pim send-rp-discovery scope** *ttl* **Example:** `Device(config)# ip pim send-rp-discovery scope 50`	Finds a device whose connectivity is not likely to be interrupted, and assigns it the role of RP-mapping agent. For **scope** *ttl*, specify the time-to-live value in hops to limit the RP discovery packets. All devices within the hop count from the source device receive the Auto-RP discovery messages. These messages tell other devices which group-to-RP mapping to use to avoid conflicts (such as overlapping group-to-RP ranges). There is no default setting. The range is 1 to 255.

	Command or Action	Purpose
		Note To remove the device as the RP-mapping agent, use the **no ip pim send-rp-discovery** global configuration command.
Step 7	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 8	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 9	**show ip pim rp mapping** **Example:** `Device# show ip pim rp mapping`	Displays active RPs that are cached with associated multicast routing entries.
Step 10	**show ip pim rp** **Example:** `Device# show ip pim rp`	Displays the information cached in the routing table.
Step 11	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Prevent join messages to false RPs

Use the **show running-config** privileged EXEC command to determine whether the **ip pim accept-rp** command was configured across the network previously. If the **ip pim accept-rp** command is not configured on any device, this problem can be addressed later. In routers or multilayer switches where the **ip pim accept-rp** command is already configured, enter the command again to accept the newly advertised RP.

Filter incoming RP announcement messages

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

This procedure is optional.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip pim rp-announce-filter rp-list** *access-list-number* **group-list** *access-list-number* **Example:** `Device(config)# ip pim rp-announce-filter rp-list 10 group-list 14`	Filters incoming RP announcement messages. Enter this command on each mapping agent in the network. Without this command, all incoming RP-announce messages are accepted by default. For **rp-list** *access-list-number*, configure an access list of candidate RP addresses that, if permitted, is accepted for the group ranges supplied in the **group-list** *access-list-number* variable. If this variable is omitted, the filter applies to all multicast groups. If more than one mapping agent is used, the filters must be consistent across all mapping agents to ensure that no conflicts occur in the group-to-RP mapping information.
Step 4	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] **Example:** `Device(config)# access-list 10 permit 10.8.1.0 255.255.224.0`	Creates a standard access list, repeating the command as many times as necessary. • For *access-list-number*, enter the access list number specified in Step 2. • The **deny** keyword denies access if the conditions are matched. • The **permit** keyword permits access if the conditions are matched. • Create an access list that specifies from which routers and multilayer switches the mapping agent accepts candidate RP announcements (rp-list ACL).

	Command or Action	Purpose
		• Create an access list that specifies the range of multicast groups from which to accept or deny (group-list ACL). • For *source*, enter the multicast group address range for which the RP should be used. • (Optional) For *source-wildcard*, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore. The access list is always terminated by an implicit deny statement for everything.
Step 5	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 6	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configure PIMv2 BSR

The process for configuring PIMv2 BSR may involve these optional tasks:

- Defining the PIM domain border
- Defining the IP multicast boundary
- Configuring candidate BSRs
- Configuring candidate RPs

Define the PIM domain border

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet 1/0/1`	Specifies the interface to be configured, and enters interface configuration mode.
Step 4	**ip pim bsr-border** **Example:** `Device(config-if)# ip pim bsr-border`	Defines a PIM bootstrap message boundary for the PIM domain. Enter this command on each interface that connects to other bordering PIM domains. This command instructs the device to neither send nor receive PIMv2 BSR messages on this interface. **Note** To remove the PIM border, use the **no ip pim bsr-border** interface configuration command.
Step 5	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 6	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:**	(Optional) Saves your entries in the configuration file.

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

Define the IP multicast boundary

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

This procedure is optional.

Procedure

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

Step 3	**access-list** *access-list-number* **deny** *source* [*source-wildcard*]
	Example:
	`Device(config)#`
	`access-list 12 deny 224.0.1.39`
	`access-list 12 deny 224.0.1.40`
		Creates a standard access list, repeating the command as many times as necessary.
		• For *access-list-number*, the range is 1 to 99.
		• The **deny** keyword denies access if the conditions are matched.
		• For *source*, enter multicast addresses 224.0.1.39 and 224.0.1.40, which carry Auto-RP information.
		• (Optional) For *source-wildcard*, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore.
		The access list is always terminated by an implicit deny statement for everything.

Step 4	**interface** *interface-id* **Example:**	Specifies the interface to be configured, and enters interface configuration mode.

	Command or Action	Purpose
	`Device(config)# interface gigabitethernet 1/0/1`	
Step 5	**ip multicast boundary** *access-list-number* **Example:** `Device(config-if)# ip multicast boundary 12`	Configures the boundary, specifying the access list you created in Step 2.
Step 6	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 7	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 8	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configure candidate BSRs

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

This procedure is optional.

Procedure

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

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**ip pim bsr-candidate** *interface-id hash-mask-length* [*priority*] **Example:** `Device(config)# ip pim bsr-candidate gigabitethernet 1/0/3 28 100`	Configures your device to be a candidate BSR. • For *interface-id*, enter the interface on this device from which the BSR address is derived to make it a candidate. This interface must be enabled with PIM. Valid interfaces include physical ports, port channels, and VLANs. • For *hash-mask-length*, specify the mask length (32 bits maximum) that is to be ANDed with the group address before the hash function is called. All groups with the same seed hash correspond to the same RP. For example, if this value is 24, only the first 24 bits of the group addresses matter. • (Optional) For *priority*, enter a number from 0 to 255. The BSR with the larger priority is preferred. If the priority values are the same, the device with the highest IP address is selected as the BSR. The default is 0.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 5	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configure candidate RPs

Configure one or more candidate RPs. Similar to BSRs, the RPs should also have good connectivity to other devices and be in the backbone portion of the network. An RP is capable of serving the complete IP multicast address space or just a segment of it. Candidate RPs send candidate RP advertisements to the BSR.

This procedure is optional.

Before you begin

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

- In a network of Cisco routers and multilayer switches where only Auto-RP is used, any device can be configured as an RP.
- In a network that includes only Cisco PIMv2 routers and multilayer switches and with routers from other vendors, any device can be used as an RP.
- In a network including Cisco PIMv1 routers, Cisco PIMv2 routers, and routers from other vendors, configure RPs using only Cisco PIMv2 routers and multilayer switches.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip pim rp-candidate** *interface-id* [**group-list** *access-list-number*] **Example:** `Device(config)# ip pim rp-candidate gigabitethernet 1/0/5 group-list 10`	Configures your device to be a candidate RP. • For *interface-id*, specify the interface whose associated IP address is advertised as a candidate RP address. Valid interfaces include physical ports, port channels, and VLANs. • (Optional) For **group-list** *access-list-number*, enter an IP standard access list number from 1 to 99. If no group-list is specified, the device is a candidate RP for all groups.
Step 4	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*]	Creates a standard access list, repeating the command as many times as necessary.

	Command or Action	Purpose
	Example: `Device(config)# access-list 10 permit 239.0.0.0 0.255.255.255`	• For *access-list-number*, enter the access list number specified in Step 2. • The **deny** keyword denies access under matched conditions. The **permit** keyword grants access under matched conditions. • For *source*, enter the number of the network or host from which the packet is being sent. • (Optional) For *source-wildcard*, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore. The access list is always terminated by an implicit deny statement for everything.
Step 5	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 6	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configure sparse mode with Auto-RP

Before you begin

All access lists that are needed when Auto-RP is configured should be configured prior to beginning the configuration task.

Note

- If a group has no known RP when the interface is configured to sparse-dense mode, it is treated as dense mode, causing data to flood 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) or specify sparse mode (Step 7).
- When you configure sparse-dense mode, dense mode failover may result in a network dense-mode flood. To avoid this condition, use PIM sparse mode with the Auto-RP listener feature.

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

Procedure

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

	Command or Action	Purpose
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 8	Repeat Steps 1 through 9 on all PIM interfaces.	--
Step 9	**ip pim send-rp-announce** {*interface-type interface-number* | *ip-address*} **scope** *ttl-value* [**group-list** *access-list*] [**interval** *seconds*] **Example:** `Device(config)# ip pim send-rp-announce loopback0 scope 31 group-list 5`	Sends RP announcements out all PIM-enabled interfaces. • Perform this step on the RP device only. • Use the*interface-type* and *interface-number* arguments to define which IP address is to be used as the RP address. • Use the*ip-address* argument to specify a directly connected IP address as the RP address. **Note** If the*ip-address* argument is configured for this command, the RP-announce message will be sourced by the interface to which this IP address is connected (that is, the source address in the IP header of the RP-announce message is the IP address of that interface). • This example shows that the interface is enabled with a maximum of 31 hops. The IP address by which the device wants to be identified as RP is the IP address associated with loopback interface 0. Access list 5 describes the groups for which this device serves as RP.
Step 10	**ip pim send-rp-discovery** [*interface-type interface-number*] **scope** *ttl-value* [**interval** *seconds*] **Example:** `Device(config)# ip pim send-rp-discovery loopback 1 scope 31`	Configures the device to be an RP mapping agent. • Perform this step on RP mapping agent devices or on combined RP/RP mapping agent devices. **Note** Auto-RP allows the RP function to run separately on one device. Alternatively, it can deploy both the RP and RP mapping agent on a combined RP/RP mapping agent device.

	Command or Action	Purpose
		• Use the optional *interface-type* and *interface-number* arguments to define which IP address is to be used as the source address of the RP mapping agent. • Use the **scope** keyword and *ttl-value* argument to specify the Time-to-Live (TTL) value in the IP header of Auto-RP discovery messages. • Use the optional **interval** keyword and *seconds* argument to specify the interval at which Auto-RP discovery messages are sent. **Note** Lowering the interval at which Auto-RP discovery messages are sent from the default value of 60 seconds results in more frequent floodings of the group-to-RP mappings. In some network environments, the disadvantages of lowering the interval (more control packet overhead) may outweigh the advantages (more frequent group-to-RP mapping updates). • The example shows limiting the Auto-RP discovery messages to 31 hops on loopback interface 1.
Step 11	**ip pim rp-announce-filter rp-list** *access-list* **group-list** *access-list* **Example:** `Device(config)# ip pim rp-announce-filter rp-list 1 group-list 2`	Filters incoming RP announcement messages sent from candidate RPs (C-RPs) to the RP mapping agent. • Perform this step on the RP mapping agent only.
Step 12	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 1/0/0`	Selects an interface that is connected to hosts on which PIM can be enabled.
Step 13	**ip multicast boundary** *access-list* [**filter-autorp**] **Example:** `Device(config-if)# ip multicast boundary 10 filter-autorp`	Configures an administratively scoped boundary. • Perform this step on the interfaces that are boundaries to other devices. • The access list is not shown in this task.

	Command or Action	Purpose
		• An access list entry that uses the **deny** keyword creates a multicast boundary for packets that match that entry.
Step 14	**end** **Example:** `Device(config-if)# end`	Returns to global configuration mode.
Step 15	**show ip pim autorp** **Example:** `Device# show ip pim autorp`	(Optional) Displays the Auto-RP information.
Step 16	**show ip pim rp** [**mapping**] [*rp-address*] **Example:** `Device# show ip pim rp mapping`	(Optional) Displays RPs known in the network and shows how the device learned about each RP.
Step 17	**show ip igmp groups** [*group-name* | *group-address*| *interface-type interface-number*] [**detail**] **Example:** `Device# show ip igmp groups`	(Optional) Displays the multicast groups having receivers that are directly connected to the device and that were learned through Internet Group Management Protocol (IGMP). • A receiver must be active on the network at the time that this command is issued in order for receiver information to be present on the resulting display.
Step 18	**show ip mroute** [*group-address* | *group-name*] [*source-address* | *source-name*] [*interface-type interface-number*] [**summary**] [**count**] [**active** *kbps*] **Example:** `Device# show ip mroute cbone-audio`	(Optional) Displays the contents of the IP multicast routing (mroute) table.

Delay PIM shortest-path tree

Configure a traffic rate threshold for switching multicast routing from the source tree to the shortest-path tree.

This procedure is optional.

Procedure

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

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] **Example:** `Device(config)# access-list 16 permit 225.0.0.0 0.255.255.255`	Creates a standard access list. • For *access-list-number*, the range is 1 to 99. • The **deny** keyword denies access if the conditions are matched. • The **permit** keyword permits access if the conditions are matched. • For *source*, specify the multicast group to which the threshold will apply. • (Optional) For *source-wildcard*, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore. The access list is always terminated by an implicit deny statement for everything.
Step 4	**ip pim spt-threshold** {*kbps* | **infinity**} [**group-list** *access-list-number*] **Example:** `Device(config)# ip pim spt-threshold infinity group-list 16`	Specifies the threshold that must be reached before moving to shortest-path tree (spt). • For *kbps*, specify the traffic rate in kilobits per second. The default is 0 kbps. **Note** Because of device hardware limitations, 0 kbps is the only valid entry even though the range is 0 to 4294967. • Specify **infinity** if you want all sources for the specified group to use the shared tree, never switching to the source tree. • (Optional) For **group-list** *access-list-number*, specify the access list created in Step 2. When the value is 0 or the group list is unused, the threshold applies to all groups.

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

Modify the PIM router-query message interval

PIM routers and multilayer switches send PIM router-query messages to determine the designated router (DR) for each LAN segment (subnet). The DR sends IGMP host-query messages to all hosts on the directly connected LAN.

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

This procedure is optional.

Procedure

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

	Command or Action	Purpose
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet 1/0/1`	Specifies the interface to be configured, and enters interface configuration mode.
Step 4	**ip pim query-interval** *seconds* **Example:** `Device(config-if)# ip pim query-interval 45`	Configures the frequency at which the device sends PIM router-query messages. The default is 30 seconds. The range is 1 to 65535.
Step 5	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 6	**show ip igmp interface** [*interface-id*] **Example:** `Device# show ip igmp interface`	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Enable high availability on PIM using RPF

Procedure

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

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**ip multicast redundancy rpf-sync** **Example:** `Device# ip multicast redundancy rpf-sync`	Synchronizes the RPF information into PIM. RPF sync can also be enabled by enabling the **evpn-mcast** command.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Monitor and troubleshoot PIM

This section provides command information to monitor and troublehsoot PIM configuration.

Monitor PIM information

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

Table 15: PIM monitoring commands

Command	Purpose
show ip pim all-vrfs tunnel [**tunnel** *tunnel_number* | **verbose**]	Displays all VRFs.
show ip pim autorp	Displays global auto-RP information.
show ip pim boundary	Displays information about mroutes filtered by administratively scoped IPv4 multicast boundaries configured on an interface.
show ip pim interface	Displays information about interfaces configured for PIM.
show ip pim neighbor	Displays the PIM neighbor information.
show ip pim rp[*group-name* | *group-address*]	Displays RP routers associated with a sparse-mode multicast group. This command is available in all software images.
show ip pim tunnel [**tunnel** | **verbose**]	Displays information about PIM tunnel interfaces

Command	Purpose
show ip pim vrf { **word** { **all-vrfs** | **autorp** | **boundary** | **bsr-router** | **interface** | **mdt** | **neighbor** | **rp** | **rp-hash** | **tunnel** } }	Displays the VPN routing/forwarding instance.
show ip igmp groups detail	Displays the interested clients that have joined the specific multicast source group.

Monitor the RP mapping and BSR information

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

Table 16: RP mapping monitoring commands

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

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

Table 17: BSR monitoring commands

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

Troubleshoot PIMv1 and PIMv2 interoperability problems

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

1. Verify RP mapping with the **show ip pim rp-hash** privileged EXEC command, making sure that all systems agree on the same RP for the same group.
2. Verify interoperability between different versions of DRs and RPs. Make sure that the RPs are interacting with the DRs properly (by responding with register-stops and forwarding decapsulated data packets from registers).

Configuration examples for PIM

This section provides configuration examples for PIM.

Example: Enable PIM stub routing

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

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

Example: Verify PIM stub routing

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

```
Device# show ip pim interface

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

Example: Manually assign an RP to multicast groups

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

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

Example: Configure auto-RP

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

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

Example: Sparse mode with auto-RP

This example configures sparse mode with Auto-RP:

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

Example: Define IP multicast boundary to deny auto-RP information

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

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

Example: Filter incoming RP announcement messages

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

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

The mapping agent accepts candidate RP announcements from only two devices, 172.16.5.1 and 172.16.2.1 for specific multicast groups. These groups must fall in the range of 224.0.0.0 to 239.255.255.255. Otherwise, the mapping agent does not accept candidate RP announcements from any other devices. Furthermore, the mapping agent does not accept announcements from 172.16.5.1 and 172.16.2.1 for groups in the 239.0.0.0 to 239.255.255.255 range. This range is the administratively scoped address range.

Example: Prevent join messages to false RPs

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

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

Example: Configure candidate BSRs

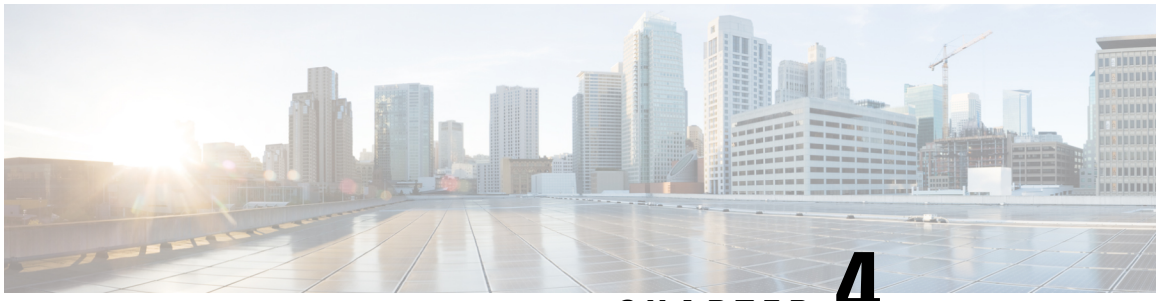
This example shows how to configure a candidate BSR, which uses the IP address 172.21.24.18 on a port as the advertised BSR address, uses 30 bits as the hash-mask-length, and has a priority of 10.

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

Example: Configure candidate RPs

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

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



CHAPTER 4

MSDP

Feature history for MSDP

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature name and description	Supported platform
Cisco IOS XE 17.18.1	MSDP: MSDP is a mechanism for connecting multiple PIM-SM domains and discovers multicast sources in other PIM domains.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

Understand MSDP

MSDP is a mechanism for connecting multiple PIM-SM domains and discovers multicast sources in other PIM domains.

This section provides information about using MSDP to interconnect multiple PIM-SM domains.

Benefits and use of MSDP

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. MSDP uses a more manageable approach to build multicast distribution trees between multiple domains.

An RP can join the interdomain source tree for sources that are sending to groups for which it has receivers. The RP is the root of the shared tree with branches to all active receivers in its domain. 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.

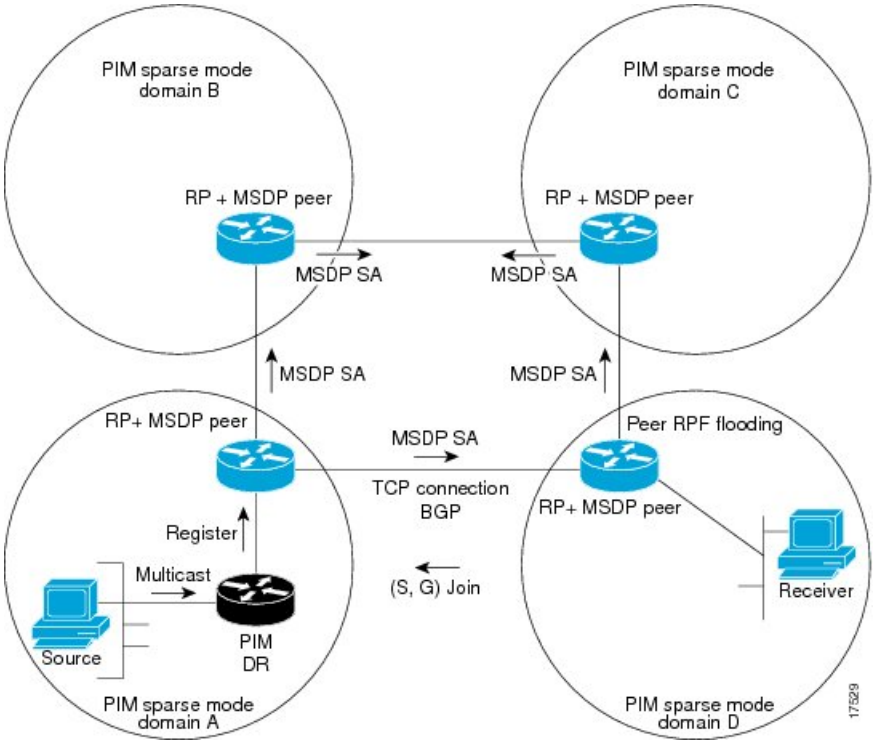
When MSDP is enabled, an RP in a PIM-SM domain maintains MSDP peering relationships with MSDP-enabled devices in other domains. This peering relationship occurs over a TCP connection, where primarily a list of sources sending to multicast groups is exchanged. MSDP uses TCP (port 639) for its peering connections. As with BGP, you must explicitly configure each peer for point-to-point TCP peering. 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 announces sources that send data to a multicast group. These announcements must originate at the RP of the domain.

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

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

Note Although this illustration and example uses routers in the configuration, any device (router or switch) can be used.

Figure 15: MSDP running between RP peers



When MSDP is implemented, this sequence of events occurs:

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

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

2. 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.
3. Each MSDP peer that receives the SA message floods the SA message to all of its peers downstream from the originator. In some cases (such as the case with the RPs in PIM-SM domains B and C in the figure), an RP may receive a copy of an SA message from more than one MSDP peer. To prevent looping, the RP consults the BGP next-hop database to determine the next hop toward the originator of the SA message. If both MBGP and unicast BGP are configured, MBGP is checked first, and then unicast BGP. That next-hop neighbor is the RPF-peer for the originator. SA messages that are received from the originator on any interface other than the interface to the RPF peer are dropped. The SA message flooding process, therefore, is referred to as peer-RPF flooding. Because of the peer-RPF flooding mechanism, BGP or MBGP must be running in conjunction with MSDP.

Note

- MBGP is not required in MSDP mesh group scenarios. For more information about MSDP mesh groups, see the Configure an MSDP mesh group, on page 139 section.
- MBGP is not required in default MSDP peer scenarios or in scenarios where only one MSDP peer is configured. For more information, see the Configure a default MSDP peer, on page 138 section.

4. 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.
5. 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, the 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 a Type, Length, and Value (TLV) data format.

SA messages

SA messages are used to advertise active sources in a domain. These SA messages may contain the initial multicast data packet 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 messaging, on page 129 section.

SA request messages

SA request messages 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. SA request messages reduce join latency by providing a list of active sources for a group, avoiding a wait time of up to 60 seconds for originating RPs to readvertise all active sources.

Note For more information about SA request messages, see the Request source information from MSDP peers, on page 141 section.

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.

Note For more information about SA response messages, see the Control the response to outgoing SA request messages from MSDP peers, on page 147 section.

Keepalive messages

Keepalive messages are sent every 60 seconds to maintain the MSDP session's activity. If no keepalive messages or SA messages are received for 75 seconds, the MSDP session is reset.

Note For more information about keepalive messages, see the Adjust the MSDP keepalive and hold-time intervals, on page 142 section.

SA messaging

This section describes SA messaging in detail.

SA message origin

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

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

When a source is in the local PIM-SM domain, it triggers the RP to create (S, G) state. 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 source's initial multicast packet, encapsulated in the register message or directly received, is included in the initial SA message.

SA message receipt

SA messages are accepted only from the MSDP RPF peer that provides the best path back to 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 MBGP routing data as the best approximation of the MSDP topology for the SA RPF check mechanism. Therefore, an MSDP topology must follow the same general structure as the BGP peer topology. With a few exceptions, such as default MSDP peers and MSDP peers in unique configurations, most MSDP peers should also be 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 MBGP peer.
- Rule 2: Applied when the sending MSDP peer is also an exterior MBGP peer.
- Rule 3: Applied when the sending MSDP peer is not an MBGP peer.

RPF checks are not performed in these 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 determines which RPF rule to apply to RPF checks using this logic. Find the MBGP neighbor that has the same IP address as the sending MSDP peer.

- If the matching MBGP neighbor is an internal BGP (iBGP) peer, apply Rule 1.
- If the matching MBGP neighbor is an external BGP (eBGP) peer, apply Rule 2.
- If no match is found, apply Rule 3.

The IP address used to configure an MSDP peer must match the one used for configuring the MBGP peer on the same device.

Rule 1 of RPF checking of SA messages in MSDP

Rule 1 of RPF checking in MSDP applies when the sending MSDP peer is also an iMBGP peer. When Rule 1 is applied, the RPF check proceeds as follows:

Summary

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 (the best path is found), the peer finds the address of the BGP neighbor for this path. It will be the address of the BGP neighbor that sent the peer the path in BGP update messages.

Note

- The BGP neighbor address is not the same as the next-hop address in the path. Since iMBGP peers do not modify the next-hop attribute, this address typically differs from the BGP peer's address that provided the path.
- The BGP neighbor address is not necessarily the same as the BGP ID of the peer that sent the peer the path.

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

Implications of Rule 1 of RPF Checking on MSDP

The MSDP topology must mirror the MBGP topology. In general, wherever there is an iMBGP 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 match the far-end iMBGP peer connection. The addresses must be the same because the BGP topology between iMBGP peers inside an autonomous system is not described by the AS path.

Instead, if iMBGP peers updated the next-hop address when sending an update, the peer could rely on it to describe the iMBGP topology (and hence the MSDP topology). However, because the default behavior for iMBGP peers is to not update the next-hop address, the peer cannot rely on the next-hop address to describe the MBGP topology (MSDP topology). Instead, the iMBGP peer uses the address of the iMBGP peer that sent the path to describe the iMBGP topology (MSDP topology) inside the autonomous system.

Tip Ensure that you use the same address for both iMBGP 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 eMBGP peer. When Rule 2 is applied, the RPF check proceeds as follows:

Summary

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 the search does not find a path, 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 eMBGP 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 MBGP topology. Configure an MSDP peer connection wherever there is an eMBGP peer connection between two devices. Unlike Rule 1, the IP address of the far-end MSDP peer connection does not have to match the far-end eMBGP peer connection.The reason that the addresses do not have to be identical is that BGP topology between two eMBGP 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 MBGP peer at all. When Rule 3 is applied, the RPF check proceeds as follows:

Summary

1. The peer searches the BGP MRIB for the best path to the RP that originated the SA message. Without a path in the MRIB, the peer 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.

3. 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 request messages

You can configure a noncaching device to send SA request messages to one or more specified MSDP peers. To reduce join latency for a noncaching RP, enable it to send SA request messages to its MSDP peer that is caching SAs. 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 to the requestor.

Note Caching of MSDP SA messages is mandatory in all current and supported software releases; it 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.

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

Default MSDP peers

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

If your switch does not support BGP and MBGP, you cannot configure an MSDP peer on the local switch by using the **ip msdp peer** global configuration command. Instead, you define a default MSDP peer (by using the ip msdp default-peer global configuration command) which can accept all SA messages for the switch. The default MSDP peer must be a previously configured MSDP peer. When your switch does not peer with an MSDP peer, configure a default MSDP peer. If only one MSDP peer is configured, your switch accepts all SA messages from it.

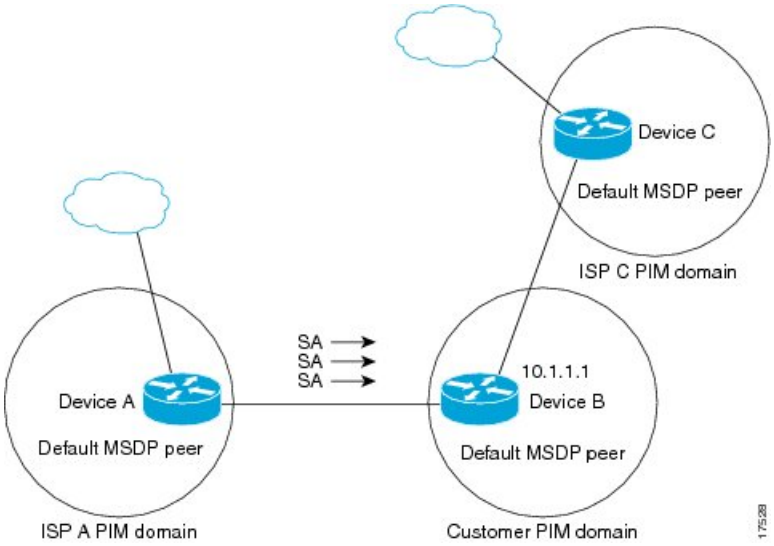
A stub autonomous system may use MSDP peerings with multiple RPs for redundancy. For example, SA messages cannot just be accepted from multiple default peers, because there is no RPF check mechanism. Instead, SA messages are accepted from only one peer. If that peer fails, SA messages are then accepted 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 likely uses a prefix list to accept prefixes from the customer device. The customer defines multiple default peers with associated prefixes. The customer has two ISPs to use. The customer defines both ISPs as default peers. As long as the first default peer identified in the configuration is up and running, it will be the default peer and the customer will accept all SA messages it receives from that peer.

Note Although the illustration uses routers in the configuration, you can use any device, such as a router or switch.

Figure 16: Default MSDP peer scenario



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

If you specify a prefix list, the peer will be a default peer only for the prefixes in the list. You can have multiple active default peers when you have a prefix list associated with each. Without prefix lists, configure multiple default peers. Only the first is active, assuming it is connected and alive. If the first configured peer or its connectivity goes down, the second configured peer becomes the active default.

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 MSDP peer in the group must establish an MSDP peering relationship (MSDP connection) with every other peer in the group. When an MSDP mesh group is configured between a group of MSDP peers, SA message flooding is reduced. When an MSDP peer in the group receives an SA message from another peer, it assumes the message has been sent to all other 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 by allowing two or more peers to efficiently share information within the group.
- SA messages are not flooded to other mesh group peers reducing the amount of SA traffic across the Internet.
- SA messages are always accepted from mesh group peers by eliminating RPF checks on arriving SA messages.

MSDP MD5 password authentication

The MSDP Message Digest 5 (MD5) password authentication feature enhances security by supporting 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 verifies each segment sent on the TCP connection between MSDP peers. The **ip msdp password peer** command enables 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. For the connection to be established, MD5 authentication must be configured with the same password on both MSDP peers. 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

- MSDP is protected against the threat of spoofed TCP segments introduced into the TCP connection stream.
- The industry-standard MD5 algorithm is used for improved reliability and security.

MSDP intervals

You can configure MSDP intervals for message and peer communication.

MSDP keepalive interval

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 sends a keepalive message and sets a 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 both when an MSDP keepalive message is sent to the peer and when the timer expires. The keepalive timer is deleted when an MSDP peering session is closed. The hold-time interval is set to a default of 75 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.

MSDP hold-time interval

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 how long the MSDP peer waits for keepalive messages before declaring peers down.

MSDP connection-retry interval

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

MSDP TTL thresholds

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. When a multicast packet is encapsulated inside a unicast SA message with a TTL of 255, its TTL does not decrease during the travel to the MSDP peer. The total number of hops traversed by the SA message can differ significantly from a normal multicast packet because multicast and unicast traffic might take paths that are entirely different to the MSDP peer and 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.

Configure MSDP

Before you configure MSDP, the addresses of all MSDP peers must be known in Border Gateway Protocol (BGP).

MSDP peer configuration

Configuring an MSDP peer is required; all other tasks are optional.

Configure 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.

Procedure

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

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 may 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.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip msdp shutdown** {*peer-name* | *peer-address*} **Example:** `Device(config)# ip msdp shutdown 192.168.1.3`	Administratively shuts down the specified MSDP peer.
Step 4	Repeat Step 3 to shut down additional MSDP peers.	--
Step 5	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configure a default MSDP peer

Perform this optional task to configure a default MSDP peer.

Before you begin

You must first configure an MSDP peer before designating it as a default MSDP peer.

Procedure

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

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

Configure an MSDP mesh group

You can configure multiple mesh groups per device.

Procedure

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

	Command or Action	Purpose
Step 4	Repeat Step 3 to add MSDP peers as members of the mesh group.	--
Step 5	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configure MSDP MD5 password authentication between MSDP peers

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip msdp 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.

	Command or Action	Purpose
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.

What to do next

If a device has a password configured for an MSDP peer but the MSDP peer does not, a message such as this 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 this 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.

Request 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.

Procedure

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

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

MSDP timer adjustments

Perform the tasks in this section to configure MSDP timers.

Adjust 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 up to 75 seconds for an MSDP peer to detect that a peering session with another MSDP peer has terminated. In network environments with redundant MSDP peers, decreasing the hold-time interval can expedite reconvergence of MSDP peers if an MSDP peer fails.

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

Procedure

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

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

Adjust the MSDP connection-retry interval

Perform this optional task to adjust the interval MSDP peers wait to reestablish peering sessions after they are reset. In environments where fast recovery of SA messages is required, such as trading floors, consider decreasing the connection-retry interval from the default 30 seconds.

Procedure

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

SA messaging

Perform the tasks in this section for SA messaging.

Control 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.

Procedure

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

Control SA messages forwarding 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.

Procedure

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

Control SA messages receipt 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.

Procedure

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

Limit the multicast data sent in SA messages using TTL thresholds

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

Procedure

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

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

Control the response to outgoing SA request messages from MSDP peers

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip msdp filter-sa-request** {*peer-address* | *peer-name*} [**list** *access-list*] **Example:** `Device(config)# ip msdp filter sa-request 172.31.2.2 list 1`	Enables a filter for outgoing SA request messages. **Note** Only one SA request filter can be configured per MSDP peer.
Step 4	Repeat Step 3 to configure SA request filters for additional MSDP peers.	--
Step 5	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configure 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 these 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 Configure an MSDP peer, on page 136 section.

Procedure

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

Prevent DoS attacks by limiting the number of SA messages

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.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip msdp sa-limit** {*peer-address* | *peer-name*} *sa-limit* **Example:** `Device(config)# ip msdp sa-limit 192.168.10.1 100`	Limits the number of SA messages allowed in the SA cache from the specified MSDP.
Step 4	Repeat Step 3 to configure SA limits for additional MSDP peers.	--
Step 5	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 6	**show ip msdp count** [*as-number*] **Example:** `Device# show ip msdp count`	(Optional) Displays the number of sources and groups originated in MSDP SA messages and the number of SA messages from an MSDP peer in the SA cache.
Step 7	**show ip msdp peer** [*peer-address* | *peer-name*] **Example:** `Device# show ip msdp peer`	(Optional) Displays detailed information about MSDP peers. **Note** The output of this command displays the number of SA messages received from MSDP peers that are stored in the cache.
Step 8	**show ip msdp summary** **Example:**	(Optional) Displays MSDP peer status. **Note**

	Command or Action	Purpose
	`Device# show ip msdp summary`	The output of this command displays a per-peer "SA Count" field that displays the number of SAs stored in the cache.

Monitor and maintain MSDP

Use the commands in these topics to monitor and maintain MSDP statistics.

Monitor MSDP

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

Procedure

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.

This is a sample output from the **debug ip msdp** command:

Example:

```
Device# debug ip msdp

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

```
MSDP: 224.150.44.250: Peer RPF check passed for 192.168.76.241, used EMBGP peer
MSDP: 224.150.44.254: Forward 56-byte SA to peer
MSDP: 224.150.44.254: Received 116-byte message from peer
MSDP: 224.150.44.254: SA TLV, len: 116, ec: 9, RP: 172.31.3.111
MSDP: 224.150.44.254: Peer RPF check passed for 172.31.3.111, used EMBGP peer
MSDP: 224.150.44.250: Forward 116-byte SA to peer
MSDP: 224.150.44.254: Received 32-byte message from peer
MSDP: 224.150.44.254: SA TLV, len: 32, ec: 2, RP: 172.31.3.78
MSDP: 224.150.44.254: Peer RPF check passed for 172.31.3.78, used EMBGP peer
MSDP: 224.150.44.250: Forward 32-byte SA to peer
```

Step 3 **debug ip msdp resets**

Use this command to debug MSDP peer reset reasons.

Example:

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

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

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

This is a 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.

This is a 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.

This is a 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.

This 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     ?
```

Clear MSDP connections statistics and SA cache entrie

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ip msdp peer** [*peer-address* | *peer-name*] **Example:** `Device# clear ip msdp peer`	Clears the TCP connection to the specified MSDP peer and resets all MSDP message counters.

	Command or Action	Purpose
Step 3	**clear ip msdp statistics** [*peer-address* | peer-name] **Example:** `Device# clear ip msdp statistics`	Clears the statistics counters for the specified MSDP peer and resets all MSDP message counters.
Step 4	**clear ip msdp sa-cache** [*group-address*] **Example:** `Device# clear ip msdp sa-cache`	Clears SA cache entries. • If the **clear ip msdp sa-cache** is specified with the optional *group-address* argument or *source-address* argument, all SA cache entries are cleared. • Use the optional *group-address* argument to clear all SA cache entries associated with a specific group.

Enable SNMP monitoring of MSDP

Perform this optional task to enable 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 MSDP Established notification is not supported in Cisco's implementation of the MSDP MIB.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**snmp-server enable traps msdp** **Example:**	Enables the sending of MSDP notifications for use with SNMP. **Note**

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

Configuration examples

This section provides configuration examples of using MSDP to interconnect multiple PIM-SM domains.

Example: Configure an MSDP peer

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

Device A

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

Device B

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

Device C

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

```
ip msdp peer 10.220.16.1 connect 10.220.16.1 connect-source Loopback0
!
```

Example: Configure a default MSDP peer

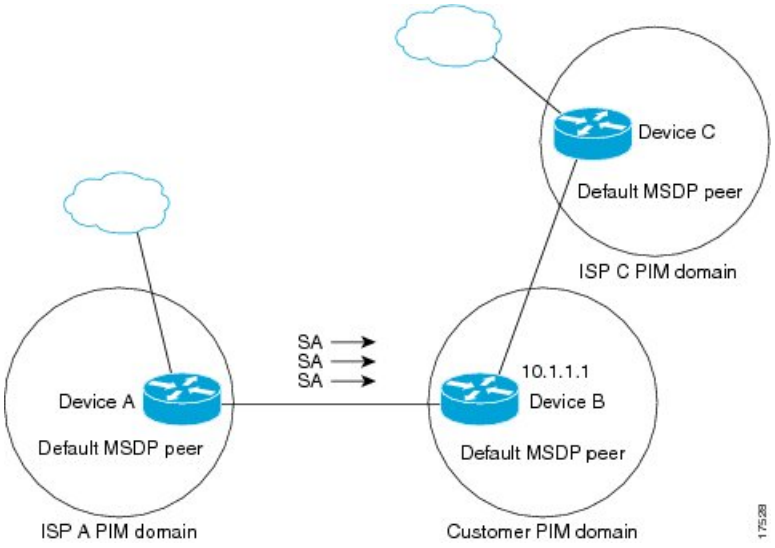
The figure illustrates a scenario where default MSDP peers might be used. In the figure, a customer that owns Device B is connected to the internet through two ISPs, one that owns Device A and the other that owns Device C. They are not running MBGP between them. In order for the customer to learn about 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.

Note Although this illustration and example uses routers in the configuration, any device (router or switch) can be used.

Figure 17: Default MSDP peer scenario



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

If you specify a prefix list, the peer will be a default peer only for the prefixes in the list. You can have multiple active default peers when you have a prefix list associated with each. When you do not have any prefix lists, you can configure multiple default peers, but only the first one is the active default peer as long as the device

has connectivity to this peer and the peer is alive. If the first configured peer goes down or the connectivity to this peer goes down, the second configured peer becomes the active default, and so on.

This 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: Configure MSDP mesh groups

This 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
```

Example: Configure MSDP MD5 password authentication

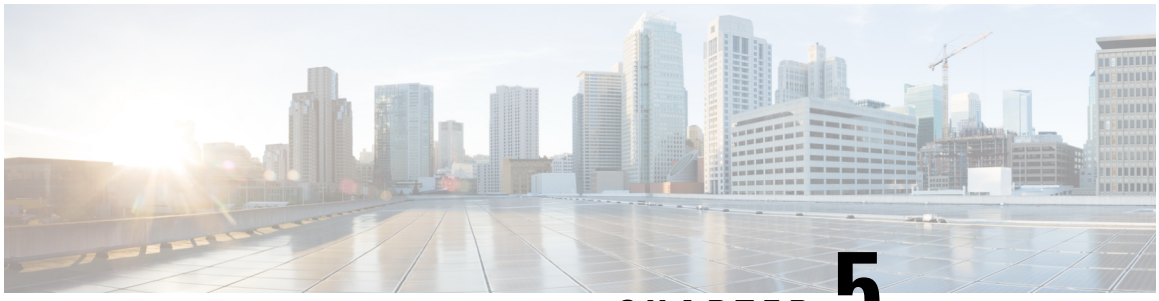
This example shows how to enable MD5 password authentication for TCP connections between two MSDP peers:

Device A

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

Device B

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



CHAPTER 5

SSM

Feature history for SSM

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature name and description	Supported platform
Cisco IOS XE 17.18.1	SSM: SSM extends IP multicast by forwarding datagram traffic to receivers only from multicast sources that receivers explicitly join.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

Understand SSM

Source-specific multicast (SSM) extends IP multicast by forwarding datagram traffic to receivers only from multicast sources that receivers explicitly join.

This section describes how to configure source-specific multicast (SSM). To get a complete description of the SSM commands in this section, check the *IP Multicast Command Reference*.

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 used in Cisco's IP multicast solutions, specifically designed for audio and video broadcast applications. The device contains components necessary for SSM implementation:

- Protocol Independent Multicast Source-Specific Mode (PIM-SSM), a routing protocol supporting SSM, derived from PIM Sparse Mode (PIM-SM)
- Internet Group Management Protocol version 3 (IGMPv3)

SSM and ISM

The current IP multicast infrastructure in the Internet and many enterprise intranets is based on the PIM-SM protocol and Multicast Source Discovery Protocol (MSDP). These protocols have the limitations of the Internet Standard Multicast (ISM) service model. For example, with ISM, the network must maintain knowledge about which hosts in the network are actively sending multicast traffic.

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

In SSM, delivery of datagrams is based on (S, G) channels. In both SSM and ISM, no signaling is required to become a source. In SSM, receivers must subscribe to specific (S, G) channels to receive traffic and unsubscribe to stop receiving traffic. 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 for channel subscription signaling uses IGMP and includes mode membership reports, which are supported only in IGMP version 3.

SSM IP address range

SSM can coexist with the ISM service by applying the SSM delivery model to a configured subset of the IP multicast group address range. You can configure SSM in Cisco IOS software for IP multicast addresses from 224.0.0.0 to 239.255.255.255. Existing IP multicast applications using an address within the SSM range will not receive traffic unless they are explicitly modified for (S, G) channel subscription

SSM operations

A network using PIM-SM for IP multicast services can support SSM. SSM can also be deployed alone in a network without the full range of protocols required for interdomain PIM-SM, such as MSDP, Auto-RP, or bootstrap router (BSR), if only SSM service is needed.

If SSM is deployed in a network already configured for PIM-SM, only the last-hop routers support SSM. You do not need to support SSM for routers not directly connected to receivers. In general, these not-last-hop routers must only run PIM-SM in the SSM range and might need additional access control configuration to suppress MSDP signalling, registering, or PIM-SM shared tree operations from occurring within the SSM range.

Use the **ip pim ssm** global configuration command to configure the SSM range and to enable SSM. This configuration has these effects:

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

SSM mapping

A typical set-top box deployment assigns each TV channel a separate IP multicast group, with one active server host sending the TV channel. A single server can send multiple TV channels, but each to a different group. In this network environment, if a router receives an IGMPv1 or IGMPv2 membership report for a particular group, the report addresses the well-known TV server for the TV channel associated with the multicast group.

When SSM mapping is configured, if a router receives an IGMPv1 or IGMPv2 membership report for a particular group, the router translates this report into one or more channel memberships for the well-known sources associated with this group.

When the router receives an IGMPv1 or IGMPv2 membership report for a group, the router uses SSM mapping to determine one or more source IP addresses for the group. SSM mapping translates the membership report into an IGMPv3 report and processes it accordingly. The router then sends PIM joins and continues to be joined to these groups as long as it continues to receive the IGMPv1 or IGMPv2 membership reports, and the SSM mapping for the group remains the same.

The last hop router uses SSM mapping to determine source addresses from a statically configured table or a DNS server. When the statically configured table or the DNS mapping changes, the router leaves the current sources associated with the joined groups.

Use the Source Specific Multicast (SSM) mapping feature when the end system cannot support SSM due to administrative or technical reasons. Use SSM mapping for video delivery to set-top boxes that lack IGMPv3 support or have applications not using the IGMPv3 host stack.

Static SSM mapping

Static SSM mapping allows you to configure the last hop router to determine which sources send to groups. Static SSM mapping requires configuring ACLs to define group ranges. After you configure the ACLs to define group ranges, map the groups permitted by those ACLs to sources using the **ip igmp ssm-map static** global configuration command.

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

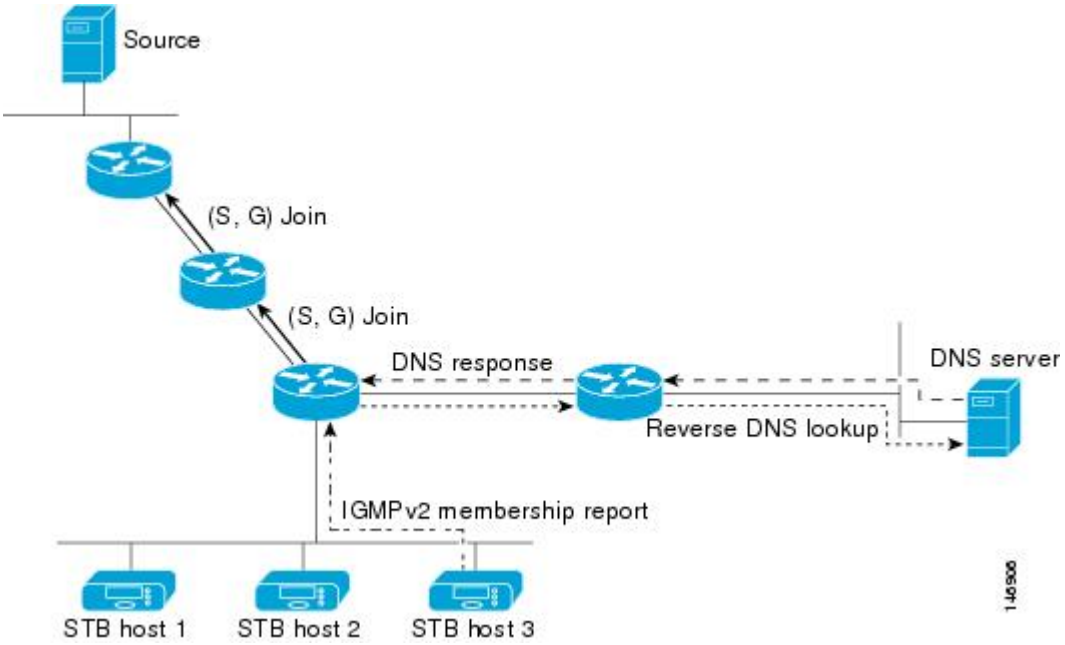
DNS-based SSM mapping

DNS-based SSM mapping allows the last hop router to perform a reverse DNS lookup to identify the sources sending to groups. When DNS-based SSM mapping is configured, the router constructs a domain name that includes the group address and performs a reverse lookup into the DNS. The router queries IP address resource

records and assigns them as 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 18: DNS-based SSM mapping

The figure displays DNS-based SSM mapping.



The SSM mapping mechanism, which enables the last hop router to join multiple sources for a group, can provide source redundancy for a TV broadcast. In this context, the last hop router provides redundancy using SSM mapping to simultaneously join two video sources for the same TV channel. To prevent the last hop router from duplicating video traffic, video sources must use a server-side switchover mechanism. One video source is active, and the other backup video source is passive. The passive source waits until an active source failure is detected before sending the video traffic for the TV channel. Thus, the server-side switchover mechanism ensures that only one of the servers is actively sending video traffic for the TV channel.

Configure these DNS records to look up source addresses for groups: G1, G2, G3, G4.

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

Refer to the DNS server documentation for details on configuring DNS resource records.

Prerequisites for SSM

Here are the prerequisites for configuring SSM and SSM mapping:

- Before configuring SSM mapping, enable IP multicast routing, PIM sparse mode, and configure SSM.
- Before configuring static SSM mapping, configure ACLs that define the group ranges to be mapped to source addresses.
- Before configuring SSM mapping with DNS lookups, add records to a running DNS server. Install a DNS server if one is not already running.

Note Use *Cisco Network Registrar* to add records to a running DNS server.

Restrictions for SSM

Here are the restrictions for configuring SSM:

- To run SSM with IGMPv3, SSM must be supported in the Cisco IOS router, the host where the application is running, and the application itself.
- Applications existing in a network before SSM must be modified to support (S, G) channel subscriptions within the SSM range. Enabling SSM might cause issues for these applications if they use addresses in the designated SSM range.
- IGMPv3 uses new membership report messages that older IGMP snooping devices might not recognize.
- When SSM is used with Layer 2 switching mechanisms, some degree of address management remains necessary. Cisco Group Management Protocol (CGMP), IGMP snooping, or Router-Port Group Management Protocol (RGMP) support only group-specific filtering, not (S, G) channel-specific filtering. If different receivers in a switched network request different (S, G) channels sharing the same group, they do not benefit from these existing mechanisms.

  Instead, both receivers receive all (S, G) channel traffic and filter out the unwanted traffic on input. SSM can reuse group addresses in the SSM range for many independent applications, potentially decreasing traffic filtering in a switched network. For this reason, it is important to use random IP addresses from the SSM range for an application to minimize the chance for re-use of a single address within the SSM range between different applications. For example, an application service providing a set of television channels should, even with SSM, use a different group for each television (S, G) channel.

  This setup guarantees that multiple receivers to different channels within the same application service never experience traffic aliasing in networks that include Layer 2 devices.
- In PIM-SSM, the last hop router will continue to periodically send (S, G) join messages if appropriate (S, G) subscriptions are on the interfaces. Therefore, as long as receivers send (S, G) subscriptions, the shortest path tree (SPT) state from the receivers to the source will be maintained, even if the source is not sending traffic for longer periods of time, or even never.

  In PIM-SM, the (S, G) state is maintained only when the source sends traffic and receivers join the group. If a source stops sending traffic for more than three minutes in PIM-SM, the (S, G) state is deleted and only reestablished after packets from the source arrive again through the RPT (rendezvous point tree). Because no mechanism in PIM-SSM notifies a receiver that a source is active, the network must maintain the (S, G) state in PIM-SSM as long as receivers are requesting receipt of that channel.

Here are the restrictions for configuring SSM mapping:

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

IGMPv3 (but not SSM), the hosts send IGMPv3 group reports. SSM mapping does not support these IGMPv3 group reports, and the router does not correctly associate sources with these reports.

Configure SSM

This section provides configuration information about SSM and SSM mapping.

Configure SSM

Follow these steps to configure SSM:

This procedure is optional.

Before you begin

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip pim ssm** [**default** | **range** *access-list*] **Example:** `Device(config)# ip pim ssm range 20`	Defines the SSM range of IP multicast addresses.
Step 4	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 1/0/1`	Selects an interface that is connected to hosts on which IGMPv3 can be enabled, and enters the interface configuration mode.
Step 5	**ip pim** {**sparse-mode**} **Example:**	Enables PIM on an interface.

	Command or Action	Purpose
	`Device(config-if)# ip pim sparse-mode`	
Step 6	**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.
Step 7	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 8	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 9	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configure static SSM mapping

Follow these steps to configure static SSM Mapping:

Procedure

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

	Command or Action	Purpose
Step 3	**ip igmp ssm-map enable** **Example:** `Device(config)# ip igmp ssm-map enable`	Enables SSM mapping for groups in the configured SSM range. **Note** By default, this command enables DNS-based SSM mapping.
Step 4	**no ip igmp ssm-map query dns** **Example:** `Device(config)# no ip igmp ssm-map query dns`	(Optional) Disables DNS-based SSM mapping. **Note** Disable DNS-based SSM mapping if you only want to rely on static SSM mapping. By default, the **ip igmp ssm-map** command enables DNS-based SSM mapping.
Step 5	**ip igmp ssm-map static** *access-list source-address* **Example:** `Device(config)# ip igmp ssm-map static 11 172.16.8.11`	Configures static SSM mapping. • The ACL supplied for the *access-list* argument defines the groups to be mapped to the source IP address entered for the *source-address* argument. **Note** You can configure additional static SSM mappings. If additional SSM mappings are configured and the router receives an IGMPv1 or IGMPv2 membership report for a group in the SSM range, the device determines the source addresses associated with the group by walking each configured **ip igmp ssm-map static** command. The device associates up to 20 sources per group. Repeat this step to configure additional static SSM mappings, if required.
Step 6	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 7	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 8	**copy running-config startup-config** **Example:**	(Optional) Saves your entries in the configuration file.

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

Configure DNS-based SSM mapping

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip igmp ssm-map enable** **Example:** `Device(config)# ip igmp ssm-map enable`	Enables SSM mapping for groups in a configured SSM range.
Step 4	**ip igmp ssm-map query dns** **Example:** `Device(config)# ip igmp ssm-map query dns`	(Optional) Enables DNS-based SSM mapping. • By default, the **ip igmp ssm-map** command enables DNS-based SSM mapping. Only the **no** form of this command is saved to the running configuration. **Note** Use this command to reenable DNS-based SSM mapping if DNS-based SSM mapping is disabled.
Step 5	**ip domain multicast** *domain-prefix* **Example:**	(Optional) Changes the domain prefix used for DNS-based SSM mapping.

	Command or Action	Purpose
	`Device(config)# ip domain multicast ssm-map.cisco.com`	• By default, the software uses the ip-addr.arpa domain prefix.
Step 6	**ip name-server** *server-address1* [*server-address2...server-address6*] **Example:** `Device(config)# ip name-server 10.48.81.21`	Specifies the address of one or more name servers to use for name and address resolution. Repeat this step to configure additional DNS servers for redundancy, if required. .
Step 7	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 8	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 9	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configure static traffic forwarding with SSM mapping

Follow these steps to configure static traffic forwarding with SSM mapping on the last hop router:

Procedure

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

	Command or Action	Purpose
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet 1/0/1`	Selects an interface on which to statically forward traffic for a multicast group using SSM mapping, and enters interface configuration mode. **Note** Static forwarding of traffic with SSM mapping works with either DNS-based SSM mapping or statically configured SSM mapping.
Step 4	**ip igmp static-group** *group-address* **source ssm-map** **Example:** `Device(config-if)# ip igmp static-group 239.1.2.1 source ssm-map`	Configures SSM mapping to statically forward a (S, G) channel from the interface. Use this command if you want to statically forward SSM traffic for certain groups. Use DNS-based SSM mapping to determine the source addresses of the channels.
Step 5	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 6	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configure IPv6 SSM mapping

When the SSM mapping feature is enabled, DNS-based SSM mapping is automatically enabled, which means that the switch will look up the source of a multicast MLD version 1 report from a DNS server.

You can use either DNS-based or static SSM mapping, depending on your switch 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.

Note To use DNS-based SSM mapping, the switch needs to find at least one correctly configured DNS server, to which the switch may be directly attached.

To configuring SSM mapping, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 mld ssm-map enable** **Example:** `Device(config)# ipv6 mld ssm-map enable`	Enables the SSM mapping feature for groups in the configured SSM range.
Step 4	**no ipv6 mld ssm-map query dns** **Example:** `Device(config)# no ipv6 mld ssm-map query dns`	Disables DNS-based SSM mapping.
Step 5	**ipv6 mld ssm-map static** *access-list source-address* **Example:** `Device(config-if)# ipv6 mld ssm-map static SSM_MAP_ACL_2 2001:DB8:1::1`	Configures static SSM mappings.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Exits global configuration mode, and returns the switch to privileged EXEC mode.
Step 7	**show ipv6 mld ssm-map** [*source-address*] **Example:**	Displays SSM mapping information.

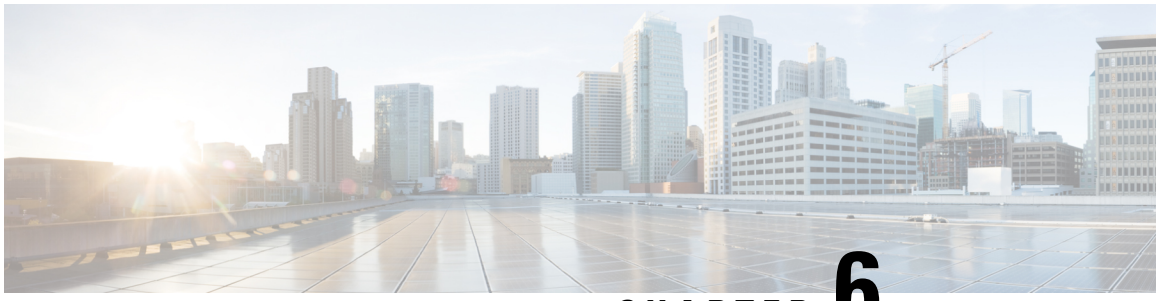
	Command or Action	Purpose
	`Device(config-if)# show ipv6 mld ssm-map`	
Step 8	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Monitor SSM

Use the privileged EXEC commands in this table to monitor SSM.

Table 18: Commands for monitoring SSM

Command	Purpose
show ip igmp groups detail	Displays the (S, G) channel subscription through IGMPv3.
show ip mroute	Displays whether a multicast group supports SSM service or whether a source-specific host report was received.
show ip igmp ssm-mapping	Displays information about SSM mapping.
show ip igmp ssm-mapping *group-address*	Displays the sources that SSM mapping uses for a particular group.
show ip igmp groups [*group-name* | *group-address* | *interface-type interface-number*] [**detail**]	Displays the multicast groups with receivers that are directly connected to the router and that were learned through IGMP.
show host	Displays the default domain name, the style of name lookup service, a list of name server hosts, and the cached list of hostnames and addresses.
debug ip igmp *group-address*	Displays the IGMP packets received and sent and IGMP host-related events.



CHAPTER 6

IPv6 Multicast Routing

Feature history for IPv6 multicast

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature name and description	Supported platform
Cisco IOS XE 17.18.1	IPv6 multicast: IPv6 multicast enables transmitting a single data stream to multiple selected hosts simultaneously.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

Understand IPv6 multicast

Traditional IP communication allows a host to send packets to an individual host (unicast) or all hosts (broadcast). IPv6 multicast enables transmitting a single data stream to multiple selected hosts simultaneously (group transmission).

An IPv6 multicast group consists of receivers wanting 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 interested in receiving data flowing to a particular group must signal their local switch to join the group. This signaling is achieved with the MLD protocol.

Switches use the MLD protocol to learn whether members of a group are present on their directly attached subnets. Hosts join multicast groups by sending MLD report messages. The network then delivers data to a potentially unlimited number of receivers, using only one copy of the multicast data on each subnet. IPv6 hosts that wish to receive the traffic are known as group members.

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

The multicast environment consists of senders and receivers. Any host can send messages to a group, whether or not it is a member. However, only members of a group can listen to and receive the message.

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

Note As per RFC 4291, the FF0x::/12 (where the T flag is set to 0 in IPv6 destination address) is for permanently assigned ("well-known") IPv6 multicast address range. Packets with this address range typically flood in the ingress VLAN.

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.

The activity, duration, and membership of a multicast group can vary from group to group and time to time. A group that has members may have no activity.

IPv6 multicast routing implementation

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

- MLD is used by IPv6 switches to discover multicast listeners (nodes that want to receive multicast packets destined for specific multicast addresses) on directly attached links. MLD has two versions.

  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 IGMP for IPv4. Cisco IOS software uses both MLD version 2 and MLD version 1 for IPv6 multicast. 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 switch running MLD version 2. LANs with both MLD version 1 and version 2 hosts are supported.

- PIM-SM operates between switches to track multicast packets for forwarding to directly connected LANs.
- PIM in Source Specific Multicast (PIM-SSM) is similar to PIM-SM. It adds the ability to report interest in packets from specific source addresses or exclude specific source addresses to an IP multicast address.

IPv6 multicast listener discovery protocol

To implement multicasting in the campus network, you need to define the multicast recipients. The MLD protocol is used by IPv6 switches 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. MLD protocol can be used to discover local group and source-specific group membership.

The MLD protocol automatically controls multicast traffic flow and limits it using special multicast queriers and hosts.

Multicast queriers and hosts

A multicast querier is a network device that sends query messages to discover which devices are members of a multicast group, such as a switch.

A multicast host is a receiver, including switches, that sends report messages to inform the querier about its host membership.

A multicast group consists of queriers and hosts that receive multicast data streams from the same source. Queriers and hosts use MLD reports to join, leave multicast groups, and 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 have the switch alert option set. The switch alert option indicates that the hop-by-hop option header is implemented.

MLD access group

The MLD access group provides receiver access control in Cisco IOS IPv6 multicast switches. This feature limits the list of groups a receiver can join and determines which sources are allowed or denied for joining SSM channels.

Explicit tracking

The explicit tracking feature enables a switch to monitor host behavior within its IPv6 network. This feature also enables the fast leave mechanism to be used with MLD version 2 host reports.

Protocol independent multicast

Protocol Independent Multicast (PIM) is used between switches so that they can track which multicast packets to forward to each other and to their directly connected LANs. PIM operates independently of the unicast routing protocol, transmitting and receiving multicast route updates.

Regardless of the unicast routing protocols used in the LAN to populate the unicast routing table, Cisco IOS PIM leverages the existing unicast table for the Reverse Path Forwarding (RPF) check, avoiding the need for a separate routing table.

Configure IPv6 multicast to use PIM-SM, PIM-SSM, or 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. However, it is not dependent on any specific unicast routing protocol.

PIM-SM is used in a multicast network when relatively few switches are involved in each multicast and these switches 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. Initially, PIM-SM requires a RP and uses shared trees.

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 switch 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 switch.

As a PIM join travels up the tree, switches 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 switch 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 switch 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 switch (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 switches 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.

IPv6 BSR RP mapping

PIM switches in a domain map multicast groups to the correct RP address. The BSR protocol for PIM-SM dynamically distributes group-to-RP mapping information across a domain. The IPv6 BSR feature detects when an RP is unreachable and updates the mapping tables, ensuring its removal. These updated tables are distributed quickly across the domain.

Every PIM-SM multicast group must associate with the IP or IPv6 address of an RP. A new multicast sender's local DR encapsulates data packets in a PIM register message and sends them to the RP for that multicast group. A new multicast receiver's local DR sends a PIM join message to the RP for that multicast group. Identify the next switch toward the RP when sending a (*, G) join message to facilitate message delivery.

When forwarding data packets using (*, G) state, a PIM switch must identify the correct incoming interface for packets destined for G. Packets arriving on other interfaces are rejected.

A limited number of switches within a domain are configured as candidate bootstrap switches (C-BSRs). One BSR is selected for the domain. A set of switches within a domain are also configured as candidate RPs (C-RPs); typically, these switches are the same switches that are configured as C-BSRs. Candidate RPs periodically unicast candidate-RP-advertisement (C-RP-Adv) messages to the BSR of that domain, advertising their willingness to be an RP.

A C-RP-Adv message includes the address of the advertising C-RP. It also lists optional group addresses and mask length fields, indicating the group prefixes for which the candidacy is advertised. The BSR then includes a set of these C-RPs, along with their corresponding group prefixes, in bootstrap messages (BSMs) it periodically originates. BSMs are distributed hop-by-hop throughout the domain.

Bidirectional BSR support provides mechanisms for advertising bidirectional RPs in C-RP messages and specifying bidirectional ranges in the BSM. All switches in a system must be able to use the bidirectional range in the BSM; otherwise, the bidirectional RP feature will not function.

PIM-source specific multicast

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

Further, instead of the use of RP and shared trees, SSM uses information found on source addresses for a multicast group. This information is provided by receivers through the source addresses relayed to the last-hop switches by MLD membership reports, resulting in shortest-path trees directly to the sources.

In SSM, delivery of datagrams is based on (S, G) channels. Traffic for one (S, G) channel consists of datagrams with an IPv6 unicast source address S and the multicast group address G as the IPv6 destination address. You will receive this traffic by subscribing to the (S, G) channel. Signaling is not required, but receivers must subscribe or unsubscribe to (S, G) channels to receive or not receive traffic from specific sources.

MLD version 2 is required for SSM to operate. MLD allows the host to provide source information. Ensure that SSM is supported in the Cisco IOS IPv6 switch, on the host running the application, and within the application itself before using SSM with MLD.

Routable address hello option

When an IPv6 interior gateway protocol is used to build the unicast routing table, the procedure to detect the upstream switch address assumes the address of a PIM neighbor is the same as the address of the next-hop switch, as long as they refer to the same switch. However, it may not be the case when a switch has multiple addresses on a link.

Two typical situations can lead to this issue with IPv6: one when the unicast routing table is not built by an IPv6 interior gateway protocol such as multicast BGP, and another when the address of an RP shares a subnet prefix with downstream switches (note that the RP switch address has to be domain-wide and therefore cannot be a link-local address).

The routable address hello option allows the PIM protocol to avoid these situations by including all addresses from the advertised interface in the PIM hello message. When a PIM switch finds an upstream switch for some address, the result of RPF calculation is compared with the addresses in this option, in addition to the PIM neighbor's address itself. Because this option includes all possible addresses of a PIM switch on that link, you can always include the RPF calculation result, provided it refers to the PIM switch supporting this option.

Because of size restrictions on PIM messages and the requirement that a routable address hello option fits within a single PIM hello message, a limit of 16 addresses can be configured on the interface.

PIM IPv6 stub routing

Use PIM stub routing to reduce resource usage by moving routed traffic closer to users.

For IPv6 traffic in a PIM stub routing network, the only allowable route to the user is through a switch 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.

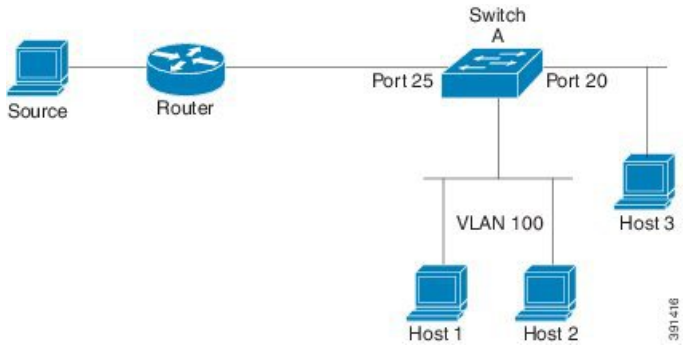
Configure the distribution and remote routers to use IPv6 multicast routing. Set the switch as a PIM stub router. Transit traffic between distribution routers cannot be routed using the switch. You also need to configure a routed uplink port on the switch. The switch uplink port cannot be used with SVIs.

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. In a non-redundant topology, the PIM passive interface assumes it is the only designated router on the access domain.

The configuration involves Switch A routed uplink port 25 connected to the router, with PIM stub routing enabled on the VLAN 100 interfaces and on Host 3. This configuration allows directly connected hosts to receive traffic from the multicast source.

Figure 19: PIM stub router configuration



IPv6 multicast process switching and fast switching

The unified MFIB supports fast switching and process switching for PIM-SM and PIM-SSM in IPv6 multicast. In process switching, the IOS daemon 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 IOSd also computes the cyclic redundancy check (CRC). This switching method is the least scalable method for switching IPv6 packets.

To enhance packet forwarding performance, use IPv6 multicast fast switching instead of process switching. In IPv6 multicast switching, information that is usually stored in a route cache is stored in several data structures. 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, you will see it point 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 when a switch is configured for load balancing and redundancy. For each resolved path, a pointer is added for the adjacency corresponding to the next-hop interface for that path. The load balancing mechanism utilizes this method across several paths.

Multiprotocol BGP for the IPv6 multicast address family

Multiprotocol BGP for IPv6 multicast has extensions similar to those of IPv4 BGP. These IPv6 enhancements to multicast BGP include support for an IPv6 multicast address family, network layer reachability information (NLRI), and next-hop attributes that use IPv6 addresses.

Multicast BGP is an enhanced BGP that allows the deployment of interdomain IPv6 multicast. Multiprotocol BGP carries routing information, such as IPv6 address family and IPv6 multicast routes. The IPv6 multicast address family contains routes used for RPF lookup by the IPv6 PIM protocol. Additionally, multicast BGP IPv6 provides interdomain transport for these routes. Use multiprotocol BGP for IPv6 multicast if you are using IPv6 multicast with BGP. Unicast BGP learned routes won't be used for IPv6 multicast.

A separate address family context provides multicast BGP functionality. A subsequent address family identifier (SAFI) provides information about the type of network layer reachability information carried in the attribute. Multiprotocol BGP unicast uses SAFI 1 messages, and multiprotocol BGP multicast uses SAFI 2 messages. SAFI 1 messages indicate that the routes are only usable for IP unicast, but not IP multicast. Because of this functionality, BGP routes in the IPv6 unicast RIB must be ignored in the IPv6 multicast RPF lookup.

A separate BGP routing table is maintained to configure incongruent policies and topologies (forexample, IPv6 unicast and multicast) by using IPv6 multicast RPF lookup. Multicast RPF lookup is very similar to the IP unicast route lookup.

No MRIB is associated with the IPv6 multicast BGP table. However, Pv6 multicast BGP processes the unicast IPv6 RIB when necessary. Multicast BGP does not insert or update routes into the IPv6 unicast RIB.

Embedded RP

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. If a device serves as the RP, configure it statically as the RP.

Search for embedded RP group addresses within MLD reports, 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, 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:

- The first-hop device encapsulates data in register packets and unicasts it directly to the RP while operating as the DR.
- The RPF forwarding algorithm, as described in the PIM-Sparse Mode section, dictates multicast forwarding if the RP has joined the source tree.

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. Match an access list or compare the AS path for the registered source with the AS path specified in a route map.

Static mroutes

IPv6 static mroutes function in a similar manner to IPv4 static mroutes, influencing the RPF check. IPv6 static mroutes share the same database as IPv6 static routes. They also extend static route support for RPF checks. Static mroutes support equal-cost multipath mroutes and unicast-only static routes.

MRIB

The Multicast Routing Information Base (MRIB) is a protocol-independent repository of multicast routing entries instantiated by multicast routing protocols (routing clients). ts primary function is to allow routing protocols to operate independently from 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 accommodates both forwarding clients (MFIB instances) and special clients like MLD. MFIB retrieves forwarding entries from MRIB and notifies MRIB about events related to packet reception. Routing clients can request notifications, or MFIB can generate them independently.

To coordinate multiple routing clients for multicast connectivity within the same session, utilize MRIB. Additionally, MRIB facilitates 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 to read the IPv6 multicast forwarding table and receive notifications when the forwarding table changes. The MFIB provides information with clearly defined forwarding semantics. This design makes it easy for the platform to translate the information to its specific hardware or software forwarding mechanisms.

When network routing or topology changes, the IPv6 routing table is updated, and these 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.

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

Configure IPv6 multicast

This section explains how to configure IPv6 multicast.

Enable IPv6 multicast routing

Perform this procedure to enable IPv6 multicast routing:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enter global configuration mode.
Step 3	**ipv6 multicast-routing** **Example:** `Device(config)# ipv6 multicast-routing`	Enables multicast routing on all IPv6-enabled interfaces and enables multicast forwarding for PIM and MLD on all enabled interfaces of the switch.
Step 4	**copy running-config startup-config** **Example:** `Device(config)# copy running-config startup-config`	(Optional) Save your entries in the configuration file.

Customize and verify the MLD protocol

This section explains how to configure and verify the MLD protocol.

Customize and verify MLD on an interface

Perform this procedure to customize and verify MLD on an interface:

Procedure

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

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 1/0/1`	Specifies an interface type and number, and places the switch in interface configuration mode.
Step 4	**ipv6 mld join-group** [*group-address*] [**include** | **exclude**] {*source-address* | **source-list** [*acl*]} **Example:** `Device(config-if)# ipv6 mld join-group FF04::10`	Configures MLD reporting for a specified group and source.
Step 5	**ipv6 mld access-group** *access-list-name* **Example:** `Device(config-if)# ipv6 access-list acc-grp-1`	Allows the user to perform IPv6 multicast receiver access control.
Step 6	**ipv6 mld static-group** [*group-address*] [**include** | **exclude**] {*source-address* | source-list [*acl*]} **Example:** `Device(config-if)# ipv6 mld static-group ff04::10 include 100::1`	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.
Step 7	**ipv6 mld query-max-response-time** *seconds* **Example:** `Device(config-if)# ipv6 mld query-timeout 130`	Configures the timeout value before the switch takes over as the querier for the interface.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Enter this command twice to exit interface configuration mode and enter privileged EXEC mode.
Step 9	**show ipv6 mld groups [link-local]** [*group-name* | *group-address*] [*interface-type interface-number*] [**detail** | **explicit**] **Example:** `Device# show ipv6 mld groups GigabitEthernet 1/0/1`	Displays the multicast groups that are directly connected to the switch and that were learned through MLD.
Step 10	**show ipv6 mld groups summary** **Example:**	Displays the number of (*, G) and (S, G) membership reports present in the MLD cache.

	Command or Action	Purpose
	`Device# show ipv6 mld groups summary`	
Step 11	**show ipv6 mld interface** [*type number*] **Example:** `Device# show ipv6 mld interface GigabitEthernet 1/0/1`	Displays multicast-related information about an interface.
Step 12	**debug ipv6 mld** [*group-name* | *group-address* | *interface-type*] **Example:** `Device# debug ipv6 mld`	Enables debugging on MLD protocol activity.
Step 13	**debug ipv6 mld explicit** [*group-name* | *group-address*] **Example:** `Device# debug ipv6 mld explicit`	Displays information related to the explicit tracking of hosts.
Step 14	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Implement MLD group limits

Per-interface and global MLD limits operate independently of each other. Both per-interface and global MLD limits can be configured on the same switch. The number of MLD limits, globally or per interface, is not configured by default; the limits must be configured by the user. A membership report that exceeds either the per-interface or the global state limit is ignored.

Implement MLD group limits globally

Perform this procedure to implement MLD group limits globally:

Procedure

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

	Command or Action	Purpose
Step 3	**ipv6 mld** [*vrf vrf-name*] **state-limit** *number* **Example:** `Device(config)# ipv6 mld state-limit 300`	Limits the number of MLD states globally.
Step 4	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Implement MLD group limits per interface

Perform this procedure to implement MLD group limits per interface:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface type** *number* **Example:** `Device(config)# interface GigabitEthernet 1/0/1`	Specifies an interface type and number, and places the switch in interface configuration mode.
Step 4	**ipv6 mld limit** *number* [**except**] *access-list* **Example:** `Device(config-if)# ipv6 mld limit 100`	Limits the number of MLD states on a per-interface basis.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Configure explicit tracking of receivers to track host behavior

The explicit tracking feature allows a switch to track the behavior of the hosts within its IPv6 network and enables the fast leave mechanism to be used with MLD version 2 host reports.

Perform this procedure to configuring explicit tracking of receivers to track host behavior:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enter global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 1/0/1`	Specifies an interface type and number, and places the switch in interface configuration mode.
Step 4	**ipv6 mld explicit-tracking** *access-list-name* **Example:** `Device(config-if)# ipv6 mld explicit-tracking list1`	Enables explicit tracking of hosts.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Reset the MLD traffic counters

Perform this procedure to reset the MLD traffic counters:

Procedure

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

	Command or Action	Purpose
Step 3	**clear ipv6 mld traffic** **Example:** `Device# clear ipv6 mld traffic`	Resets all MLD traffic counters.
Step 4	**show ipv6 mld traffic** **Example:** `Device# show ipv6 mld traffic`	Displays the MLD traffic counters.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Clear the MLD interface counters

Perform this procedure to clearing the MLD interface counters:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**clear ipv6 mld counters** *interface-type* **Example:** `Device# clear ipv6 mld counters Ethernet1/0`	Clears the MLD interface counters.
Step 4	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Configuring PIM

This section explains how to configure PIM.

Configure PIM-SM and display PIM-SM information for a group range

Perform this procedure to configuring PIM-SM and view PIM-SM information for a group range:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 pim rp-address** *ipv6-address* [*group-access-list*] **Example:** `Device(config)# ipv6 pim rp-address 2001:DB8::01:800:200E:8C6C acc-grp-1`	Configures the address of a PIM RP for a particular group range.
Step 4	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode, and returns the switch to privileged EXEC mode.
Step 5	**show ipv6 pim interface** [**state-on**] [**state-off**] [*type-number*] **Example:** `Device# show ipv6 pim interface`	Displays information about interfaces configured for PIM.
Step 6	**show ipv6 pim group-map** [*group-name* | *group-address*] | [*group-range* | *group-mask*] [**info-source {bsr** | **default** | **embedded-rp** | **static**}] **Example:** `Device# show ipv6 pim group-map`	Displays an IPv6 multicast group mapping table.
Step 7	**show ipv6 pim neighbor** [**detail**] [*interface-type interface-number* | **count**] **Example:** `Device# show ipv6 pim neighbor`	Displays the PIM neighbors discovered by the Cisco IOS software.

	Command or Action	Purpose
Step 8	**show ipv6 pim range-list** [**config**] [*rp-address* | *rp-name*] **Example:** `Device# show ipv6 pim range-list`	Displays information about IPv6 multicast range lists.
Step 9	**show ipv6 pim tunnel** [*interface-type interface-number*] **Example:** `Device# show ipv6 pim tunnel`	Displays information about the PIM register encapsulation and de-encapsulation tunnels on an interface.
Step 10	**debug ipv6 pim** [*group-name* | *group-address* | **interface** *interface-type* | **bsr** | **group** | **mvpn** | **neighbor**] **Example:** `Device# debug ipv6 pim`	Enables debugging on PIM protocol activity.
Step 11	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Configure PIM options

Perform this procedure to configure PIM options:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 pim spt-threshold infinity** [**group-list** *access-list-name*] **Example:** `Device(config)# ipv6 pim spt-threshold infinity group-list acc-grp-1`	Configures when a PIM leaf switch joins the SPT for the specified groups.

	Command or Action	Purpose
Step 4	**ipv6 pim accept-register** {**list** *access-list* | **route-map** *map-name*} **Example:** `Device(config)# ipv6 pim accept-register route-map reg-filter`	Accepts or rejects registers at the RP.
Step 5	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 1/0/1`	Specifies an interface type and number, and places the switch in interface configuration mode.
Step 6	**ipv6 pim dr-priority** *value* **Example:** `Device(config-if)# ipv6 pim dr-priority 3`	Configures the DR priority on a PIM switch.
Step 7	**ipv6 pim hello-interval** *seconds* **Example:** `Device(config-if)# ipv6 pim hello-interval 45`	Configures the frequency of PIM hello messages on an interface.
Step 8	**ipv6 pim join-prune-interval** *seconds* **Example:** `Device(config-if)# ipv6 pim join-prune-interval 75`	Configures periodic join and prune announcement intervals for a specified interface.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Enter this command twice to exit interface configuration mode and enter privileged EXEC mode.
Step 10	**ipv6 pim join-prune statistic** [*interface-type*] **Example:** `Device(config-if)# show ipv6 pim join-prune statistic`	Displays the average join-prune aggregation for the most recently aggregated packets for each interface.
Step 11	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Reset PIM traffic counters

If PIM malfunctions or in order to verify that the expected number of PIM packets are received and sent, the user can clear PIM traffic counters. Once the traffic counters are cleared, the user can enter the show ipv6 pim traffic command to verify that PIM is functioning correctly and that PIM packets are being received and sent correctly.

Perform this procedure to reset the PIM traffic counters:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**clear ipv6 pim traffic** **Example:** `Device# clear ipv6 pim traffic`	Resets the PIM traffic counters.
Step 4	**show ipv6 pim traffic** **Example:** `Device# show ipv6 pim traffic`	Displays the PIM traffic counters.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Clear PIM topology table to reset 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.

Perform this procedure to clear the PIM topology table to reset the MRIB connection:

Procedure

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

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**clear ipv6 pim topology** [*group-name* | *group-address*] **Example:** `Device# clear ipv6 pim topology FF04::10`	Clears the PIM topology table.
Step 4	**show ipv6 mrib client** [**filter**] [**name** {*client-name* | *client-name* **: client-id**}] **Example:** `Device# show ipv6 mrib client`	Displays multicast-related information about an interface.
Step 5	**show ipv6 mrib route** {**link-local** | **summary** | [*sourceaddress-or-name* | *] [*groupname-or-address* [*prefix-length*]]] **Example:** `Device# show ipv6 mrib route`	Displays the MRIB route information.
Step 6	**show ipv6 pim topology** [*groupname-or-address* [*sourceaddress-or-name*] | **link-local** | **route-count** [**detail**]] **Example:** `Device# show ipv6 pim topology`	Displays PIM topology table information for a specific group or all groups.
Step 7	**debug ipv6 mrib client** **Example:** `Device# debug ipv6 mrib client`	Enables debugging on MRIB client management activity.
Step 8	**debug ipv6 mrib io** **Example:** `Device# debug ipv6 mrib io`	Enables debugging on MRIB I/O events.

	Command or Action	Purpose
Step 9	**debug ipv6 mrib proxy** **Example:** `Device# debug ipv6 mrib proxy`	Enables debugging on MRIB proxy activity between the switch processor and line cards on distributed switch platforms.
Step 10	**debug ipv6 mrib route** [*group-name* | *group-address*] **Example:** `Device# debug ipv6 mrib route`	Displays information about MRIB routing entry-related activity.
Step 11	**debug ipv6 mrib table** **Example:** `Device# debug ipv6 mrib table`	Enables debugging on MRIB table management activity.
Step 12	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Configure PIM IPv6 stub routing

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

PIM IPv6 stub routing configuration guidelines

- Before configuring PIM stub routing, you must have IPv6 multicast routing configured on both the stub router and the central router. PIM mode (sparse-mode) must be configured on the uplink interface of the stub router.
- The PIM stub router prevents the routing of transit traffic between distribution routers. Unicast (EIGRP) stub routing enforces this behavior. Configure unicast stub routing to support PIM stub router behavior. For more information, see the *EIGRP Stub Routing* section.
- Only directly connected multicast (MLD) receivers and sources are allowed in the Layer 2 access domains. The PIM protocol is not supported in access domains.
- The redundant PIM stub router topology is not supported.

Default IPv6 PIM routing configuration

This table displays the default IPv6 PIM routing configuration.

Table 19: Default multicast routing configuration

Feature	Default Setting
Multicast routing	Disabled on all interfaces.

Feature	Default Setting
PIM version	Version 2.
PIM mode	No mode is defined.
PIM stub routing	None configured.
PIM RP address	None configured.
PIM domain border	Disabled.
PIM multicast boundary	None.
Candidate BSRs	Disabled.
Candidate RPs	Disabled.
Shortest-path tree threshold rate	0 kb/s.
PIM router query message interval	30 seconds.

Enable IPV6 PIM stub routing

Perform this procedure to enable IPV6 PIM stub routing:

Before you begin

PIM stub routing is disabled in IPv6 by default.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 multicast pim-passive-enable** **Example:** `Device(config-if)# ipv6 multicast pim-passive-enable`	Enables IPv6 Multicast PIM routing on the switch.

	Command or Action	Purpose
Step 4	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet 9/0/6`	Specifies the interface on which you want to enable PIM stub routing, and enters interface configuration mode. The specified interface must be one of the following: • A routed port: A physical port that has been configured as a Layer 3 port by entering the **no switchport** interface configuration command. You will also need to enable IP PIM sparse mode on the interface, and join the interface as a statically connected member to an MLD static group. • An SVI: A VLAN interface created by using the **interface vlan** *vlan-id* global configuration command. You will also need to enable IP PIM sparse mode on the VLAN, join the VLAN as a statically connected member to an MLD static group, and then enable MLD snooping on the VLAN, the MLD static group, and physical interface. These interfaces must have IPv6 addresses assigned to them.
Step 5	**ipv6 pim** **Example:** `Device(config-if)# ipv6 pim`	Enables the PIM on the interface.
Step 6	**ipv6 pim** {**bsr** | {**dr-priority** | *value*} | {**hello-interval** | *seconds*} | {**join-prune-interval** | *seconds*} | **passive**} **Example:** `Device(config-if)# ipv6 pim bsr|dr-priority|hello-interval|join-prune-interval|passive`	Configures the various PIM stub features on the interface. Enter **bsr** to configure BSR on a PIM switch Enter **dr-priority** to configure the DR priority on a PIM switch. Enter **hello-interval** to configure the frequency of PIM hello messages on an interface. Enter **join-prune-interval** to configure periodic join and prune announcement intervals for a specified interface. Enter **passive** to configure the PIM in the passive mode.

	Command or Action	Purpose
Step 7	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Disable 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.

Perform this procedure to disable embedded RP support in IPv6 PIM:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**no ipv6 pim** [**vrf** *vrf-name*] **rp embedded** **Example:** `Device(config)# no ipv6 pim rp embedded`	Disables embedded RP support in IPv6 PIM.
Step 4	**interface** *type number* **Example:** `Device(config)# interface FastEthernet 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.

Monitor IPv6 PIM stub routing

Table 20: PIM stub configuration show commands

Command	Purpose
show ipv6 pim interface	Displays the PIM stub that is enabled on each interface.
show ipv6 mld groups	Displays the interested clients that have joined the specific multicast source group.
show ipv6 mroute	Verifies that the multicast stream forwards from the source to the interested clients.

Configuring a BSR

This section explains how to configure BSR.

Configure a BSR and verify BSR information

Perform this procedure to configure and verify BSR Information:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 pim bsr candidate bsr** *ipv6-address* [*hash-mask-length*] [**priority** *priority-value*] **Example:** `Device(config)# ipv6 pim bsr candidate bsr 2001:DB8:3000:3000::42 124 priority 10`	Configures a switch to be a candidate BSR.
Step 4	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 1/0/1`	Specifies an interface type and number, and places the switch in interface configuration mode.

	Command or Action	Purpose
Step 5	**ipv6 pim bsr border** **Example:** `Device(config-if)# ipv6 pim bsr border`	Specifies an interface type and number, and places the switch in interface configuration mode.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Enter this command twice to exit interface configuration mode and enter privileged EXEC mode.
Step 7	**show ipv6 pim** bsr {**election** | **rp-cache** | **candidate-rp**} **Example:** `Device(config-if)# show ipv6 pim bsr election`	Displays information related to PIM BSR protocol processing.
Step 8	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Send PIM RP advertisements to the BSR

Perform this procedure to send PIM RP advertisements to the BSR:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 pim bsr candidate rp** *ipv6-address* [**group-list** *access-list-name*] **[priority** *priority-value*] [**interval** seconds] **Example:** `Device(config)# ipv6 pim bsr candidate rp 2001:DB8:3000:3000::42 priority 0`	Sends PIM RP advertisements to the BSR.

	Command or Action	Purpose
Step 4	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 1/0/1`	Specifies an interface type and number, and places the switch in interface configuration mode.
Step 5	**ipv6 pim bsr border** **Example:** `Device(config-if)# ipv6 pim bsr border`	Configures a border for all BSMs of any scope on a specified interface.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Configure BSR for use within scoped zones

Perform this procedure to configure BSR for use within scoped zones:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 pim bsr candidate rp** *ipv6-address* [*hash-mask-length*] [**priority** *priority-value*] **Example:** `Device(config)# ipv6 pim bsr candidate bsr 2001:DB8:1:1:4`	Configures a switch to be a candidate BSR.
Step 4	**ipv6 pim bsr candidate rp** *ipv6-address* [**group-list** *access-list-name*] [**priority** *priority-value*] [**interval** seconds] **Example:** `Device(config)# ipv6 pim bsr candidate rp 2001:DB8:1:1:1 group-list list scope 6`	Configures the candidate RP to send PIM RP advertisements to the BSR.

	Command or Action	Purpose
Step 5	**interface** *type number* **Example:** `Device(config-if)# interface GigabitEthernet 1/0/1`	Specifies an interface type and number, and places the switch in interface configuration mode.
Step 6	**ipv6 multicast boundary scope** *scope-value* **Example:** `Device(config-if)# ipv6 multicast boundary scope 6`	Configures a multicast boundary on the interface for a specified scope.
Step 7	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Configure BSR switches to announce scope-to-RP mappings

IPv6 BSR switches can be statically configured to announce scope-to-RP mappings directly. A user might configure a BSR switch to announce these mappings so that an RP not supporting BSR is incorporated into the BSR. This action also allows an RP located outside the enterprise's BSR domain to be detected by the local candidate BSR switch.

Perform this procedure to configure BSR switches to announce Scope-to-RP mappings:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 pim bsr announced rp** *ipv6-address* [**group-list** *access-list-name*] [**priority** *priority-value*] **Example:** `Device(config)# ipv6 pim bsr announced rp 2001:DB8:3000:3000::42 priority 0`	Announces scope-to-RP mappings directly from the BSR for the specified candidate RP.
Step 4	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Configure static mroutes

Static multicast routes (mroutes) in IPv6 can be implemented as an extension of IPv6 static routes. You can configure your switch to use a static route for unicast routing only, to use a static multicast route for multicast RPF selection only, or to use a static route for both unicast routing and multicast RPF selection.

Perform this procedure to configure static mroutes:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 route** {*ipv6-prefix / prefix-length ipv6-address* | *interface-type interface-number ipv6-address*]} [*administrative-distance*] [*administrative-multicast-distance* | *unicast* | *multicast*] [**tag** *tag*] **Example:** `Device(config)# ipv6 route 2001:DB8::/64 6::6 100`	Establishes static IPv6 routes. The example shows a static route used for both unicast routing and multicast RPF selection.
Step 4	**exit** **Example:** `Device# exit`	Exits global configuration mode, and returns the switch to privileged EXEC mode.
Step 5	**show ipv6 mroute** [link-local | [*group-name* | *group-address* [*source-address* | *source-name*]] **[summary**] [**count**] **Example:** `Device# show ipv6 mroute ff07::1`	Displays the contents of the IPv6 multicast routing table.
Step 6	**show ipv6 mroute** [**link-local** | *group-name* | *group-address*] **active** [*kbps*] **Example:** `Device(config-if)# show ipv6 mroute active`	Displays the active multicast streams on the switch.

	Command or Action	Purpose
Step 7	**show ipv6 rpf** [*ipv6-prefix*] **Example:** `Device(config-if)# show ipv6 rpf 2001::1:1:2`	Checks RPF information for a given unicast host address and prefix.
Step 8	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Verify MFIB operation in IPv6 multicast

Perform this procedure to verify MFIB operation in IPv6 multicast:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**show ipv6 mfib** [**verbose** | *group-address-name* | *ipv6-prefix / prefix-length* | *source-address-name* | **count** | **interface** | **status** | **summary**] **Example:** `Device# show ipv6 mfib`	Displays the forwarding entries and interfaces in the IPv6 MFIB.
Step 3	**show ipv6 mfib** [**all** | **linkscope** | *group-name* | *group-address* [*source-name* | *source-address*]] **count** **Example:** `Device# show ipv6 mfib ff07::1`	Displays the contents of the IPv6 multicast routing table.
Step 4	**show ipv6 mfib interface** **Example:** `Device# show ipv6 mfib interface`	Displays information about IPv6 multicast-enabled interfaces and their forwarding status.
Step 5	**show ipv6 mfib status** **Example:** `Device# show ipv6 mfib status`	Displays general MFIB configuration and operational status.

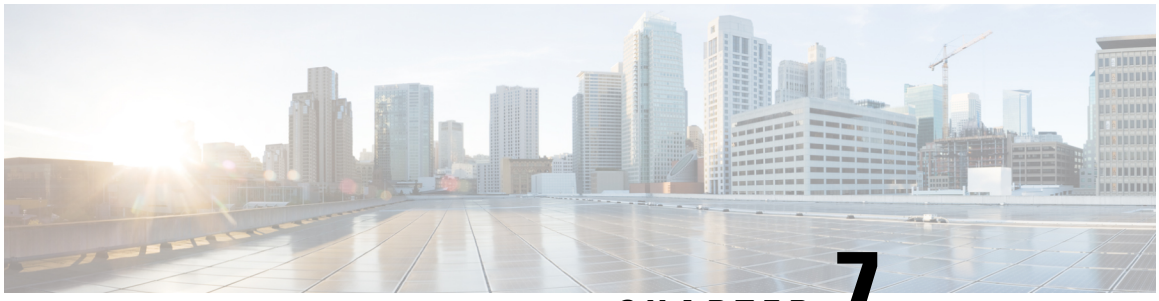
	Command or Action	Purpose
Step 6	**show ipv6 mfib summary** **Example:** `Device# show ipv6 mfib summary`	Displays summary information about the number of IPv6 MFIB entries and interfaces.
Step 7	**debug ipv6 mfib** [*group-name* | *group-address*] [**adjacency** | **db** | **fs** | **init** | **interface** | **mrib** [**detail**] | **nat** | **pak** | **platform** | **ppr** | **ps** | **signal** | **table**] **Example:** `Device# debug ipv6 mfib FF04::10 pak`	Enables debugging output on the IPv6 MFIB.

Reset MFIB traffic counters

Perform this procedure to reset MFIB traffic counters:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**clear ipv6 mfib counters** [*group-name* | **group-address** [*source-address* | *source-name*]] **Example:** `Device# clear ipv6 mfib counters FF04::10`	Resets all active MFIB traffic counters.



CHAPTER 7

MLD Snooping

Feature history for MLD snooping

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature name and description	Supported platform
Cisco IOS XE 17.18.1	MLD snooping: MLD snooping enables efficient distribution of IPv6 multicast data to clients and routers in a switched network on the device.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

Understand MLD snooping

Multicast Listener Discovery (MLD) snooping enables efficient distribution of IPv6 multicast data to clients and routers in a switched network on the device. Unless otherwise noted, the term device refers to a standalone device and to a device stack.

In IPv4, Layer 2 devices can use Internet Group Management Protocol (IGMP) snooping to limit the flooding of multicast traffic by dynamically configuring Layer 2 interfaces so that multicast traffic is forwarded to only those interfaces associated with IP multicast devices. In IPv6, MLD snooping performs a similar function. MLD snooping ensures that IPv6 multicast data is forwarded only to designated ports, avoiding flooding of all VLAN ports. This list is constructed by snooping IPv6 multicast control packets.

MLD is a protocol used by IPv6 multicast routers to discover the presence of multicast listeners (nodes wishing to receive IPv6 multicast packets) on the links that are directly attached to the routers and to discover which multicast packets are of interest to neighboring nodes. MLD is derived from IGMP; MLD Version 1 (MLDv1) is equivalent to IGMPv2, and MLD Version 2 (MLDv2) is equivalent to IGMPv3. MLD is a subprotocol of Internet Control Message Protocol Version 6 (ICMPv6), and MLD messages are a subset of ICMPv6 messages, identified in IPv6 packets by a preceding Next Header value of 58.

MLD snooping can be enabled or disabled globally or for each VLAN. When MLD snooping is enabled, IPv6 multicast address tables per VLAN are constructed both in software and hardware. The device then performs IPv6 multicast-address based bridging in hardware.

To use IPv6, you must configure the dual IPv4 and IPv6 Switch Database Management (SDM) template on the device.

MLD snooping versions

The device supports two versions of MLD snooping:

- MLDv1 snooping is a process that detects MLDv1 control packets to set up traffic bridging using IPv6 destination multicast addresses.
- MLDv2 basic snooping (MBSS) uses MLDv2 control packets to set up traffic forwarding based on IPv6 destination multicast addresses.

The device can perform snooping on both MLDv1 and MLDv2 protocol packets and bridge IPv6 multicast data based on destination IPv6 multicast addresses.

Note The device does not support MLDv2 enhanced snooping, which sets up IPv6 source and destination multicast address-based forwarding.

MLD messages

MLDv1 supports three types of messages:

- Listener Queries are the equivalent of IGMPv2 queries and are either General Queries or Multicast-Address-Specific Queries (MASQs).
- Multicast Listener Reports are the equivalent of IGMPv2 reports
- Multicast Listener Done messages are the equivalent of IGMPv2 leave messages.

MLDv2 supports MLDv2 queries and reports, as well as MLDv1 Report and Done messages.

Message timers and state transitions resulting from messages being sent or received are the same as those of IGMPv2 messages. MLD devices ignore MLD messages that do not have valid link-local IPv6 source addresses.

MLD queries

The device sends out MLD queries, constructs an IPv6 multicast address database, and generates MLD group-specific and MLD group-and-source-specific queries in response to MLD Done messages. The device

also supports report suppression, report proxying, Immediate-Leave functionality, and static IPv6 multicast group address configuration.

When MLD snooping is disabled, MLD queries flood the ingress VLAN. When MLD snooping is enabled, received MLD queries are flooded in the ingress VLAN, and a copy of the query is sent to the CPU for processing. From the received query, The system uses MLD snooping to build the IPv6 multicast address database. It detects multicast router ports, maintains timers, sets report response time, learns the querier IP source address for the VLAN, learns the querier port in the VLAN, and maintains multicast-address aging.

When a group exists in the MLD snooping database, the device responds to a group-specific query by sending an MLDv1 report. When the group is unknown, the group-specific query is flooded to the ingress VLAN.

When a host wants to leave a multicast group, it can send out an MLD Done message (equivalent to IGMP Leave message). When the device receives an MLDv1 Done message, if Immediate- Leave is not enabled, the device sends an MASQ to the port from which the message was received to determine if other devices connected to the port should remain in the multicast group.

Multicast client aging robustness

You can configure port membership removal from addresses based on the number of queries. A port's membership to an address is removed only when there are no reports to that address on the port for the configured number of queries. The default number is 2.

Multicast router discovery

Like IGMP snooping, MLD snooping performs multicast router discovery, with these characteristics:

- Ports configured by a user never age out.
- Dynamic port learning results from MLDv1 snooping queries and IPv6 PIMv2 packets.
- If there are multiple routers on the same Layer 2 interface, MLD snooping tracks the router on the port that most recently sent a control packet.
- Dynamic multicast router port aging uses a 5-minute default timer. If no control packet is received, the router is deleted from the port list.
- IPv6 multicast router discovery occurs when MLD snooping is enabled on the device.
- IPv6 multicast router control packets are always flooded to the ingress VLAN, irrespective of MLD snooping being enabled on the device.
- IPv6 multicast data is initially sent to all ingress VLAN. Upon discovering the first IPv6 multicast router port, unknown IPv6 multicast data is sent only to discovered router ports.

MLD reports

The processing of MLDv1 join messages is the same as that of IGMPv2. The device does not process or forward reports if it detects no IPv6 multicast routers in a VLAN. The device enters an IPv6 multicast group address in the VLAN MLD database when IPv6 multicast routers are detected and an MLDv1 report is received. Subsequently, the device forwards all IPv6 multicast traffic to the group within the VLAN using this address. When MLD snooping is disabled, reports are flooded in the ingress VLAN.

When MLD snooping is enabled, MLD report suppression (also known as listener message suppression) is automatically enabled. With report suppression, the device forwards the first MLDv1 report received by a group to IPv6 multicast routers; subsequent reports for the group are not sent to the routers. When MLD snooping is disabled, report suppression is disabled, and all MLDv1 reports are flooded to the ingress VLAN.

The device also supports MLDv1 proxy reporting. When an MLDv1 MASQ is received, the device responds with MLDv1 reports for the address on which the query arrived if the group exists in the device on another port and if the port on which the query arrived is not the last member port for the address.

MLD done messages and immediate-leave

When the Immediate-Leave feature is enabled, and a host sends an MLDv1 Done message (equivalent to an IGMP leave message), the port receiving the Done message is immediately removed from the group. Enable Immediate-Leave on VLANs. As with IGMP snooping, use this feature only on VLANs where a single host is connected to the port. If the port was the last member of a group, the group is also deleted, and the leave information is forwarded to the detected IPv6 multicast routers.

When Immediate Leave is not enabled in a VLAN (when there are multiple clients for a group on the same port) and a Done message is received on a port, an MASQ is generated on that port. You can control when a port membership is removed for an existing address in terms of the number of MASQs. A port is removed from an address membership if no MLDv1 reports exist for the configured number of queries.

Use the **ipv6 mld snooping last-listener-query count** global configuration command to configure the number of MASQs generated. The default number is 2.

The MASQ is sent to the IPv6 multicast address for which the Done message was sent. If there are no reports sent to the IPv6 multicast address specified in the MASQ during the device maximum response time, the port on which the MASQ was sent is deleted from the IPv6 multicast address database. The maximum response time is the time configured by using the **ipv6 mld snooping last-listener-query-interval** global configuration command. If the deleted port is the last member of the multicast address, the multicast address is also deleted, and device transmits the address leave information to all detected multicast routers.

Topology change notification processing

When topology change notification (TCN) solicitation is enabled by using the **ipv6 mld snooping tcn query solicit** global configuration command, MLDv1 snooping sets the VLAN to flood all IPv6 multicast traffic with a configured number of MLDv1 queries before it begins sending multicast data only to selected ports. You set this value by using the **ipv6 mld snooping tcn flood query count** global configuration command. The default is to send two queries. The device also generates MLDv1 global Done messages with valid link-local IPv6 source addresses when the device becomes the STP root in the VLAN or when it is configured by the user. This process is the same as that used in IGMP snooping.

Default MLD snooping configuration

Table 21: Default MLD snooping configuration

Feature	Default setting
MLD snooping (global)	Disabled.

Feature	Default setting
MLD snooping (per VLAN)	Enabled. MLD snooping must be globally enabled for VLAN MLD snooping to take place.
IPv6 Multicast addresses	None configured.
IPv6 Multicast router ports	None configured.
MLD snooping Immediate Leave	Disabled.
MLD snooping robustness variable	Global: 2; Per VLAN: 0. **Note** The VLAN value overrides the global setting. When the VLAN value is 0, the VLAN uses the global count.
Last listener query count	Global: 2; Per VLAN: 0. **Note** The VLAN value overrides the global setting. When the VLAN value is 0, the VLAN uses the global count.
Last listener query interval	Global: 1000 (1 second); VLAN: 0. **Note** The VLAN value overrides the global setting. When the VLAN value is 0, the VLAN uses the global interval.
TCN query solicit	Disabled.
TCN query count	2.
MLD listener suppression	Disabled.

MLD snooping configuration guidelines

Consider these guidelines when you configure MLD snooping:

- You can configure MLD snooping characteristics at any time, but you must enable MLD snooping globally using the **ipv6 mld snooping** global configuration command for the configuration to take effect.
- MLD snooping and IGMP snooping are independent. Enable them both on the device simultaneously.
- A device or device stack allows a maximum of 4000 address entries.

Configure MLD snooping

This section provides configuration information about MLD snooping.

Configure MLD snooping on the device

By default, IPv6 MLD snooping is globally disabled on the device and enabled on all VLANs. If MLD snooping is turned off globally, it is not active on any VLANs. When you globally enable MLD snooping, the VLAN configuration overrides the global configuration. That is, MLD snooping is enabled only on VLAN interfaces in the default state (enabled).

You can enable and disable MLD snooping on a per-VLAN basis or for a range of VLANs, but if you globally disable MLD snooping, it is disabled in all VLANs. If global snooping is enabled, you can enable or disable VLAN snooping.

To globally enable MLD snooping on the device, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 mld snooping** **Example:** `Device(config)# ipv6 mld snooping`	Enables MLD snooping on the device.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 5	**copy running-config startup-config** **Example:** `Device(config)# copy running-config startup-config`	(Optional) Save your entries in the configuration file.
Step 6	**reload** **Example:**	Reload the operating system.

	Command or Action	Purpose
	`Device(config)# reload`	

Configure MLD snooping on a VLAN

To enable MLD snooping on a VLAN, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 mld snooping** **Example:** `Device(config)# ipv6 mld snooping`	Enables MLD snooping on the device.
Step 4	**ipv6 mld snooping vlan** *vlan-id* **Example:** `Device(config)# ipv6 mld snooping vlan 1`	Enables MLD snooping on the VLAN. The VLAN ID range is 1 to 1001 and 1006 to 4094. **Note** MLD snooping must be globally enabled for VLAN snooping to be enabled.
Step 5	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Configure a static multicast group

While hosts or Layer 2 ports typically join multicast groups dynamically, you have the option to manually configure an IPv6 multicast address and member ports for a VLAN.

To add a Layer 2 port as a member of a multicast group, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 mld snooping vlan** *vlan-id* **static** *ipv6_multicast_address* **interface** *interface-id* **Example:** `Device(config)# ipv6 mld snooping vlan 1 static 3333.0000.1111 interface gigabitethernet1/0/1`	Configures a multicast group with a Layer 2 port as a member of a multicast group: • *vlan-id* is the multicast group VLAN ID. The VLAN ID range is 1 to 1001 and 1006 to 4094. • *ipv6_multicast_address* is the 128-bit group IPv6 address. The address must be in the form specified in RFC 2373. • *interface-id* is the member port. It can be a physical interface or a port channel (1 to 48).
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Step 5	Use one of the following:
	• **show ipv6 mld snooping address**
	• **show ipv6 mld snooping address vlan** *vlan-id*
	Example:
	`Device# show ipv6 mld snooping address`
	or
	`Device# show ipv6 mld snooping vlan 1`
		Verifies the static member port and the IPv6 address.

Configure a multicast router port

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

To add a multicast router port to a VLAN, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 mld snooping vlan** *vlan-id* **mrouter interface** *interface-id* **Example:** `Device(config)# ipv6 mld snooping vlan 1 mrouter interface gigabitethernet 1/0/2`	Specifies the multicast router VLAN ID, and specify the interface to the multicast router. • The VLAN ID range is 1 to 1001 and 1006 to 4094. • The interface can be a physical interface or a port channel. The port-channel range is 1 to 48.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 5	**show ipv6 mld snooping mrouter** [**vlan** *vlan-id*] **Example:** `Device# show ipv6 mld snooping mrouter vlan 1`	Verifies that IPv6 MLD snooping is enabled on the VLAN interface.

Enable MLD immediate leave

To enable MLDv1 immediate leave, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 mld snooping vlan** *vlan-id* **immediate-leave** **Example:** `Device(config)# ipv6 mld snooping vlan 1 immediate-leave`	Enables MLD Immediate Leave on the VLAN interface.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 5	**show ipv6 mld snooping vlan** *vlan-id* **Example:** `Device# show ipv6 mld snooping vlan 1`	Verifies that Immediate Leave is enabled on the VLAN interface.

Configure MLD snooping queries

To configure MLD snooping query characteristics for the device or for a VLAN, perform this procedure:

Procedure

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

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	[**no**] **ipv6 mld snooping robustness-variable** *value* **Example:** `Device(config)# ipv6 mld snooping robustness-variable 3`	(Optional) Sets the number of queries that are sent before device will deletes a listener (port) that does not respond to a general query. The range is 1 to 3; the default is 2. Use the no form of this command to disable this feature.
Step 4	**ipv6 mld snooping vlan** *vlan-id* **robustness-variable** *value* **Example:** `Device(config)# ipv6 mld snooping vlan 1 robustness-variable 3`	(Optional) Sets the robustness variable on a VLAN basis, which determines the number of general queries that MLD snooping sends before aging out a multicast address when there is no MLD report response. The range is 1 to 3; the default is 0. When set to 0, the number that is used is the global robustness variable value.
Step 5	[**no**] **ipv6 mld snooping last-listener-query-count** *count* **Example:** `Device(config)# ipv6 mld snooping last-listener-query-count 7`	(Optional) Sets the number of MASQs that the device sends before aging out an MLD client. The range is 1 to 7; the default is 2. The queries are sent 1 second apart. Use the no form of this command to disable this feature.
Step 6	**ipv6 mld snooping vlan** *vlan-id* **last-listener-query-count** *count* **Example:** `Device(config)# ipv6 mld snooping vlan 1 last-listener-query-count 7`	(Optional) Sets the last-listener query count on a VLAN basis. This value overrides the value that is configured globally. The range is 1 to 7; the default is 0. When set to 0, the global count value is used. Queries are sent 1 second apart.
Step 7	[**no**] **ipv6 mld snooping last-listener-query-interval** *interval* **Example:** `Device(config)# ipv6 mld snooping last-listener-query-interval 2000`	(Optional) Sets the maximum response time that the device waits after sending out a MASQ before deleting a port from the multicast group. The range is 100 to 32,768 thousands of a second. The default is 1000 (1 second). Use the no form of this command to disable this feature.
Step 8	**ipv6 mld snooping vlan** *vlan-id* **last-listener-query-interval** *interval* **Example:** `Device(config)# ipv6 mld snooping vlan 1 last-listener-query-interval 2000`	(Optional) Sets the last-listener query interval on a VLAN basis. This value overrides the value that is configured globally. The range is 0 to 32,768 thousands of a second. The default is 0. When set to 0, the global last-listener query interval is used.

	Command or Action	Purpose
Step 9	[**no**] **ipv6 mld snooping tcn query solicit** **Example:** `Device(config)# ipv6 mld snooping tcn query solicit`	(Optional) Enables topology change notification (TCN) solicitation, which means that VLANs flood all IPv6 multicast traffic for the configured number of queries before sending multicast data to only those ports requesting to receive it. The default is for TCN to be disabled. Use the no form of this command to disable this feature.
Step 10	[**no**] **ipv6 mld snooping tcn flood query count** *count* **Example:** `Device(config)# ipv6 mld snooping tcn flood query count 5`	(Optional) When TCN is enabled, specifies the number of TCN queries to be sent. The range is from 1 to 10; the default is 2. Use the no form of this command to disable this feature.
Step 11	[**no**] **ipv6 mld snooping querier** **Example:** `Device(config)# ipv6 mld snooping querier`	(Optional) Enables MLD snooping queries. Use the no form of this command to disable MLD snooping queries.
Step 12	[**no**] **ipv6 mld snooping vlan** *vlan_id* **querier** **Example:** `Device(config)# ipv6 mld snooping vlan 1 querier`	(Optional) Enables MLD snooping queries in a VLAN. Use the no form of this command to disable MLD snooping queries in a VLAN.
Step 13	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 14	**show ipv6 mld snooping querier** [**vlan** *vlan-id*] **Example:** `Device# show ipv6 mld snooping querier vlan 1`	(Optional) Verifies that the MLD snooping querier information for the device or for the VLAN.

Disable MLD listener message suppression

MLD snooping listener message suppression is enabled by default. When it is enabled, the device forwards only one MLD report per multicast router query. When message suppression is disabled, multiple MLD reports could be forwarded to the multicast routers.

To disable MLD listener message suppression, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enter global configuration mode.
Step 3	**no ipv6 mld snooping listener-message-suppression** **Example:** `Device(config)# no ipv6 mld snooping listener-message-suppression`	Disable MLD message suppression.
Step 4	**end** **Example:** `Device(config)# end`	Return to privileged EXEC mode.
Step 5	**show ipv6 mld snooping** **Example:** `Device# show ipv6 mld snooping`	Verify that IPv6 MLD snooping report suppression is disabled.

Monitor MLD snooping configuration

You can display MLD snooping information for dynamically learned and statically configured router ports and VLAN interfaces. You can also display IPv6 group address multicast entries for a VLAN configured for MLD snooping.

Table 22: Commands for displaying MLD snooping configuration

Command	Purpose
show ipv6 mld snooping [**vlan** *vlan-id*]	Displays the MLD snooping configuration information for all VLANs on the device or for a specified VLAN. (Optional) Enter **vlan** *vlan-id* to display information for a single VLAN. The VLAN ID range is 1 to 1001 and 1006 to 4094.

Command	Purpose
show ipv6 mld snooping mrouter [**vlan** *vlan-id*]	Displays information on dynamically learned and manually configured multicast router interfaces. When you enable MLD snooping, the device automatically learns the interface to which a multicast router is connected. These are dynamically learned interfaces. (Optional) Enters **vlan** *vlan-id* to display information for a single VLAN. The VLAN ID range is 1 to 1001 and 1006 to 4094.
show ipv6 mld snooping querier [**vlan** *vlan-id*]	Displays information about the IPv6 address and incoming port for the most-recently received MLD query messages in the VLAN. (Optional) Enters **vlan** *vlan-id* to display information for a single VLAN.The VLAN ID range is 1 to 1001 and 1006 to 4094.
show ipv6 mld snooping address [**vlan** *vlan-id*] [**count** | **dynamic** | **user**]	Displays all IPv6 multicast address information or specific IPv6 multicast address information for the device or a VLAN. • Enters **count** to show the group count on the device or in a VLAN. • Enters **dynamic** to display MLD snooping learned group information for the device or for a VLAN. • Enters **user** to display MLD snooping user-configured group information for the device or for a VLAN.
show ipv6 mld snooping address vlan *vlan-id* [*ipv6-multicast-address*]	Displays MLD snooping for the specified VLAN and IPv6 multicast address.

Configuration examples

Refer this section for configuration examples of MLD snooping.

Example: Configure a static multicast group

This example shows how to statically configure an IPv6 multicast group:

```
Device# configure terminal
Device(config)# ipv6 mld snooping vlan 2 static 3333.0000.1111 interface gigabitethernet1/0/1
Device(config)# end
```

Example: Configure a multicast router port

This example shows how to add a multicast router port to VLAN 200:

```
Device# configure terminal
Device(config)# ipv6 mld snooping vlan 200 mrouter interface gigabitethernet1/0/2
Device(config)# exit
```

Example: Enable MLD immediate leave

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

```
Device# configure terminal
Device(config)# ipv6 mld snooping vlan 130 immediate-leave
Device(config)# exit
```

Example: Configure MLD snooping queries

This example shows how to set the MLD snooping global robustness variable to 3:

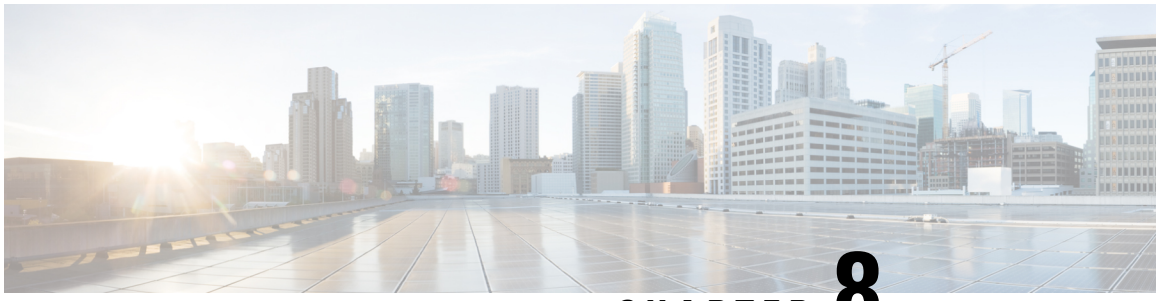
```
Device# configure terminal
Device(config)# ipv6 mld snooping robustness-variable 3
Device(config)# exit
```

This example shows how to set the MLD snooping last-listener query count for a VLAN to 3:

```
Device# configure terminal
Device(config)# ipv6 mld snooping vlan 200 last-listener-query-count 3
Device(config)# exit
```

This example shows how to set the MLD snooping last-listener query interval (maximum response time) to 2000 (2 seconds):

```
Device# configure terminal
Device(config)# ipv6 mld snooping last-listener-query-interval 2000
Device(config)# exit
```



CHAPTER 8

Mroute Limit and IGMP Limit

Feature history for mroute limit and IGMP limit

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature name and description	Supported platform
Cisco IOS XE 17.18.1	Mroute limit: The Multicast Route Limit feature allows global and per MVRF state limiters configuration, which impose limits on the number of mroutes that can be added to the global table or to a particular MVRF table. IGMP limit: The IGMP State Limit feature allows IGMP state limiters configuration, which impose limits on mroute states resulting from IGMP membership reports.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

Understand mroute limit and IGMP limit

The Multicast Route Limit feature allows global and per MVRF state limiters configuration, 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.

The IGMP State Limit feature allows IGMP state limiters configuration, which impose limits on mroute states resulting from Internet Group Management Protocol (IGMP) membership reports (IGMP joins).

Mroute state limit

Multicast route state limit refers to a limit on the number of multicast routing entries (mroute states) that a device can maintain. This limit helps prevent control plane overload by restricting the number of multicast routes the device processess and stores, ensuring stable and fair resource usage. In Multi-VRF (MVRF) environments, the MVRF mroute state limit allows administrators to set multicast route limits individually for each VRF.

Global mroute state limiters limit the number of mroutes added to the global table on a device. Configuring a global mroute state limiter protects a device in a multicast DoS attack by preventing mroutes from overrunning the device.

Per VRF mroute state limiters limit the number of mroutes added to an MVRF table. Use per MVRF mroute state limits to ensure fair sharing of mroutes between different MVRFs.

Mroute state limit feature design

Global and per MVRF mroute state limiters are configured using the **ip multicast route-limit** command in global configuration mode. This command imposes limits on the number of mroutes that can be added to the global table or to a particular MVRF table, respectively.

The syntax of the **ip multicast route-limit** command is as follows:

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

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

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.

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

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

Mechanics of mroute state limiters

Here is how global and per MVRF mroute state limiters work:

- When an mroute state is created on a device, the Cisco IOS software checks if the global or per MVRF mroute state limiter's limit has been reached.

- States for mroutes that exceed the configured limit for the global or the per MVRF mroute state limiter are not created on the device, and a warning message in this format is generated:

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

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

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

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

IGMP state limit

The IGMP State Limit feature allows the configuration of IGMP state limiters to impose limits on mroute states resulting from IGMP membership reports (IGMP joins), applicable globally or on a per-interface basis. Membership reports that exceed the configured limits are excluded from the IGMP cache. You can use the IGMP State Limit feature to prevent DoS attacks and enable a multicast CAC mechanism in network environments where multicast flows utilize similar bandwidth.

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

IGMP state limit feature design

- In global configuration mode, configure IGMP state limiters to set a device-wide limit on cached IGMP membership reports.
- In interface configuration mode, configure IGMP state limiters to limit the number of IGMP membership reports per interface.
- Use ACLs to exclude certain groups or channels from being counted against the interface limit. A standard or an extended ACL can be specified. Standard ACL usage: Define the (*, G) state to be excluded from an interface's limit. Extended ACL usage: Define the (S, G) state excluded from the interface limit. 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:

- When a device receives an IGMP membership report for a group or channel, the Cisco IOS software determines whether the limit for either the global IGMP state limiter or the per interface IGMP state limiter is reached.
- IGMP membership reports are honored if only a global IGMP state limiter is configured and its limit has not been reached. When the configured limit is reached, subsequent IGMP membership reports are ignored, and a warning message is generated in these formats:

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

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

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

Prerequisites for mroute limit and IGMP limit

- Ensure IP multicast is enabled and configure the Protocol Independent Multicast (PIM) interfaces.
- Configure MVRFs on the PE device before configuring per MVRF mroute state limiters.

Configure mroute limit and IGMP limit

This section provides configuration information about mroute limit and IGMP limit.

Configure 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.

Procedure

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

	Command or Action	Purpose
Step 3	**ip multicast route-limit** *limit* [*threshold*] **Example:** `Device(config)# ip multicast route-limit 1500 1460`	Limits the number of mroutes that can be added to the global table. • For the required *limit* argument, specify the limit on the number of mroutes that can be added to the global table. The range is from 1 to 2147483647. • Use the optional *threshold*argument to set an mroute threshold limit. The range is from 1 to 2147483647.
Step 4	**end** **Example:** `Device(config)# end`	Ends the current configuration session and returns to privileged EXEC mode.
Step 5	**show ip mroute count** **Example:** `Device# show ip mroute count`	(Optional) Displays mroute data and packet count statistics. • Use this command to verify the number of mroutes in the global table.

Configure per MVRF mroute state limiters

Perform this optional task to configure per MVRF mroute state limiters, limiting the number of mroutes added to a specific MVRF table.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast vrf** *vrf-name* **route-limit** *limit* [*threshold*] **Example:** `Device(config)# ip multicast vrf red route-limit 1500 1460`	Limits the number of mroutes that can be added to a particular MVRF table. • For the **vrf** keyword and *vrf-name* argument, specify the MVRF for which to apply the limit.

	Command or Action	Purpose
		• For the required *limit* argument, specify the limit on the number of mroutes that can be added to the MVRF table (for the specified MVRF). The range is from 1 to 2147483647. • Use the optional *threshold* argument to set an mroute threshold limit. The range is from 1 to 2147483647.
Step 4	**end** **Example:** `Device(config)# end`	Ends the current configuration session and returns to privileged EXEC mode.
Step 5	**show ip mroute vrf** *vrf-name* **count** **Example:** `Device# show ip mroute vrf red count`	(Optional) Displays mroute data and packet count statistics related to the specified MVRF. • Use this command to verify the number of mroutes in a particular MVRF table.

Configure global IGMP state limiters

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip igmp limit** *number* **Example:** `Device(config)# ip igmp limit 150`	Configures a global limit on the number of mroute states resulting from IGMP membership reports (IGMP joins).
Step 4	**end** **Example:**	Ends the current configuration session and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-if)# end`	
Step 5	**show ip igmp groups** **Example:** `Device# show ip igmp groups`	(Optional) Displays the multicast groups with receivers that are directly connected to the device and that were learned through IGMP.

Configure per interface IGMP state limiters

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

Procedure

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

Step 5	Do one of these:
	• **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.

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

Configuration examples

Refer this section for configuration examples of mroute limit and IGMP limit.

Example: Configure mroute state limiters

This 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
```

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

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

This is a sample mroute exceeded warning message. The output shows that the configured mroute limit of 1500 has been exceeded by one mroute. The device does not create states for mroutes that exceed the configured limit of the global mroute state limiter.

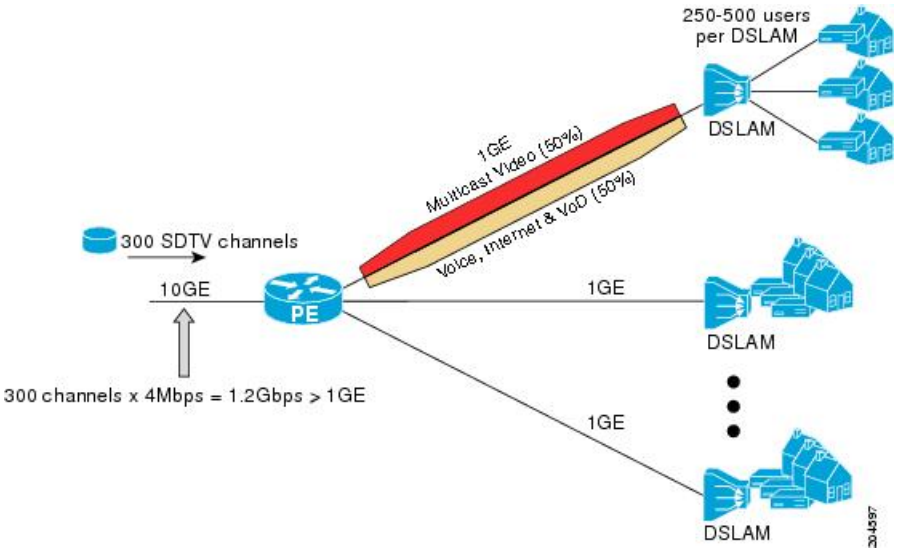
```
%MROUTE-4-ROUTELIMIT : 1501 routes exceeded multicast route-limit of 1500
```

Example: Configure IGMP state limiters

This 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. Although this illustration and example uses devices in the configuration, any device can be used.

Figure 20: IGMP state limit example topology



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

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

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

500Mbps / 4Mbps = 125 mroutes

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

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

```
interface GigabitEthernet0/0/0
description --- Interface towards the DSLAM ---
.
.
.
ip igmp limit 125
```