IP Multicast Routing Configuration Guide, Cisco Catalyst IE3x00 Rugged, IE3400 Heavy Duty, and ESS3300 Series Switches

Cisco Catalyst IE3200 Rugged Series
Updated March 31, 2025

1 Configuring Basic Multicast Routing

Topics:

- Prerequisites for basic IP multicast routing
- Restrictions for basic IP multicast routing
- Information about basic IP multicast routing
- How to configure basic IP multicast routing
- Monitoring and maintaining basic IP multicast routing
- Configuration examples for basic IP multicast routing
- Feature information for basic IP multicast routing

Prerequisites for basic IP multicast routing

Outlines prerequisites for IP multicast routing configuration.

Configure multicast routing protocols, such as Protocol Independent Multicast (PIM), to enable IP multicast routing.The switch populates its multicast routing table based on the mode setting. It forwards multicast packets received from directly connected LANs. You can configure an interface to operate in PIM dense mode or sparse mode.

- Ensure that IGMP operation is enabled when you enable PIM on an interface.
- Enabling PIM on an interface also enables IGMP operation on that interface.
- Multicast hosts, routers, and multilayer devices must enable IGMP to participate in IP multicasting.

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

Restrictions for basic IP multicast routing

Consider these restrictions when implementing IP multicast routing.

Review these restrictions when configuring IP multicast routing:

- PIM enabled interface scale limit is 26 targets (Routed port + SVIs).
- PIM sparse-dense mode is not supported.
- IPv6 PIM dense mode is not supported.

Information about basic IP multicast routing

Describes how IP multicast routing enables efficient network resource usage by allowing a source host to send packets to multiple receivers using special multicast group addresses, with routers forwarding packets only to group members.

IP multicasting enables efficient use of network resources. It allows a source host to send packets to multiple receivers using a specific multicast group address.

- uses special IP multicast group addresses for destination routing,
- forwards packets only to interfaces leading to group members, and
- supports bandwidth-intensive services such as audio and video.

Multicast routing operations and support

The sending host inserts the multicast group address into the IP destination address field of the packet. IP multicast routers and multilayer devices forward incoming IP multicast packets out all interfaces that lead to members of the multicast group. Any host, regardless of whether it is a member of a group, can send to a group, but only members receive the message.

Starting from the IOS XE 17.17.1 release, multicast is supported over port channels or EtherChannels. This ensures that multicast traffic is efficiently distributed across multiple physical links within a port channel while maintaining redundancy and preventing duplicate traffic.

Multicast forwarding information base

Describes the Multicast Forwarding Information Base architecture and its role in IP multicast implementation.

A Multicast Forwarding Information Base (MFIB) is a multicast routing protocol–independent forwarding engine that

- provides modularity and separation between the multicast control plane and the multicast forwarding plane,
- operates independently of PIM or any other multicast routing protocol, and
- works with the Multicast Routing Information Base (MRIB) for IP multicast implementation.

MFIB responsibilities

The MFIB is responsible for:

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

The MRIB serves as the communication channel between MRIB clients, such as PIM, IGMP, the multicast routing (mroute) table, and the MFIB.

The switch controller device uses the MFIB architecture and the MRIB for IP multicast. This architecture is used in Cisco IOS IPv6 multicast implementations.

Default IP multicast routing configuration

Lists the default IP multicast routing configuration settings.

This reference provides the default configuration settings for IP multicast routing features and parameters.

Table 1: 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 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.

How to configure basic IP multicast routing

Describes how to configure basic IP multicast routing functionality to enable the forwarding of multicast traffic across network devices and establish multicast communication paths.

This section provides information about configuring basic IP multicast routing.

Configuring basic IP multicast routing

Configure basic IP multicast routing to enable multicast traffic forwarding and routing table population.

This task enables IP multicast routing on a switch to populate the multicast routing table and forward multicast packets from directly connected LANs according to the configured mode setting.

Before you begin

By default, multicast routing is disabled, and the system does not apply a mode setting.

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.

During multicast routing table population, dense-mode interfaces are always added to the table. If multicast traffic from a specific source is sufficient, the receiver's first-hop router might send join messages toward the source to build a source-based distribution tree.

Follow these steps to configure basic IP multicast routing:

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **interface** *interface-id* command to specify the Layer 3 interface on which you want to enable multicast routing, and enter the interface configuration mode.

   Example

   ```
   Device(config)# interface gigabitethernet 1/3
   ```

   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 must also enable IP PIM mode on the interface and join the interface to an IGMP static group as a statically connected member.
   - An SVI—A VLAN interface created by using the **interface VLAN** *VLAN-id* global configuration command. You must also enable IP PIM mode on the VLAN, join the VLAN to an IGMP static group as a statically connected member, and enable IGMP snooping on the VLAN, IGMP static group, and physical interface.

   These interfaces must have IP addresses assigned to them.

3. Use the **IP PIM** {**dense-mode** | **sparse-mode** } command to enable PIM mode on the interface.

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.

> **Note**
>
> To disable PIM on an interface, use the **no IP PIM** interface configuration command.

4. Use the **end** command to return to privileged EXEC mode.

Example

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

5. Use the **show running-config** command to verify your entries.

Example

```
Device# show running-config
```

Configure IP multicast forwarding

Configure IP multicast forwarding to enable IPv4 Multicast Forwarding Information Base (MFIB) interrupt-level IP multicast forwarding of incoming and outgoing packets on the device.

This task enables IPv4 Multicast Forwarding Information Base (MFIB) interrupt-level IP multicast forwarding for incoming and outgoing packets on the device.

Use this procedure to configure IPv4 Multicast Forwarding Information Base (MFIB) interrupt-level IP multicast forwarding for incoming or outgoing packets on the device.

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

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **ip mfib** command to enable IP multicast forwarding.

   Example

   ```
   Device(config)# ip mfib
   ```

3. Use the **exit** command to return to privileged EXEC mode.

   Example

   ```
   Device(config)# exit
   ```

4. (Optional) Use the **show running-config** command to verify your entries.

   Example

   ```
   Device# show running-config
   ```

Configuring optional IP multicast routing features

Describes the configuration procedures for optional IP multicast routing features that enhance multicast functionality beyond basic routing capabilities.

This section provides information about configuring optional IP multicast routing features.

Define the IP multicast boundary

Configure 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.

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

> **Note**
>
> Support for this feature on IPv6 was introduced in IOS XE Release 17.3.1. It is not supported on IPv4.

Follow these steps to define the IP multicast boundary:

This procedure is optional.

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **access-list** *access-list-number* **deny** *source* [*source-wildcard*] command to create a standard access list that denies Auto-RP multicast addresses.

   Example

   ```
   Device(config)# access-list 12 deny 224.0.1.39
   Device(config)# access-list 12 deny 224.0.1.40
   ```

   Repeat the command 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. The wildcard bits are applied to the source. Place ones in the bit positions you want to ignore.

   The access list is always terminated by an implicit deny statement for everything.

3. Use the **interface** *interface-id* command to specify the interface on which you want to configure the multicast boundary, and enter interface configuration mode.

   Example

   ```
   Device(config)# interface gigabitethernet 1/3
   ```

4. Use the **ip multicast boundary** *access-list-number* command to configure the multicast boundary.

   Example

   ```
   Device(config-if)# ip multicast boundary 12
   ```

   Specify the access list that you created earlier in this procedure.

5. Use the **end** command to return to privileged EXEC mode.

Example

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

6. Use the **show running-config** command to verify your entries.

Example

```
Device# show running-config
```

Monitoring and maintaining basic IP multicast routing

Describes the procedures for monitoring IP multicast routing operations and performing maintenance tasks to ensure proper multicast traffic delivery and network performance.

Clear caches, tables, and databases

Lists commands for clearing caches, tables, and databases when contents are invalid or suspected to be invalid.

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.

Table 2: Commands for clearing caches, tables, and databases

You can use any of the privileged EXEC commands in this table to clear IP multicast 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 mroute { ***** | [*hostname* | *IP address*] | **vrf** *name* **group** [*hostname* | *IP address*] }	Deletes entries from the IP multicast routing table.
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

Lists commands for displaying system and network statistics.

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

You can display information to learn resource usage and solve network problems. You can also display information about node reachability. Additionally, you can discover the routing path that packets from your device take through the network.

Table 3: Commands for displaying system and network statistics

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

Command	Purpose
ping [*group-name* | *group-address*]	Sends an ICMP Echo Request to a multicast group address.

Command	Purpose
show ip igmp filter	Displays IGMP filter information.
show ip igmp groups [*group-name*|*group-address*|*type-number*]	Displays the multicast groups that are directly connected to the device and that were learned through IGMP.
show ip igmp interface [*type number*]	Displays multicast-related information about an interface.
show ip igmp profile [profile_number]	Displays IGMP profile information.
show ip igmp ssm-mapping [*hostname/ip address*]	Displays IGMP SSM mapping information.
show ip igmp static-group {**class-map** [**interface** [type]]	Displays static group information.
show ip igmp membership [*name/group address* | **all** | **tracked**]	Displays IGMP membership information for forwarding.
show ip igmp vrf	Displays the selected VPN Routing/Forwarding instance by name.
show ip mfib [*type number*]	Displays the IP multicast forwarding information base.
show ip mrib { **client** | **route** | **VRF** }	Displays the multicast routing information base.
show ip mrm { **interface** | **manager** | **status-report** }	Displays the IP multicast routing monitor information.
show ip mroute [*group-name* | *group-address*] [*source*] [**count** | **interface** | **proxy** | **pruned** | **summary** | **verbose**]	Displays the contents of the IP multicast routing table.
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 autorp	Display global auto-RP 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.
show ip pim neighbor [*type number*]	Lists the PIM neighbors discovered by the device.
show ip pim rp [*group-name* | *group-address*]	Displays the RP routers associated with a sparse-mode multicast group.
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 for basic IP multicast routing

Describes configuration examples for implementing basic IP multicast routing functionality. Provides practical setup scenarios and commands for enabling multicast routing in network environments.

This section provides configuration examples for Basic IP Multicast Routing.

Configure an IP multicast boundary (ipv6 only)

Describes the configuration of an IP multicast boundary to control the scope of multicast traffic. This process prevents administratively-scoped multicast packets from crossing network boundaries.

IP multicast boundaries are used to limit the scope of multicast traffic within network segments. This configuration prevents certain multicast groups from being forwarded across specific interfaces.

```
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 gigabitethernet 1/3
Device(config-if)# ip multicast boundary 1
```

Feature information for basic IP multicast routing

Lists the feature information for basic IP multicast routing across different Cisco IOS XE releases.

This reference provides feature information for basic IP multicast routing, including IP multicast support over port channel and basic IP multicast routing capabilities across different Cisco IOS XE releases and supported platforms.

Table 4: Feature information for basic IP multicast routing

Feature Name	Release	Feature Information
IP Multicast Support Over Port Channel	Cisco IOS XE 17.17.1	This feature is supported on the Cisco Catalyst IE3400 Rugged, IE3400 Heavy Duty, and IES-30 Series Switches

Feature Name	Releases	Feature Information
Basic IP Multicast Routing	Cisco IOS XE 17.3.1	IP Multicast is an efficient way to use network resources, especially for bandwidth-intensive services such as audio and video. IP multicast routing enables a host (source) to send packets to a group of hosts (receivers) anywhere within the IP network by using a special form of IP address called the IP multicast group address.

2 Configuring MSDP

Topics:

- Prerequisites for using MSDP to interconnect multiple PIM-SM domains
- Information about using MSDP to interconnect multiple PIM-SM domains
- How to use MSDP to interconnect multiple PIM-SM domains
- Configuration examples for using MSDP to interconnect multiple PIM-SM domains
- Troubleshooting tips

Prerequisites for using MSDP to interconnect multiple PIM-SM domains

Describes the requirement that all MSDP peer addresses must be known in Border Gateway Protocol (BGP) before configuring MSDP to interconnect multiple PIM-SM domains.

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

Information about using MSDP to interconnect multiple PIM-SM domains

Describes how to use Multicast Source Discovery Protocol (MSDP) to interconnect multiple Protocol Independent Multicast Sparse Mode (PIM-SM) domains for multicast routing across domain boundaries.

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

Benefits of using MSDP to interconnect multiple PIM-SM domains

Lists the advantages of using MSDP for interconnecting multiple PIM-SM domains in multicast network environments.

This reference provides the key benefits of using Multicast Source Discovery Protocol (MSDP) to interconnect multiple Protocol Independent Multicast Sparse Mode (PIM-SM) domains, highlighting operational advantages and management improvements.

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

How MSDP interconnects multiple PIM-SM domains

Describes how MSDP enables multiple PIM-SM domains to discover multicast sources across domain boundaries using interdomain source trees and Source-Active message exchanges between rendezvous points.

MSDP connects multiple PIM-SM domains and enables discovery of multicast sources in other PIM domains. The main advantage of MSDP is that it reduces interconnection complexity. It allows the use of an interdomain source tree rather than a common shared tree.

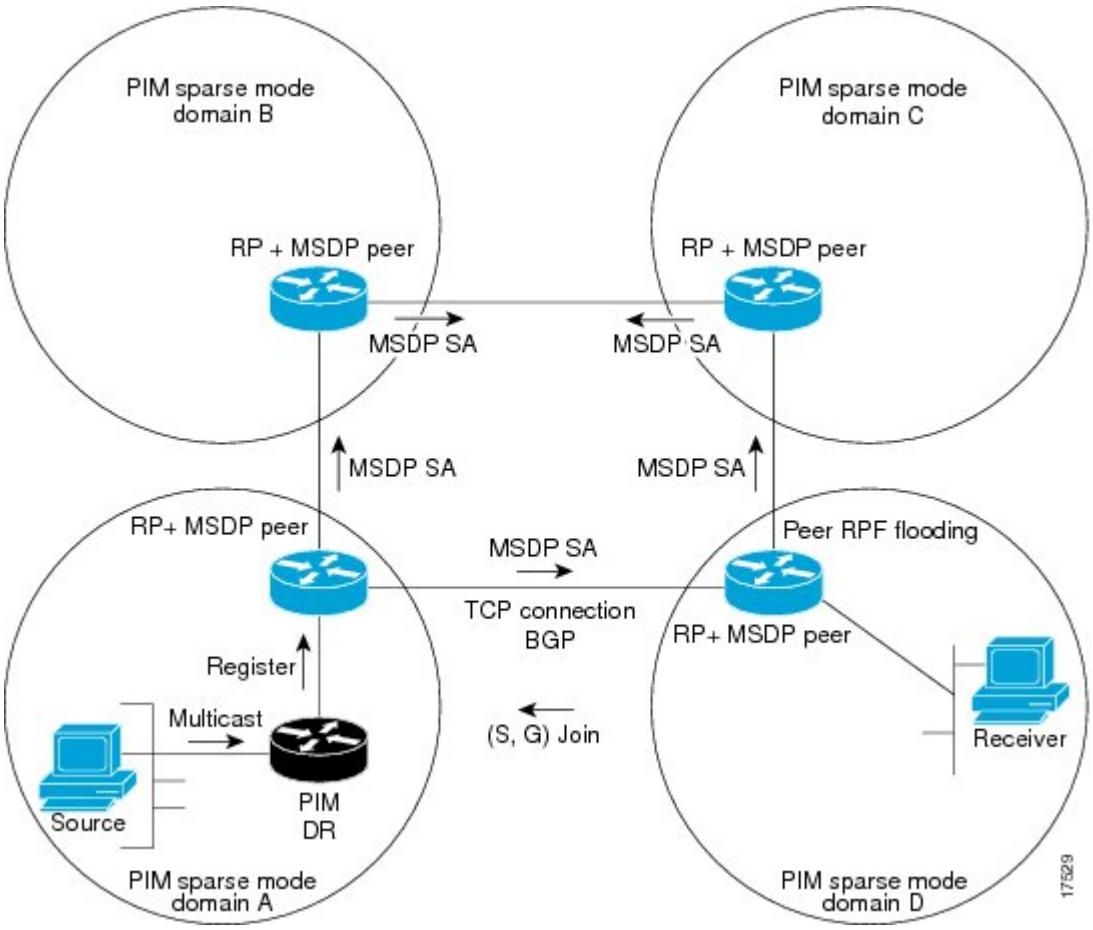
Summary

The key components involved in MSDP interconnection of PIM-SM domains are:

- Rendezvous Points (RPs): Exchange source information with RPs in other domains and maintain MSDP peering relationships over TCP connections
- PIM designated routers (DRs): Register sources with their local RP within each domain
- Source-Active (SA) messages: Contain source address, group information, and originator RP address to announce active sources
- BGP routing system: Provides the underlying connectivity for TCP peering and supports peer-RPF flooding mechanism
- Interdomain source tree: Allows multicast data delivery across domain boundaries when receivers exist

Workflow

Figure 1: MSDP running between RP peers



Note

Although the illustration uses routers in the configuration, any device, such as a router or switch, can be used.

These stages describe how MSDP interconnects multiple PIM-SM domains:

1. When a source becomes active, the PIM designated router (DR) registers the source with its local RP, which triggers the RP to send a Source-Active (SA) message to all of its MSDP peers.

   Note

   The DR sends the encapsulated data to the RP only once per source (when the source goes active). If the source times out, this process repeats when the source becomes active again. This situation differs from periodic SA messages, which contain all sources registered to the originating RP. SA messages are MSDP control packets. They do not include encapsulated data from active sources.

2. The SA message identifies the source address, the group the source is sending to, and the address or originator ID of the RP, if configured.
3. Each MSDP peer that receives the SA message floods it to all downstream peers by using the peer-RPF flooding mechanism.

- 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
- That next-hop neighbor becomes the RPF-peer for the originator
- SA messages received from any interface other than the interface to the RPF peer are dropped

> **Note**
>
> - (M)BGP is not required in MSDP mesh group scenarios. For more information about MSDP mesh groups, see the Configure an MSDP mesh group section.
> - (M)BGP is not required in default MSDP peer scenarios or in scenarios where only one MSDP peer is configured. For more information, see the Configure a default MSDP peer section.

4. When an RP receives an SA message, it determines whether to join the interdomain source tree based on local group membership.
   - The RP checks whether there are any members of the advertised groups in its domain by examining 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.
   - 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 forwarded down the RP's shared tree to the group members in its domain.
   - The members' DRs can then join the rendezvous point tree (RPT) to the source using standard PIM-SM procedures.

> **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.

5. The originating RP continues to send periodic SA messages for the (S, G) state every 60 seconds, as long as the source is sending packets to the group. Receiving RPs cache SA messages for future use.
   - When an RP receives an SA message, it caches the SA message
   - If the RP finds no active group members, it passes the SA message to downstream peers.
   - If a host later joins the group, the RP can immediately use the cached SA information to join the source tree

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

> **Note**
>
> MSDP depends on BGP for interdomain operation. We recommend running MSDP on RPs that send to global multicast groups.

MSDP message types

Describes the four basic MSDP message types that are each encoded in their own Type, Length, and Value (TLV) data format for multicast source discovery protocol communication.

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

SA messages

Describes multicast protocol messages that advertise active sources in a domain.

SA messages are multicast protocol messages that perform the following functions:

- advertise active sources in a domain,
- contain the IP address of the originating RP and one or more (S, G) pairs being advertised, and
- may contain the initial multicast data packet that was sent by the source or an encapsulated data packet.

Additional information

> **Note**
>
> For more information about SA messages, see the SA message origination receipt and processing on page 19 section.

SA request messages

Describes messages used to request active source lists for specific groups from MSDP SA cache.

SA request messages are MSDP protocol messages that

- request a list of active sources for a specific group
- are sent to an MSDP SA cache, which maintains a list of active (S, G) pairs, and
- reduce join latency by requesting the list of active sources for a group, which eliminates the need to wait up to 60 seconds for all active sources in the group to be readvertised by originating RPs.

Additional information

> **Note**
>
> For more information about SA request messages, see the Request source information from MSDP peers section.

SA response messages

Describes messages sent by MSDP peers in response to SA request messages.

An SA response message is a network message that

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

Additional information

> **Note**
>
> For more information about SA response messages, see the Control the response to outgoing SA request messages from MSDP peers using SA request filters section.

Keepalive messages

Describes the periodic messages that maintain MSDP session connectivity.

A keepalive message is a periodic communication mechanism that maintains active MSDP sessions by transmitting every 60 seconds to prevent session timeouts.

Session timeout behavior

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 section.

SA message origination receipt and processing

Describes the detailed processes for Security Association (SA) message origination, receipt, and processing within the system.

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

SA message origination

Describes how SA messages are triggered by an RP when new sources become active within a local PIM-SM domain.

SA message origination is a MSDP mechanism that triggers SA messages when any new source goes active within a local PIM-SM domain.

SA message origination process

SA messages are triggered by an RP, provided that MSDP is configured, when a new source becomes active within a local PIM-SM domain. A local source is one 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 that register with it in its PIM-SM domain.

> **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 **showipmroute** 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, the RP creates (S, G) state. The RP detects new sources when it receives a register message or when the first (S, G) packet arrives from a directly connected source. The RP encapsulates the initial multicast packet, which is sent by the source, in the initial SA message. This packet may be included in the register message or received from a directly connected source.

SA message receipt

Describes how SA messages are accepted and processed by MSDP peers to prevent routing loops.

SA message receipt is an MSDP mechanism with three main functions: It accepts SA messages only from the MSDP RPF peer that is in the best path back toward the originator, ignores the same SA message arriving from other MSDP peers to prevent SA loops, and uses (M)BGP routing data as the best approximation of the MSDP topology for the SA RPF check mechanism.

RPF peer selection requirements

Deterministically selecting the MSDP RPF peer for an arriving SA message requires knowledge of the MSDP topology. However, MSDP does not distribute topology information in the form of routing updates. MSDP infers this information by using (M)BGP routing data as the best approximation of the MSDP topology for the SA RPF check mechanism.

An MSDP topology must follow the same general topology as the BGP peer topology. With a few exceptions, such as default MSDP peers and MSDP peers in MSDP mesh groups, most MSDP peers should also be (M)BGP peers.

RPF check rules for SA messages

Describes how RPF check rules are applied to SA messages based on BGP peerings between MSDP peers.

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

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

RPF checks are not performed in 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 software determines RPF check rules

Describes the logic software uses to determine which reverse path forwarding rule to apply to RPF checks based on BGP neighbor matching.

Summary

Software finds the (M)BGP neighbor that has the same IP address as the sending MSDP peer.

Workflow

These stages describe how software determines the RPF check rule to apply

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

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

How rule 1 of RPF checking of SA messages in MSDP works

Describes how Rule 1 of RPF checking in MSDP is applied when the sending MSDP peer is also an i(M)BGP peer.

Summary

Rule 1 of RPF checking in MSDP is applied when the sending MSDP peer is also an i(M)BGP peer.

Workflow

Rule 1 RPF checking involves these stages:

1. The peer searches for the best path to the RP that originated the SA message in the routing databases.
   - First searches the BGP Multicast Routing Information Base (MRIB)
   - If path not found in MRIB, searches the Unicast Routing Information Base (URIB)

   If a path is still not found after searching both databases, the RPF check fails.
2. If the path search succeeds, the peer determines the BGP neighbor address for the best path.

   The BGP neighbor address is the address of the BGP neighbor that sent the peer the path in BGP update messages.

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

   The BGP neighbor address is not necessarily the same as the BGP ID of the peer that sent the peer the path.
3. The peer compares the IP address of the sending MSDP peer with the BGP neighbor address.

   If the IP address of the sending MSDP peer is the same as the BGP neighbor address, then the RPF check succeeds; otherwise it fails.

Implications of rule 1 of RPF checking on MSDP

Explains how Rule 1 of RPF checking affects MSDP topology configuration and peer address requirements.

The implications of Rule 1 of RPF checking on MSDP are configuration requirements that

- require the MSDP topology to mirror the (M)BGP topology
- require that an MSDP peer connection is configured wherever there is an i(M)BGP peer connection between two devices, and
- require the IP address of the far-end MSDP peer connection to be the same as that of the far-end i(M)BGP peer connection.

Address matching requirements

The addresses must be the same because the BGP topology between i(M)BGP peers inside an autonomous system is not described by the AS path. If i(M)BGP peers always updated the next-hop address in the path when sending an update to another i(M)BGP peer, then the peer could rely on the next-hop address to describe the i(M)BGP topology and, by extension, the MSDP topology. However, because i(M)BGP peers by default do not update the next-hop address, the peer cannot rely on the next-hop address to describe the (M)BGP topology or MSDP topology. Instead, the i(M)BGP peer uses the address of the i(M)BGP peer that sent the path to describe the i(M)BGP topology or MSDP topology within the autonomous system.

Tip

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

How rule 2 of RPF checking of SA messages in MSDP works

Describes how Rule 2 of RPF checking in MSDP functions when the sending MSDP peer is also an e(M)BGP peer.

Summary

Rule 2 of RPF checking in MSDP is applied when the sending MSDP peer is also an e(M)BGP peer.

Workflow

These stages describe how Rule 2 of RPF checking proceeds:

1. The peer searches the BGP MRIB for the best path to the RP that originated the SA message. If no path exists in the MRIB, the peer searches the URIB. If the URIB also lacks a path, the RPF check fails.
2. If the peer finds a best path in the previous search, it examines the path. When the first autonomous system in the best path to the RP matches the autonomous system of the e(M)BGP peer (also the sending MSDP peer), the RPF check succeeds; if not, the RPF check fails.

Implications of rule 2 of RPF checking on MSDP

Describes the topology mirroring requirements between MSDP and BGP protocols.

The MSDP topology must mirror the (M)BGP topology.

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

How rule 3 of RPF checking of SA messages in MSDP works

Describes the process for Rule 3 of RPF checking when the sending MSDP peer is not a (M)BGP peer at all.

Summary

Rule 3 of RPF checking is applied when the sending MSDP peer is not a (M)BGP peer at all.

Workflow

Rule 3 of RPF checking involves these stages:

1. The peer searches for the best path to the RP that originated the SA message.
   - The peer first searches the BGP MRIB for the best path to the RP
   - If no path is found in the MRIB, the peer searches the URIB
   - If no path is found in either location, the RPF check fails
2. If the RP path search succeeds, the peer searches for the best path to the MSDP peer that sent the SA message.
   - The peer searches the BGP MRIB for the best path to the MSDP peer
   - If no path is found in the MRIB, the peer searches the URIB
   - If no path is found in either location, 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. The peer compares autonomous systems to determine RPF check success or failure.
   - If the first autonomous system in the best path to the RP matches the autonomous system of the sending MSDP peer, the RPF check succeeds
   - If the autonomous systems do not match, the RPF check fails

SA message processing

Describes how MSDP peers process SA messages to manage multicast source discovery and distribution.

Summary

These stages describe how an MSDP peer processes an SA message:

Workflow

These stages describe how an MSDP peer processes an SA message:

1. The peer uses the group address G of the (S, G) pair in the SA message to locate the associated (, G) entry in the mroute table. If the entry is found and its outgoing interface list is not empty, the device confirms that active receivers exist in the PIM-SM domain for the advertised source.
2. The MSDP peer then creates an (S, G) entry for the advertised source.
3. If the (S, G) entry did not exist, the MSDP peer immediately triggers an (S, G) join toward the source to establish the source tree.
4. The peer then floods the SA message to all other MSDP peers with the exception of:
   - The MSDP peer from which the SA message was received.

- Peers in the same MSDP mesh group as this device (when the peer is a member of a mesh group).

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

MSDP peers

Describes the neighbor relationships established between MSDP peers and their connection characteristics.

An MSDP peer is a network device that

- establishes neighbor relationships with other MSDP peers using TCP port 639
- maintains active or passive connection roles based on IP address comparison, and
- sends keepalive messages every 60 seconds to maintain session state.

MSDP peer connection behavior

Like BGP, MSDP establishes neighbor relationships with other MSDP peers. The connection process follows these characteristics:

- The lower IP address peer takes the active role of opening the TCP connection
- The higher IP address peer waits in LISTEN state for the other to make the connection
- The arrival of data performs the same function as the keepalive message and keeps the session from timing out
- If no keepalive messages or data is received for 75 seconds, the TCP connection is reset

MSDP MD5 password authentication

Describes an MSDP security enhancement that uses MD5 signatures to protect TCP connections between MSDP peers.

MSDP MD5 password authentication is a security enhancement that

- supports Message Digest 5 (MD5) signature protection on a TCP connection between two MSDP peers
- provides added security by protecting MSDP against the threat of spoofed TCP segments being introduced into the TCP connection stream, and
- enhances the security of MSDP peer communications through cryptographic authentication.

MSDP MD5 password authentication

Describes how MSDP MD5 password authentication works to verify TCP segments between MSDP peers.

MSDP MD5 password authentication is a security feature that

- verifies each segment sent on the TCP connection between MSDP peers,
- is developed in accordance with RFC 2385, and
- uses the **ip MSDP password peer** command to enable MD5 authentication for TCP connections between two MSDP peers.

Authentication process

When MD5 authentication is enabled between two MSDP peers, each segment sent on the TCP connection between the peers is verified. MD5 authentication must be configured with the same password on both MSDP peers. If the passwords do not match, the connection will fail. When you configure MD5 authentication, the Cisco IOS software generates and verifies the MD5 digest for every segment sent on the TCP connection.

Benefits of MSDP MD5 password authentication

Describes the security advantages of implementing MD5 password authentication for MSDP connections.

Benefits of MSDP MD5 password authentication are security advantages that

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

SA message limits

Describes the mechanism used to limit the number of SA messages that a device can accept from specified MSDP peers.

An SA message limit is a security mechanism that limits the overall number of SA messages that a device can accept from specified MSDP peers. It maintains a per-peer count of SA messages stored in the SA cache. If the configured SA message limit for a peer has been reached, new messages from that peer are ignored.

Configuration and protection features

The **ip msdp sa-limit** command is used to configure SA message limits and was introduced as a means to protect an MSDP-enabled device from denial of service (DoS) attacks.

Configuration recommendations include:

- Configure SA message limits for all MSDP peerings on the device
- Configure an appropriately low SA limit on peerings with a stub MSDP region (for example, a peer that may have some further downstream peers but that will not act as a transit for SA messages across the rest of the Internet)
- Configure a high SA limit for all MSDP peerings that act as transits for SA messages across the Internet

MSDP keepalive and hold-time intervals

Describes the intervals used to maintain MSDP peer connections through keepalive messages and connection monitoring.

MSDP keepalive and hold-time intervals are connection maintenance parameters that

- control the frequency of keepalive messages sent between MSDP peers
- determine how long peers wait for keepalive messages before declaring connections down, and
- ensure stable MSDP peering sessions through configurable timer values.

Timer configuration and behavior

The **ip msdp keepalive** command adjusts the interval at which an MSDP peer sends keepalive messages and the interval at which the MSDP peer waits for keepalive messages from other peers before declaring them down.

Once an MSDP peering session is established, each side of the connection sends a keepalive message and sets a keepalive timer. If the keepalive timer expires, the local MSDP peer sends a keepalive message and restarts its keepalive timer. This interval is called the keepalive interval. The *keepalive-interval* argument adjusts the interval for which keepalive messages are sent. When the peer comes up, the keepalive timer is set to the value specified for the *keepalive-interval* argument when the peer comes up. The keepalive timer is reset to the value of the *keepalive-interval* argument whenever

an MSDP keepalive message is sent to the peer and reset when the timer expires. The keepalive timer is deleted when an MSDP peering session is closed. By default, the keepalive timer is set to 60 seconds.

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

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

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

MSDP connection-retry interval

Describes the interval at which MSDP peers wait after peering sessions are reset before attempting to reestablish the peering sessions.

An MSDP connection-retry interval is a timing parameter that controls the waiting period after MSDP peering sessions are reset before attempting to reestablish sessions. This interval applies to all MSDP peering sessions on the device. The default value is 30 seconds, but it can be adjusted as needed.

Default MSDP peers

Describes MSDP peers that accept SA messages without performing peer-RPF checks in autonomous systems that do not support BGP.

A default MSDP peer is an MSDP peer configuration that accepts all SA messages without performing the peer-RPF check, eliminates the dependency on BGP for MSDP operations in stub or nontransit autonomous systems, and must be a previously configured MSDP peer.

MSDP and BGP peer relationships

In most network scenarios, an Multicast Source Discovery Protocol (MSDP) peer also functions as a Border Gateway Protocol (BGP) peer.

Stub Autonomous Systems and BGP

If an Autonomous System (AS) is a stub or nontransit AS—and particularly if it is not multihomed—running BGP to its transit AS is rarely necessary. Typically, the following routing setup is sufficient:

- A static default route at the stub AS pointing to the transit AS.
- A static route at the transit AS pointing to the stub prefixes.

BGP dependency for RPF checks

If a stub AS operates as a multicast domain and its Rendezvous Point (RP) must peer with an RP in a neighboring domain, MSDP normally relies on the BGP next-hop database to perform Reverse Path Forwarding (RPF) checks.

However, you can bypass this dependency. By defining a default peer, you can configure the system to accept all Source Active (SA) messages from that peer without performing the standard peer-RPF check.

MSDP without BGP

If your switch does not support BGP, you cannot use the standard **ip msdp peer** global configuration command to set up an MSDP peer.

Instead, you must define a default MSDP peer using the **ip msdp default-peer** global configuration command. If you configure only a single MSDP peer, the switch bypasses the RPF check and automatically accepts all SA messages from that peer.

Redundancy in Stub Autonomous Systems

A stub AS may want to establish MSDP peerings with more than one RP to ensure redundancy. Because there is no active RPF check mechanism when using default peers, SA messages cannot simply be accepted from multiple default peers simultaneously.

Instead, the system handles redundancy through the following mechanism:

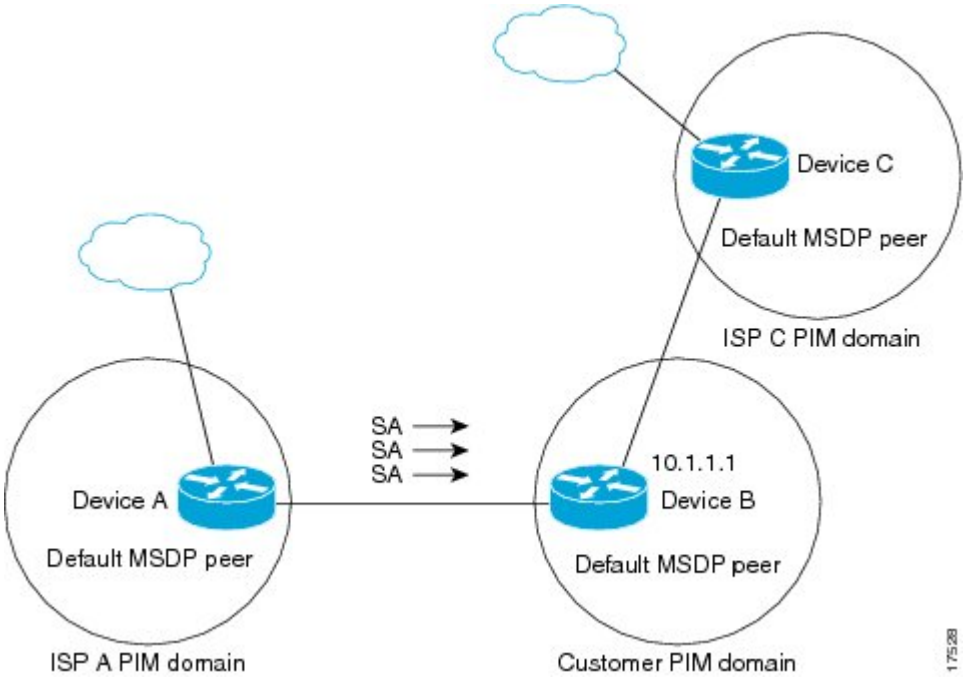
1. SA messages are accepted from only one active peer at a time.
2. If the primary active peer fails, the system switches to accept SA messages from the backup peer.

> **Note**
>
> This redundancy model operates under the assumption that both default peers are transmitting identical SA messages.

Network setup

Consider a scenario where a customer owns **Device B** and is connected to the Internet via two different Internet Service Providers (ISPs), which own **Device A** and **Device C** respectively. BGP is not running between the customer and the ISPs.

Figure 2: Default MSDP peer scenario



Traffic flow and failover

Device B establishes default MSDP peer relationships to learn about multicast sources within ISP or other external domains:

- **Advertising**: Device B advertises SA messages to both Device A and Device C.
- **Receiving**: Device B accepts incoming SA messages from either Device A only or Device C only.
- **Failover Logic**: If Device A is listed first in the configuration, it remains the active peer as long as it is up and running. Device B will only fall back to accept SA messages from Device C if Device A becomes unreachable.

Advanced filtering with prefix lists

To refine traffic behavior and allow simultaneous usage of multiple peers, ISPs and customers can implement prefix lists.

ISP-side filtering

The ISP will likely implement a prefix list to strictly define and restrict which multicast prefixes it will accept from the customer device.

Customer-side filtering

The customer can define multiple default peers, assigning one or more specific prefixes to each peer using a prefix list.

Configuration Type	Multiple Active Peers?	Behavior / Failover Mechanism
Without Prefix Lists	**No**	Only the first configured peer is active. The system fails over to the second peer only if the first peer or its connectivity goes down.
With Prefix Lists	**Yes**	A peer acts as the default peer *only* for its explicitly permitted prefixes. This allows multiple default peers to be active simultaneously for different prefix ranges.

MSDP mesh groups

Describes a group of MSDP speakers that reduces SA message flooding through fully meshed connectivity.

An MSDP mesh group is a group of MSDP speakers that

- have fully meshed MSDP connectivity between one another,
- require each MSDP peer to have an MSDP peering relationship (MSDP connection) to every other MSDP peer in the group, and
- reduce SA message flooding when configured between MSDP peers.

MSDP mesh group operation

When an MSDP peer in the group receives an SA message from another MSDP peer in the group, it assumes that this SA message was sent to all the other MSDP peers in the group. As a result, it is not necessary for the receiving MSDP peer to flood the SA message to the other MSDP peers in the group.

Benefits of MSDP mesh groups

Describes the advantages of implementing MSDP mesh groups for SA flooding optimization and traffic reduction.

MSDP mesh groups provide these specific operational improvements:

- SA flooding optimization: MSDP mesh groups are particularly useful for optimizing SA flooding when two or more peers are in a group.
- Traffic reduction: SA messages are not flooded to other mesh group peers, reducing the amount of SA traffic across the Internet.
- Simplified message processing: SA messages are always accepted from mesh group peers, eliminating RPF checks on arriving SA messages.

SA origination filters

Describes MSDP filters that control which local sources are advertised in SA messages from an RP to MSDP peers.

An SA origination filter is an MSDP configuration mechanism that

- controls what sources are advertised in SA messages from an RP
- prevents unwanted local sources (such AS those with private addresses) from being advertised to MSDP peers, and
- allows selective filtering based on extended access lists, AS-path access lists, or route maps.

SA origination filter control methods

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

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

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

Outgoing filter lists in MSDP

Describes how to control SA message forwarding to MSDP peers using outgoing filter lists.

An outgoing filter list is an MSDP filtering mechanism that

- prevents SA messages from being forwarded to MSDP peers
- applies to all SA messages, whether locally originated or received from another MSDP peer, and
- controls which SA messages a device forwards to specific peers based on various criteria.

Filtering capabilities

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

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

By default, an MSDP-enabled device forwards all SA messages it receives to all of its MSDP peers. Outgoing filter lists apply to all SA messages, whether locally originated or received from another MSDP peer, whereas SA origination filters apply only to locally originated SA messages. For more information about enabling a filter for MSDP SA messages originated by the local device, see the Control SA messages originated by an RP for local sources on page 41.

> ⚠ **Caution**
>
> Arbitrary filtering of SA messages can result in downstream MSDP peers being starved of SA messages for legitimate active sources. Therefore, use caution when using these sorts of filters. Normally, outgoing filter lists are used only to reject undesirable sources, such as sources using private addresses.

Incoming filter lists in MSDP

Explains how MSDP-enabled devices can control source information received from MSDP peers using incoming filter lists.

An incoming filter list in MSDP is a filtering mechanism that

- controls the source information that a device receives from its MSDP peers
- filters SA messages based on various criteria including (S, G) pairs, route maps, or RP addresses, and
- prevents devices from receiving all SA messages by default from MSDP peers.

Filtering capabilities

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

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

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

> ⚠ **Caution**
>
> Arbitrary filtering of SA messages can result in downstream MSDP peers not receiving SA messages for legitimate active sources. Therefore, exercise care when using these types of filters. Normally, incoming filter lists are used only to reject undesirable sources, such as sources using private addresses.

TTL thresholds in MSDP

Describes how TTL thresholds limit multicast packet hops in MSDP SA messages to prevent TTL violations.

The time-to-live (TTL) thresholds in MSDP limit the number of multicast packet hops by specifying a TTL value for data-encapsulated SA messages sent to specified MSDP peers with the **ip multicast ttl-threshold** command. They prevent multicast packets whose TTL is less than the specified threshold from being encapsulated in SA messages. This mechanism helps to solve TTL violation problems caused by the encapsulation of source multicast packets in unicast SA messages.

TTL threshold behavior and implementation

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.

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

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

SA request messages

Describes messages sent by noncaching devices to reduce join latency for multicast groups.

An SA request message is an MSDP message sent by noncaching devices to one or more specified MSDP peers. It reduces join latency for noncaching peers by requesting cached source information for specific multicast groups. It triggers SA response messages from caching peers that contain the requested source information.

SA request message behavior

When a host requests a join to a particular group, the noncaching RP sends an SA request message to its caching peers. If a peer has cached source information for the group in question, it sends the information to the requesting RP with an SA response message. The requesting RP uses the information in the SA response but does not forward the message to any other peers. If a noncaching RP receives an SA request, it sends an error message back to the requestor.

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

SA request filters

Describes MSDP peer filtering mechanisms for controlling outgoing SA request message processing.

An SA request filter is an MSDP control mechanism that controls which outgoing SA request messages a device will honor from specified MSDP peers. It manages cached source information sent to requesting MSDP peers in SA response messages. It also provides granular filtering options for SA request message processing.

SA request filter functionality

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. An SA request filter controls the outgoing SA requests that the device will honor from MSDP peers in these ways:

- You can filter all SA request messages from a specified peer by configuring the device to ignore all SA requests from the specified MSDP peer.
- You can filter a subset of SA request messages from a specified peer based on groups defined in a standard access list. Configure 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 are ignored.

How to use MSDP to interconnect multiple PIM-SM domains

Describes how to use Multicast Source Discovery Protocol (MSDP) to interconnect multiple Protocol Independent Multicast Sparse Mode (PIM-SM) domains, with one required configuration task and additional optional tasks.

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

Configure an MSDP peer

Configure an MSDP peer to enable MSDP functionality and establish communication between PIM-SM domains.

Configure an MSDP peer to enable MSDP functionality and establish communication between PIM-SM domains for multicast source discovery.

> **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.
- All MSDP peers must be configured to run BGP before being configured for MSDP, except for a single MSDP peer, the default MSDP peer, and MSDP mesh group scenarios.

Follow these steps to configure an MSDP peer:

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **ip msdp peer** {*peer-name*|*peer-address*} [*connect-source type number*] [**remote-as** *as-number*] command to enable MSDP and configure an MSDP peer as specified by the DNS name or IP address.

   Example

   ```
   Device(config)# ip msdp peer 192.168.1.2 connect-source loopback0
   ```

   > **Note**
   >
   > The device that is selected to be configured as an MSDP peer is also usually a BGP neighbor. If the device is not a BGP neighbor, refer to Configure a default MSDP peer on page 39 section or the Configure an MSDP mesh group on page 40 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.

3. (Optional) Use the **ip msdp description** {*peer-name*|*peer-address*}*text* command to configure a description for a specified peer to make it easier to identify in a configuration or in show command output.

   Example

   ```
   Device(config)# ip msdp description 192.168.1.2 router at customer a
   ```

4. Use the **end** command to return to privileged EXEC mode.

Example

```
Device(config)# end
```

Shut down an MSDP peer

Configure an MSDP peer shutdown to temporarily disable the peer connection without losing the peer configuration.

Shutting down an MSDP peer allows you to configure multiple peers before activating them or temporarily disable a peer session without losing the configuration.

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

Note

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

Follow these steps to shut down an MSDP peer:

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

Example

```
Device# configure terminal
```

2. Use the **ip msdp shutdown** {*peer-name* | *peer-address*} command to administratively shut down the specified MSDP peer.

Example

```
Device(config)# ip msdp shutdown 192.168.1.3
```

Repeat this step to shut down additional MSDP peers.

3. Use the **end** command to return to privileged EXEC mode.

Example

```
Device(config)# end
```

Configure MSDP MD5 password authentication between MSDP peers

Configure MD5 password authentication to secure connections between MSDP peers.

Configure MD5 password authentication to provide secure authentication between MSDP peers and protect the TCP connection.

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

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **ip msdp password peer** {*peer-name* | *peer-address*} [*encryption-type*] *string* command to enable MD5 password encryption for a TCP connection between two MSDP peers.

   Example

   ```
   Device(config)# ip msdp password peer 10.32.43.144 0 test
   ```

   Note

   MD5 authentication must be configured with the same password on both MSDP peers; otherwise, the devices do not establish a connection.

   - If you configure or change the password or key that 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.

3. Use the **exit** command to return to privileged EXEC mode.

   Example

   ```
   Device(config)# exit
   ```

4. (Optional) Use the **show ip msdp peer** [**peer-address** | **peer-name**] command to display detailed information about MSDP peers.

   Example

   ```
   Device# show ip msdp peer
   ```

> **Note**
>
> Use this command to verify whether MD5 password authentication is enabled on an MSDP peer.

Troubleshooting tips for MSDP password authentication

Consider these troubleshooting approaches when MSDP peers fail to establish sessions due to password authentication issues.

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

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

IP address]:179
```

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

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

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

Prevent DoS attacks by limiting the number of SA messages allowed in the SA cache from specified MSDP peers

Configure SA message limits for MSDP peers to protect against distributed denial-of-service attacks.

Limiting the total number of SA messages a device accepts from specified MSDP peers protects it from distributed denial-of-service (DoS) attacks.

Perform this 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

1. Use the **configure terminal** command to enter global configuration mode.

Example

```
Device# configure terminal
```

2. Use the **ip msdp sa-limit** {*peer-address* | *peer-name*} *SA-limit* command to limit the number of SA messages allowed in the SA cache from the specified MSDP peer.

Example

```
Device(config)# ip msdp sa-limit 192.168.10.1 100
```

3. Configure SA limits for each additional MSDP peer as needed.
4. Use the **exit** command to return to privileged EXEC mode.

Example

```
Device(config)# exit
```

5. (Optional) Use the **show ip msdp count** [*as-number*] 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.

Example

```
Device# show ip msdp count
```

6. (Optional) Use the **show ip msdp peer** [*peer-address* | *peer-name*] command to display detailed information about MSDP peers.

Example

```
Device# show ip msdp peer
```

> **Note**
>
> The output of this command displays the number of SA messages received from MSDP peers that are stored in the cache.

7. (Optional) Use the **show ip msdp summary** command to display MSDP peer status.

Example

```
Device# show ip msdp summary
```

> **Note**
>
> The output of this command displays a per-peer "SA Count" field that displays the number of SAs stored in the cache.

Adjust the MSDP keepalive and hold-time intervals

Configure the interval at which an MSDP peer sends keepalive messages and waits for keepalive messages from other peers before declaring them down.

Adjust the interval for sending keepalive messages and the interval for waiting for keepalive messages from other peers before declaring them down. In network environments with redundant MSDP peers, decreasing the hold-time interval can speed up reconvergence if a peer fails.

By default, an MSDP peer may take up to 75 seconds to detect that a peering session with another MSDP peer has gone down. This task can help improve network convergence times.

> **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

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **ip msdp keepalive** {*peer-address* | *peer-name*} *keepalive-interval hold-time-interval* command to configure 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.

   Example

   ```
   Device(config)# ip msdp keepalive 10.1.1.3 40 55
   ```

   Repeat this step to adjust the keepalive message interval for additional MSDP peers.

3. Use the **exit** command to return to privileged EXEC mode.

Example

```
Device(config)# exit
```

Adjust the MSDP connection-retry interval

Configure the interval at which MSDP peers wait after peering sessions are reset before attempting to reestablish connections.

Adjust the MSDP connection-retry interval to optimize network recovery in environments requiring fast SA message recovery, such as trading floor networks.

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

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **ip msdp timer** *connection-retry-interval* command to configure the interval at which MSDP peers will wait after peering sessions are reset before attempting to reestablish the peering sessions.

   Example

   ```
   Device(config)# ip msdp timer 45
   ```

3. Use the **exit** command to return to privileged EXEC mode.

   Example

   ```
   Device(config)# exit
   ```

Configure a default MSDP peer

Configure a default peer from which to accept all MSDP SA messages.

This task configures a default MSDP peer to accept all MSDP SA messages from a specific peer.

Perform this task to configure a default MSDP peer.

Before you begin

Before you configure a default MSDP peer, you must configure an MSDP peer. The default peer must already be configured as an MSDP peer.

Follow these steps to configure a default MSDP peer:

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **ip msdp default-peer** {*peer-address* | *peer-name*} [**prefix-list** *list*] command to configure a default peer from which to accept all MSDP SA messages.

   Example

   ```
   Device(config)# ip msdp default-peer 192.168.1.3
   ```

3. Use the **exit** command to return to privileged EXEC mode.

   Example

   ```
   Device(config)# exit
   ```

Configure an MSDP mesh group

Configure an MSDP mesh group to enable efficient multicast source discovery protocol communication between peering routers.

Configure an MSDP mesh group to create a logical grouping of MSDP peers that share multicast source information efficiently.

Perform this task to configure an MSDP mesh group.

> **Note**
>
> You can configure multiple mesh groups per device.

Before you begin

Follow these steps to configure an MSDP mesh group:

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **ip msdp mesh-group** *mesh-name* {*peer-address* | *peer-name*} command to configure an MSDP mesh group and indicate that an MSDP peer belongs to that mesh group.

Example

```
Device(config)# ip msdp mesh-group peermesh
```

Note

Each MSDP peer in a mesh group must connect to every other MSDP peer in the same group. Configure each MSDP peer on every device 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.

Repeat this step to add additional MSDP peers to the mesh group.

3. Use the **exit** command to return to privileged EXEC mode.

Example

```
Device(config)# exit
```

Control SA messages originated by an RP for local sources

Configure a filter to restrict which registered sources are advertised in SA messages by an RP.

This task enables a filter for MSDP SA messages originated by the local device to control which registered sources are advertised.

Enable a filter to restrict which registered sources are advertised in SA messages originated by an RP.

Note

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

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

Example

```
Device# configure terminal
```

2. Use the **ip msdp redistribute** [**list** *access-list*] [**asn** *as-access-list*] [**route-map** *map-name*] command to enable a filter for MSDP SA messages originated by the local device.

Example

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

> **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.

3. Use the **exit** command to return to privileged EXEC mode.

Example

```
Device(config)# exit
```

Control the forwarding of SA messages to MSDP peers using outgoing filter lists

Configure outgoing filter lists to control the forwarding of SA messages to MSDP peers.

Control the forwarding of SA messages to MSDP peers by configuring outgoing filter lists.

This task allows you 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

1. Use the **configure terminal** command to enter global configuration mode.

Example

```
Device# configure terminal
```

2. Use the **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*] command to enable a filter for outgoing MSDP messages.

Example

```
Device(config)# ip msdp sa-filter out 192.168.1.5 peerone
```

Repeat this step to configure outgoing filter lists for additional MSDP peers.

3. Use the **exit** command to return to privileged EXEC mode.

Example

```
Device(config)# exit
```

Control the receipt of SA messages from MSDP peers using incoming filter lists

Configure incoming filter lists to control the receipt of SA messages from MSDP peers.

Control the receipt of incoming SA messages from MSDP peers by configuring filter lists.

Perform this 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

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **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*] command to enable a filter for incoming MSDP SA messages.

   Example

   ```
   Device(config)# ip msdp sa-filter in 192.168.1.3
   ```

   Repeat this step to configure incoming filter lists for additional MSDP peers.

3. Use the **exit** command to return to privileged EXEC mode.

   Example

   ```
   Device(config)# exit
   ```

Use TTL thresholds to limit the multicast data sent in SA messages

Configure a time to live (TTL) threshold to limit the multicast data sent in SA messages.

This task establishes a time to live (TTL) threshold to limit the multicast data sent in SA messages.

This task helps control MSDP message behavior by setting TTL values for multicast data packets.

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **ip msdp ttl-threshold** {*peer-address* | *peer-name*} *TTL-value* command to set a TTL value for MSDP messages originated by the local device.

   Example

   ```
   Device(config)# ip msdp ttl-threshold 192.168.1.5 8
   ```

   - By default, multicast data packets in SA messages are sent to an MSDP peer, provided the TTL value of the packet is greater than 0, which is standard TTL behavior.

3. Use the **exit** command to return to privileged EXEC mode.

   Example

   ```
   Device(config)# exit
   ```

Request source information from MSDP peers

Configure a device to request source information from MSDP peers to enable SA message retrieval from caching peers.

Enable a device to request source information from MSDP peers for SA message retrieval.

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

> **Note**
>
> SA caching is enabled by default in earlier Cisco software releases, and it cannot be explicitly enabled or disabled. Therefore, you seldom need to perform this task.

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **ip msdp sa-request** {*peer-address* | *peer-name*} command to specify that the device send SA request messages to the specified MSDP peer.

Example

```
Device(config)# ip msdp sa-request 192.168.10.1
```

Repeat this step to specify that the device send SA request messages to additional MSDP caching peers.

3. Use the **exit** command to return to privileged EXEC mode.

Example

```
Device(config)# exit
```

Control the response to outgoing SA request messages from MSDP peers using SA request filters

Configure SA request filters to control the outgoing SA request messages that the device will honor from MSDP peers.

Control the outgoing SA request messages that the device will honor from MSDP peers by configuring SA request filters. This task allows you to filter outgoing SA request messages from MSDP peers for better network control.

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

Example

```
Device# configure terminal
```

2. Use the **ip msdp filter-sa-request** {*peer-address* | *peer-name*} [**list** *access-list*] command to enable a filter for outgoing SA request messages.

Example

```
Device(config)# ip msdp filter sa-request 172.31.2.2 list 1
```

> **Note**
>
> Only one SA request filter can be configured per MSDP peer.

Repeat this step to configure SA request filters for additional MSDP peers.

3. Use the **exit** command to return to privileged EXEC mode.

Example

```
Device(config)# exit
```

Configure an originating address other than the RP address

Configure an MSDP speaker to use the IP address of its interface as the RP address in SA messages instead of the default RP address.

This task allows 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 change the originator ID for any one of the following reasons:

- If you configure multiple devices in an MSDP mesh group for Anycast RP.
- If a device borders both 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 32.

Follow these steps to configure an originating address other than the RP address:

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **ip msdp originator-id** *type number* command to configure the RP address in SA messages to be the address of the originating device's interface.

   Example

   ```
   Device(config)# ip msdp originator-id ethernet 1
   ```

3. Use the **exit** command to return to privileged EXEC mode.

   Example

   ```
   Device(config)# exit
   ```

Monitor MSDP

Monitor MSDP SA messages, peers, state, and peer status to assess the health and connectivity of MSDP.

This task enables monitoring of MSDP (Multicast Source Discovery Protocol) activity to track SA messages, peer status, and state information for network troubleshooting and performance analysis.

This monitoring task allows you to observe MSDP SA messages, peers, state, and peer status to facilitate network analysis and troubleshooting.

Procedure

1. Use the **debug ip msdp** [*peer-address* | *peer-name*] [**detail**] [**routes**] command to debug MSDP activity.

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

   This example output is 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
```

2. Use the **debug ip msdp resets** command to debug MSDP peer reset reasons.

   Example

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

3. Use the **show ip msdp count** [*as-number*] 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 example output is 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
```

4. Use the **show ip msdp peer** [*peer-address* | *peer-name*] 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 example output is 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
```

5. Use the **show ip msdp sa-cache** [*group-address* | *source-address* | *group-name* | *source-name*] [*as-number*] command to display the (S, G) state learned from MSDP peers.

   This example output is 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
```

6. Use the **show ip msdp summary** command to display MSDP peer status.

   This example output is 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 entries

Clear MSDP connections, statistics, and SA cache entries to reset message counters and remove cached data.

This task clears MSDP connections, statistics, and SA cache entries to reset all MSDP message counters and remove cached multicast source-active information.

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

Procedure

1. Use the **clear ip msdp peer** [*peer-address* | *peer-name*] command to clear the TCP connection to the specified MSDP peer. This also resets all MSDP message counters.

   Example

```
Device# clear ip msdp peer
```

2. Use the **clear ip msdp statistics** [*peer-address* | peer-name] command to clear the statistics counters for the specified MSDP peer and reset all MSDP message counters.

   Example

```
Device# clear ip msdp statistics
```

3. Use the **clear ip msdp sa-cache** [*group-address*] command to clear SA cache entries.

   Example

```
Device# clear ip msdp sa-cache
```

- 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

Configure SNMP monitoring for Multicast Source Discovery Protocol (MSDP) to enable network management and troubleshooting capabilities.

Enable Simple Network Management Protocol (SNMP) monitoring of MSDP to provide network management capabilities and allow monitoring of MSDP operations through SNMP-based network management systems.

Perform this task to enable SNMP monitoring of MSDP.

Before you begin

- SNMP and MSDP are configured on your devices.
- Each PIM-SM domain should include a device 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.

Follow these steps to enable SNMP monitoring of MSDP:

Procedure

1. Use the **snmp-server enable traps msdp** command to enable the sending of MSDP notifications for use with SNMP.

   Example

   ```
   Device# snmp-server enable traps msdp
   ```

   Note

   The **SNMP-server enable traps MSDP** command enables both traps and informs.

2. Use the **snmp-server host** *host* [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth**| **priv** | **noauth**]}] *community-string* [**udp-port** *port-number*] **msdp** command to specify the recipient (host) for MSDP traps or informs.

Example

```
Device# snmp-server host examplehost msdp
```

3. Use the **exit** command to return to privileged EXEC mode.

Example

```
Device(config)# exit
```

Tip: compare MSDP MIB notifications with software output

Consider comparing MSDP MIB notification results to software command output and SNMP operations for effective troubleshooting.

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

Configuration examples for using MSDP to interconnect multiple PIM-SM domains

Describes configuration examples that demonstrate how to use Multicast Source Discovery Protocol (MSDP) to interconnect multiple Protocol Independent Multicast Sparse Mode (PIM-SM) domains.

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

Example: configure an MSDP peer

Provides configuration examples for establishing MSDP peering connections between three MSDP peers.

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
!
```

Configure MSDP MD5 password authentication

Provides an example of how to enable MD5 password authentication for TCP connections between two MSDP peers.

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
!
```

Example: configure a default MSDP peer

Provides a scenario and configuration example for using default MSDP peers in a network topology with multiple ISPs.

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 another that owns Device C. They are not running (M)BGP between them. Therefore, for the customer to learn about sources in the ISP domain or 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 from only one peer: either Device A or Device C. If Device A is the first default peer configured and is up and running, Device B uses it. If Device A is unavailable, Device B accepts SA messages from Device C.

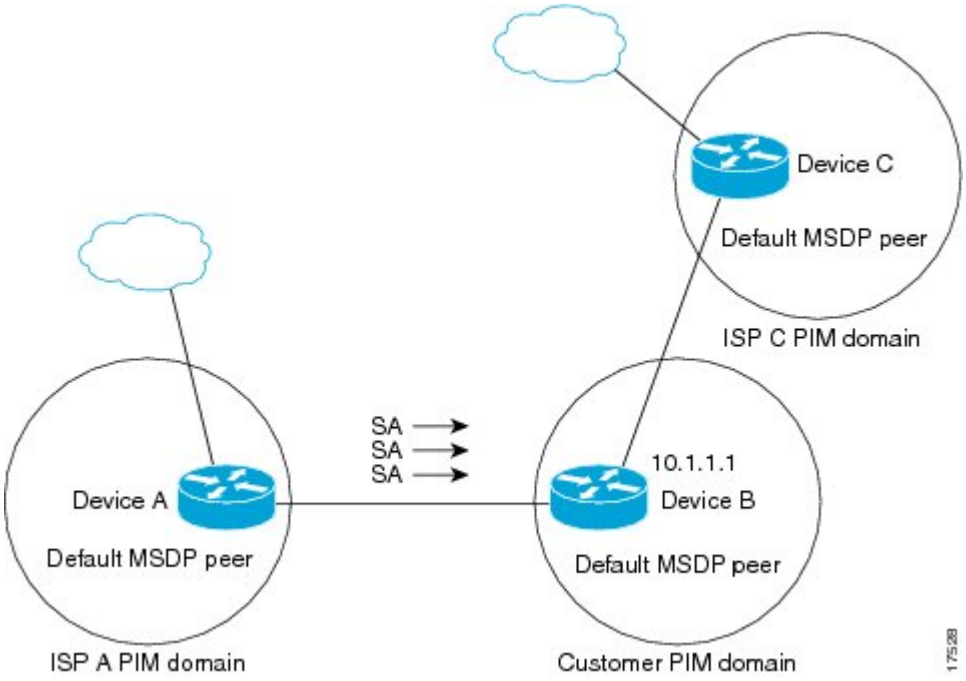
The ISP likely uses a prefix list to define which prefixes it will accept from the customer device. The customer defines multiple default peers, each associated with one or more prefixes.

The customer has two ISPs to use. The customer defines both ISPs as default peers. If the first default peer is active, it will be the default peer. The customer accepts all SA messages received from it.

Note

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

Figure 3: Default MSDP peer scenario



Device B advertises SAs to Device A and Device C, but accepts SA messages from only one of them. 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

Provides configuration examples for three devices configured as fully meshed members of an MSDP mesh group.

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
```

Troubleshooting tips

Lists commands for gathering diagnostic data and debugging multicast configurations.

You can gather diagnostic data on the Multicast configurations with the **show tech-support ipmulticast** command. You can also verify the configurations and the IGMP snooping information with the **show tech-support igmp-snopping** command.

For further troubleshooting and debugging, you can use these commands:

Debug Commands	Purpose
debug mrouting	Enables debugging on Multicast Routing activity
debug IP multicast	Enables debugging on IP Multicast activity
debug mpacket	Enables debugging on Multicast packets
debug IP IGMP	Enables debugging IP IGMP protocol activity
debug MFIB	Enables debugging MFIB routing and forwarding

3 Configuring PIM

Topics:

- Prerequisites for PIM
- Restrictions for PIM
- Restrictions for configuring Auto-RP and BSR
- PIMv1 and PIMv2 interoperability
- Information about PIM
- How to configure PIM
- Verifying PIM operations
- Monitoring and troubleshooting PIM
- Configuration Examples for PIM

Prerequisites for PIM

Consider these guidelines before beginning the PIM configuration process to determine the appropriate PIM mode for your network applications.

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

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

Restrictions for PIM

Outlines the limitations and constraints when configuring Protocol Independent Multicast (PIM) features.

These are the restrictions for configuring PIM:

- Use ACLs to designate a specified port only as a multicast host port and not as 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.
- Hot Standby Router Protocol-aware (HSRP-aware) PIM is not supported.

Restrictions for configuring Auto-RP and BSR

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

Restrictions for Auto-RP configuration

Consider your network configuration when configuring Auto-RP.

- 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 configuration

Consider your network configuration when configuring BSR.

- 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, configure backup RPs to serve the same group prefixes. This prevents the PIMv2 DRs from selecting a different RP than those selected by PIMv1 DRs, due to the longest match lookup in the RP-mapping database.

Restrictions and guidelines for Auto-RP and BSR configuration

Consider these restrictions when configuring Auto-RP and BSR together.

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

  Note

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

- Because bootstrap messages are sent hop-by-hop, a PIMv1 device prevents these messages from reaching all routers and multi-layer switches in your network. Therefore, if your network has a PIMv1 device in it and only Cisco routers and multi-layer switches, it is best to use Auto-RP.
- If you have a network that includes non-Cisco routers, configure the Auto-RP mapping agent and the BSR on a Cisco PIMv2 router or multi-layer 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 multi-layer 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.

PIMv1 and PIMv2 interoperability

Explains the interoperability and transition capabilities between PIM Version 1 and Version 2 implementations in multicast networks.

PIMv1 and PIMv2 interoperability is a multicast routing capability that

- enables incremental upgrades between PIM Version 1 and Version 2 within one network
- provides automatic version detection and downgrade functionality when mixed versions are detected, and
- maintains compatibility through Auto-RP and BSR integration for RP discovery and announcement.

Version compatibility details

To avoid misconfiguring multicast routing, review this information.

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

You can upgrade to PIMv2 incrementally. PIM Versions 1 and 2 can be configured on different routers and multilayer 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 can perform the same tasks as the PIMv2 BSR when used with the Auto-RP feature. However, Auto-RP is a standalone protocol that is separate from PIMv1 and is proprietary to Cisco. 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.

Information about PIM

Describes Protocol Independent Multicast (PIM), a multicast routing protocol that enables efficient distribution of multicast traffic across networks by building distribution trees and managing multicast group membership independent of the underlying unicast routing protocol.

Protocol independent multicast overview

Explains the Protocol Independent Multicast (PIM) protocol and its characteristics for maintaining IP multicast services.

Protocol Independent Multicast (PIM) is a multicast routing protocol that

- maintains the current IP multicast service mode of receiver-initiated membership
- is not dependent on a specific unicast routing protocol. It can leverage any unicast routing protocol used to populate the unicast routing table, such as Enhanced Interior Gateway Routing Protocol (EIGRP), Open Shortest Path First (OSPF), Border Gateway Protocol (BGP), and static routes, and
- uses unicast routing information to perform the multicast forwarding function.

Protocol independent multicast characteristics

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

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

PIMv2 improvements over PIMv1

Describes the improvements that PIMv2 includes over PIMv1.

PIMv2 includes these improvements over PIMv1.

- A single, active rendezvous point (RP) exists per multicast group, with multiple backup RPs. Multiple active RPs for the same group exist 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

Explains Multicast Source Discovery Protocol (MSDP) for inter-domain source discovery when PIM SM is used.

Multicast Source Discovery Protocol (MSDP) is a protocol that

- enables inter-domain source discovery when PIM SM is used
- allows RPs in different PIM administrative domains to signal new sources to each other, and
- uses source-active (SA) messages to communicate source information between MSDP peers across domains.

MSDP operation

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

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

PIM sparse mode

Describes a multicast routing protocol that uses a pull model to deliver traffic only to network segments with active receivers.

PIM sparse mode (PIM-SM) is a multicast routing protocol that

- uses a pull model to deliver multicast traffic only to network segments with active receivers that have explicitly requested the data
- requires the use of a rendezvous point (RP) that must be administratively configured in the network, and
- scales well to networks of any size, including those with WAN links, by preventing unwanted traffic from flooding the WAN links through its explicit join mechanism.

Sparse mode interface and forwarding behavior

Sparse mode interfaces are added to the multicast routing table only when they receive periodic Join messages from downstream routers or when a directly connected member is present on the interface.When forwarding from a LAN, sparse mode operates if an RP is known for the group. If so, the packets are encapsulated and sent toward the RP. If the multicast traffic from a specific source is sufficient, the first hop router of the receiver may send Join messages toward the source to build a source-based distribution tree.

Role of the RP in PIM-SM

PIM-SM distributes information about active sources by forwarding data packets on the shared tree. Because PIM-SM initially uses shared trees, it requires an RP. The RP must be administratively configured in the network.

Multicast group tracking and registration

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

Path evaluation and unicast routing metrics

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

Managing the SPT switchover

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, which is a global configuration level command.

Rendezvous points

Describes the role and function of rendezvous points in PIM sparse mode networks.

A rendezvous point (RP) is a role that a device performs when operating in Protocol Independent Multicast (PIM) Sparse Mode (SM) that

- acts as the meeting place for sources and receivers of multicast data,
- is required only in networks running PIM SM, and
- forwards traffic to receivers down a shared distribution tree in the PIM-SM model.

RP operation and traffic flow

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 sends a Join message directly to the source. This creates a source-based distribution tree from the source to the receiver. This source tree does not include the RP except when the RP is located within the shortest path between the source and receiver.

In the PIM-SM model, only network segments with active receivers that have explicitly requested multicast data will be forwarded the traffic.

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

Auto-RP

Describes a Cisco-implemented PIM-SM feature that automates RP distribution and configuration in multicast networks.

Auto-RP is a Cisco-implemented PIM-SM feature that

- automates the distribution of group-to-RP mappings in a PIM network
- enables multiple RPs to serve different group ranges or act as backups to each other, and
- uses designated RP-mapping agents to receive RP-announcement messages and arbitrate conflicts.

Auto-RP implementation details

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

Following the introduction of PIM-SM version 1, Cisco implemented a version of PIM-SM with the Auto-RP feature. Auto-RP has these benefits:

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

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

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

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

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

Auto-RP role in PIM networks

Describes how Auto-RP functions within Protocol Independent Multicast networks to automate rendezvous point discovery and mapping distribution.

Auto-RP is a network automation protocol that

- automates the distribution of group-to-rendezvous point (RP) mappings in a PIM network
- enables all routers to automatically discover which RP to use for the groups they support, and
- uses designated RP mapping agents to receive RP announcement messages and arbitrate conflicts.

Auto-RP operation details

The Internet Assigned Numbers Authority (IANA) has assigned two group addresses, 224.0.1.39 and 224.0.1.40, for Auto-RP.

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

Auto-RP benefits in PIM networks

Explains the advantages of implementing Auto-RP in Protocol Independent Multicast networks.

Auto-RP benefits in PIM networks are network administration advantages that

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

Multicast boundaries

Describes administratively-scoped boundaries that limit multicast traffic forwarding outside of domains or subdomains.

A multicast boundary is a network control mechanism that

- limits the forwarding of multicast traffic outside of a domain or subdomain using administratively-scoped addresses,
- prevents multicast traffic whose group addresses fall in the configured range from entering or exiting the interface, and
- provides a firewall for multicast traffic in the specified address range.

Multicast boundary configuration and operation

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

The IANA has designated the multicast address range 239.0.0.0 to 239.255.255.255 as the administratively-scoped addresses. This range of addresses can then be reused in domains administered by different organizations. These addresses are considered local, not globally unique.

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

Note

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

PIM domain border

Describes a network boundary configuration that constrains PIMv2 BSR messages between separate PIM domains.

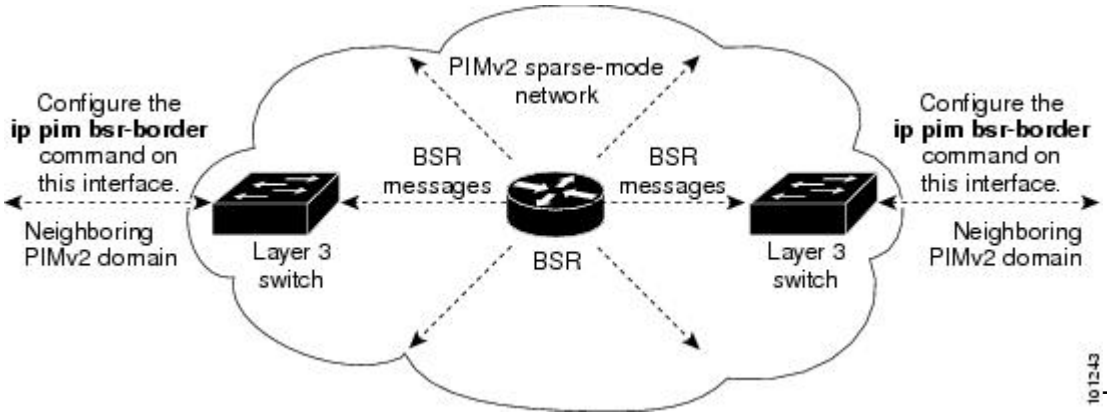
A PIM domain border is a network boundary configuration that

- constrains PIMv2 BSR messages from flowing into or out of separate PIM domains,
- prevents adverse effects on the normal BSR election mechanism that could elect a single BSR across all bordering domains, and
- prevents comingling of candidate RP advertisements that could result in the election of RPs in the wrong domain.

Domain border necessity

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, BSRs, candidate RPs, and candidate BSRs, you must constrain PIMv2 BSR messages to ensure they do not flow into or out of the domain.

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



Thisfiguredisplays how you can configure the PIM domain border by using the **IP PIM BSR-border** command.

PIMv2 bootstrap router

Describes PIMv2 Bootstrap Router (BSR) as a method to distribute group-to-RP mapping information to all PIM routers and multilayer devices in the network.

PIMv2 Bootstrap Router (BSR) is a distribution method that

- distributes group-to-RP mapping information to all PIM routers and multilayer devices in the network,
- eliminates the need to manually configure RP information in every router and switch in the network, and
- uses hop-by-hop flooding of special BSR messages to distribute the mapping information instead of using IP multicast.

BSR election and operation

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.

Flooding of BSR messages

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 RP advertisement and mapping algorithm

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

Multicast forwarding

Explains how multicast traffic is forwarded through networks using distribution trees.

Multicast forwarding is a network routing mechanism that

- accomplishes forwarding of multicast traffic through multicast-capable routers
- creates distribution trees that control the path that IP multicast traffic takes through the network, and
- delivers traffic to all receivers in the multicast group.

Distribution trees and routing notations

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

The notations used in multicast routing tables include the following:

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

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

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

Multicast distribution source trees

Describes the simplest form of multicast distribution tree that uses the shortest path through the network.

A multicast distribution source tree is a network structure that

- has its root at the source host and has branches forming a spanning tree through the network to the receivers,
- uses the shortest path through the network, also referred to as a shortest path tree (SPT), and

- represents the simplest form of a multicast distribution tree.

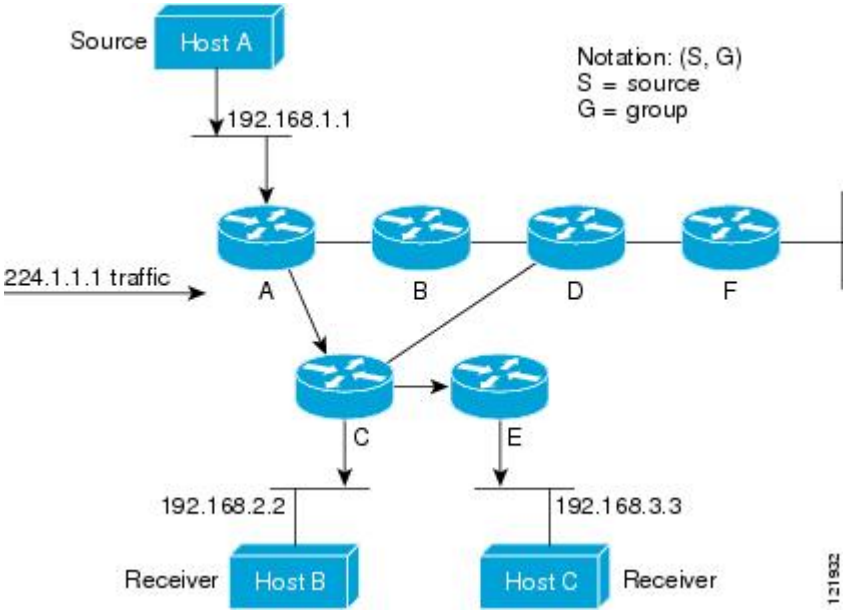
SPT notation and structure

In standard notation, the SPT uses the (S,G) format. In this format, S represents the source and G represents the group.

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

SPT example for multicast group

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



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

Multicast distribution shared tree

Describes a multicast tree topology that uses a single common root point for distributing traffic to all receivers in a multicast group.

A multicast distribution shared tree is a multicast tree topology that

- uses a single common root placed at some chosen point in the network called a rendezvous point (RP)
- forwards source traffic towards the RP on a source tree, then down the shared tree from the RP to reach all receivers, and
- uses wildcard notation written as (*, G), where * means all sources and G represents the multicast group.

Shared tree characteristics

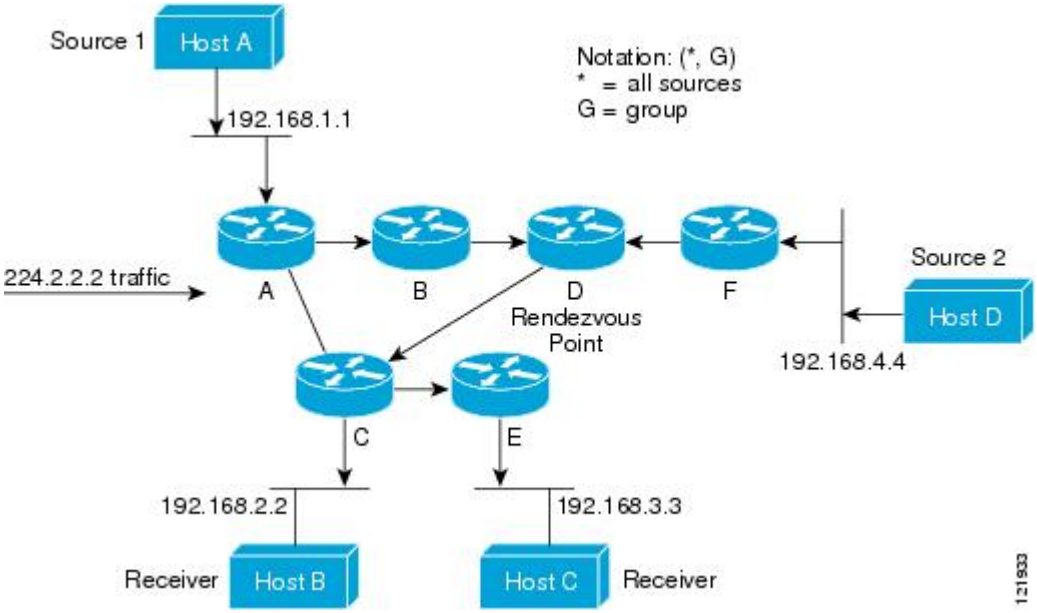
The shared tree operates with these key characteristics:

- The shared tree is unidirectional
- Source traffic is sent towards the RP on a source tree
- Traffic is forwarded down the shared tree from the RP to reach all receivers (unless the receiver is located between the source and the RP, in which case it will be serviced directly)

- Both source trees and shared trees are loop-free
- Messages are replicated only where the tree branches

Distribution trees are dynamically updated as members of multicast groups can join or leave at any time. 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.

Figure 4: Shared tree



Shared tree example

This example shows a shared tree for the group 224.2.2.2 with the root located at Router D. 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, this shared tree would be written as (*, 224.2.2.2).

Source tree advantage

Describes the benefits and trade-offs of source trees in multicast routing.

Source tree advantage is a multicast routing benefit that creates the optimal path between the source and the receivers, guaranteeing the minimum amount of network latency for forwarding multicast traffic, but requires routers to maintain path information for each source which can become a resource issue in networks with thousands of sources and groups.

Memory consumption considerations

Memory consumption from the size of the multicast routing table is a factor that network designers must take into consideration.

Shared tree advantage

Describes the benefits and trade-offs of shared tree architectures in multicast routing environments.

A shared tree advantage is a network topology benefit that

- requires the minimum amount of state in each router

- lowers the overall memory requirements for a network that only allows shared trees, and
- simplifies multicast routing compared to source-specific trees.

Shared tree considerations

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

Unicast routing

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

Multicast forwarding

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

PIM shared trees and source trees

Describes PIM shared trees and source trees distribution mechanisms for multicast data delivery.

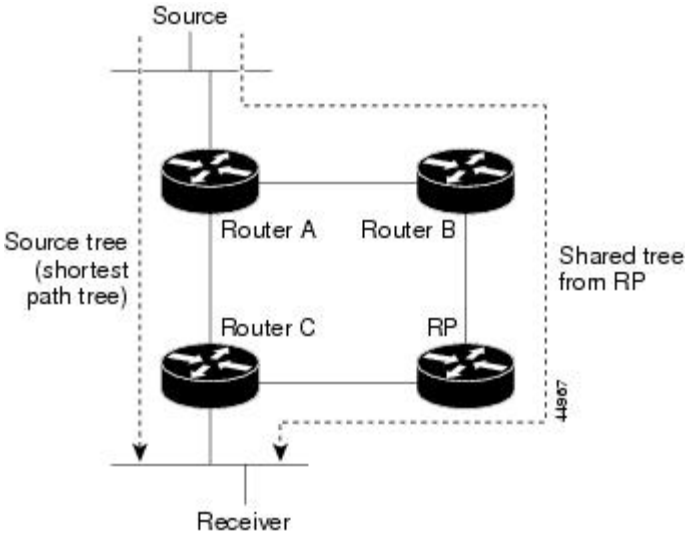
A PIM shared tree is a multicast data distribution mechanism that

- delivers data from senders to group members across a single data-distribution tree rooted at the RP
- enables leaf routers to switch to source trees when data rates warrant more efficient delivery, and
- allows configuration to stay on shared trees or set traffic thresholds for source tree transitions.

Distribution tree behavior

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

Figure 5: Shared tree and source tree (shortest-path 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 moves to a source tree after receiving the first data packet from a source.

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

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

PIM message types and forwarding paths

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

Shared tree utilization

Multiple sources sending to groups use the shared tree. You can configure the PIM device to stay on the shared tree.

SPT switchover thresholds

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.

Traffic rate and leaf router behavior

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.

Apply thresholds to specific groups

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

Describes an algorithm used for forwarding multicast datagrams by determining upstream and downstream directions in multicast routing.

Reverse Path Forwarding (RPF) is an algorithm that

- forwards multicast traffic away from the source rather than toward the receiver
- uses existing unicast routing tables to determine upstream and downstream neighbors, and
- ensures loop-free distribution trees by forwarding packets only if received on the upstream interface.

Multicast forwarding characteristics

In multicast forwarding, the source sends traffic to an arbitrary group of hosts represented by a multicast group address. The multicast router must determine which direction is the upstream direction (toward the source) and which one is the downstream direction (or directions) toward the receivers. If there are multiple downstream paths, the router replicates the packet and forwards it along the most appropriate downstream paths based on the best unicast route metric. This does not mean all available paths are used.

Protocol Independent Multicast (PIM) uses unicast routing information to create a distribution tree along the reverse path from the 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.

The RPF check helps to guarantee that the distribution tree will be loop-free by ensuring a router will forward a multicast packet only if it is received on the upstream interface.

RPF check

Describes a multicast packet validation procedure that determines whether packets are forwarded or dropped based on their arrival interface.

An RPF check is a multicast packet validation procedure that

- determines if a multicast packet arrives on the correct interface by checking the reverse path to the source
- forwards the packet if the check succeeds or drops it if the check fails, and
- uses different validation methods for source trees and shared trees in PIM environments.

RPF check procedures

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

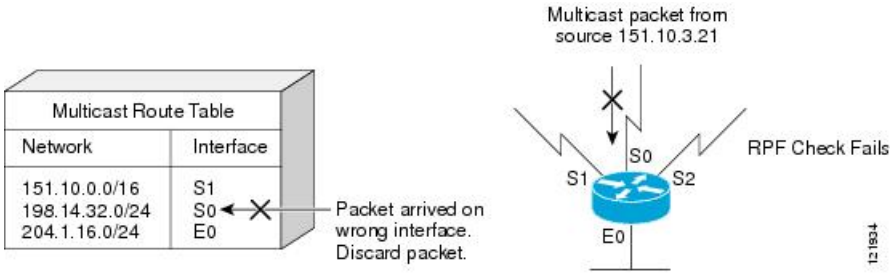
1. The router looks up the source address in the unicast routing table to determine if the packet has arrived on the interface that is on the reverse path back to the source.

2. If the packet has arrived on the interface leading back to the source, the RPF check succeeds, and the packet is forwarded to 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.

RPF check failure example

This example shows an unsuccessful RPF check.

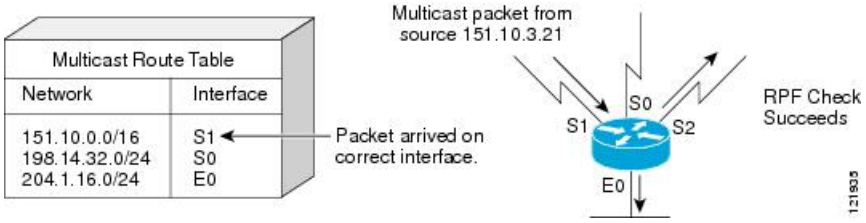
Figure 6: RPF check fails



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.

This example shows a successful RPF check.

Figure 7: 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.

Default PIM routing configuration

Lists the default PIM routing configuration settings for the device.

This reference provides the default PIM routing configuration settings for the device, including multicast routing status, PIM version, mode, RP address, and other key parameters.

Table 5: Default multicast routing configuration

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

Feature	Default Setting
Multicast routing	Disabled on all interfaces.
PIM version	Version 2.
PIM mode	No mode is defined.
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.

How to configure PIM

Describes the process of configuring Protocol Independent Multicast (PIM) on Cisco devices to enable efficient multicast routing in network environments.

Configuring a rendezvous point

Describes how to configure a rendezvous point (RP) for handling sparse multicast groups using manual assignment, Auto-RP, or Bootstrap Router (BSR) methods depending on your PIM version and network requirements.

A rendezvous point (RP) is required to handle a multicast group as a sparse group. You can use one of these methods:

- By manually assigning an RP to multicast groups
- As a standalone RP using a Cisco-proprietary protocol that operates separately from PIMv1
- By using a standards track protocol in the Internet Engineering Task Force (IETF), which includes configuring PIMv2 BSR

Note

Depending on the PIM version and router types in your network, you can use Auto-RP, BSR, or a combination of both.

Manually assign an RP to multicast groups

Configure a rendezvous point (RP) address for specific multicast groups using access lists to control which groups use the designated RP.

This task is optional and it configures a designated rendezvous point (RP) for multicast groups to serve as a meeting place for multicast sources and group members. The RP enables senders to announce their existence through register messages and allows receivers to join multicast groups using explicit join messages.

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

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

Note

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

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

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **ip pim rp-address** *IP-address* [*access-list-number*] [**override**] command to configure the address of a PIM RP.

   Example

   ```
   Device(config)# ip pim rp-address 10.1.1.1 20 override
   ```

   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**
>
> 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. A PIM domain can use only one RP address at a time. The access list conditions specify for which groups the device is an RP.

- For *IP-address*, enter the unicast address of the RP in dotted-decimal notation.
- (Optional) For *access-list-number*, enter an IP standard access list number from 1 to 99. If no access list is configured, the RP is used for all groups.
- (Optional) The **override** keyword indicates that if there is a conflict between the RP configured with this command and one learned by Auto-RP or BSR, the RP configured with this command prevails.

3. Use the **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] command to create a standard access list, repeating the command as many times as necessary.

   Example

   ```
   Device(config)# access-list 25 permit 10.5.0.1 255.224.0.0
   ```

   - For *access-list-number*, enter the access list number specified in Step 2.
   - The **deny** keyword denies access if the conditions are matched.
   - The **permit** keyword permits access if the conditions are matched.
   - For *source*, enter the multicast group address for which the RP should be used.
   - (Optional) For *source-wildcard*, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore.

   An implicit deny statement always terminates the access list.

4. Use the **end** command to return to privileged EXEC mode.

   Example

   ```
   Device(config)# end
   ```

5. Use the **show running-config** command to verify your entries.

   Example

   ```
   Device# show running-config
   ```

Set up Auto-RP in a new internetwork

Configure Auto-RP in a new internetwork to automatically distribute RP information.

Auto-RP allows for automatic RP discovery and distribution in PIM networks, eliminating the need for manual RP configuration on all devices.

This process configures Auto-RP in a new internetwork environment where you need to establish automatic RP distribution.

Before you begin

To set up Auto-RP in a new internetwork, perform these steps:

Procedure

1. Use the **show running-config** command to verify your entries.

   Example

   ```
   Device# show running-config
   ```

   It was previously configured with the IP PIM RP-address global configuration command.

   > **Note**
   >
   > This step is not required for spare-dense-mode environments.

   Choose an RP that has reliable connectivity and is available across the network. Assign this RP to the global groups, including 224.x.x.x and any other similar groups. Do not change the group address range served by this RP. Auto-RP dynamically discovered RPs override statically configured RPs. If you need an additional RP for local groups, configure that RP separately.

2. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

3. Use the **ip pim send-rp-announce** *interface-id* **scope** *ttl* **group-list** *access-list-number* **interval** *seconds* command to configure another PIM device as the candidate RP for local groups.

   Example

   ```
   Device(config)# ip pim send-rp-announce gigabitethernet 1/5 scope 20 group-list
    10 interval 120
   ```

   - 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.

4. Use the **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] command to create a standard access list, repeating the command as many times as necessary.

Example

```
Device(config)# access-list 10 permit 10.10.0.0
```

- For *access-list-number*, enter the access list number.
- The **deny** keyword denies access if the conditions are matched.
- The **permit** keyword permits access if the conditions are matched.
- For *source*, enter the multicast group address range for which the RP should be used.
- (Optional) For *source-wildcard*, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore.

> **Note**
>
> Recall that the access list is always terminated by an implicit deny statement for everything.

5. Use the **ip pim send-rp-discovery scope** *ttl* command to find a device whose connectivity is not likely to be interrupted, and assign it the role of RP-mapping agent.

   Example

   ```
   Device(config)# ip pim send-rp-discovery scope 50
   ```

   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.

6. Use the **end** command to return to privileged EXEC mode.

   Example

   ```
   Device(config)# end
   ```

7. Use the **show running-config** command to verify your entries.

   Example

   ```
   Device# show running-config
   ```

8. Use the **show ip pim rp mapping** command to display active RPs that are cached with associated multicast routing entries.

   Example

   ```
   Device# show ip pim rp mapping
   ```

9. Use the **show ip pim rp** command to display the information cached in the routing table.

Example

```
Device# show ip pim rp
```

Add Auto-RP to an existing sparse-mode cloud

Configure Auto-RP in an existing sparse-mode network to enable dynamic RP discovery while maintaining existing multicast infrastructure.

This task enables dynamic RP discovery in an existing sparse-mode multicast environment while minimizing disruption to the current infrastructure.

This section provides suggestions to minimize disruption of the multicast infrastructure when initially deploying Auto-RP into an existing sparse-mode cloud.

This procedure is optional.

Before you begin

Follow these steps to add Auto-RP to an existing sparse-mode cloud:

Procedure

1. Use the **show running-config** command to verify that a default RP is already configured on all PIM devices and the RP in the sparse-mode network.

   Example

   ```
   Device# show running-config
   ```

   Note

   This RP Iwas previously configured with the **ip pim rp-address** global configuration command.This step is not required for spare-dense-mode environments.

   The RP must have good connectivity and network-wide availability. Use this RP for global groups such as 224.x.x.x. Keep the group address range for this RP unchanged. Auto-RP dynamically discovered RPs take precedence over static configurations. If needed, designate a second RP for local groups.

2. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

3. Use the **ip pim send-rp-announce** *interface-id* **scope** *ttl* **group-list** *access-list-number* **interval** *seconds* command to configure another PIM device to be the candidate RP for local groups.

Example

```
Device(config)# ip pim send-rp-announce gigabitethernet 1/5 scope 20 group-list
 10 interval 120
```

- 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.

4. Use the **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] command to create a standard access list, repeating the command as many times as necessary.

   Example

   ```
   Device(config)# access-list 10 permit 224.0.0.0 15.255.255.255
   ```

   - 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 for the source. Use ones for the bit positions to ignore.

   Recall that the access list is always terminated by an implicit deny statement for everything.

5. Use the **ip pim send-rp-discovery scope** *ttl* command to find a device whose connectivity is not likely to be interrupted, and assign it the role of RP-mapping agent.

   Example

   ```
   Device(config)# ip pim send-rp-discovery scope 50
   ```

   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.

   > **Note**
   >
   > To remove the device as the RP-mapping agent, use the **no ip pim send-rp-discovery** global configuration command.

6. Use the **end** command to return to privileged EXEC mode.

Example

```
Device(config)# end
```

7. Use the **show running-config** command to verify your entries.

   Example

   ```
   Device# show running-config
   ```

8. Use the **show ip pim rp mapping** command to display active RPs that are cached with associated multicast routing entries.

   Example

   ```
   Device# show ip pim rp mapping
   ```

9. Use the **show ip pim rp** command to display the information cached in the routing table.

   Example

   ```
   Device# show ip pim rp
   ```

Preventing join messages to false RPs

Explains how to prevent join messages from being sent to false rendezvous points in PIM networks.

Preventing join messages to false RPs is a PIM network security practice that ensures join messages are only sent to legitimate rendezvous points by validating RP advertisements before accepting them.

Configuration verification and remediation

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

Filter incoming RP announcement messages

Configure filtering of incoming RP announcement messages to prevent maliciously configured routers from masquerading as candidate RPs.

Add configuration commands to the mapping agents to prevent a router from masquerading as a candidate RP. This step helps prevent security issues caused by unauthorized routers.

Procedure

---

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **ip pim rp-announce-filter rp-list** *access-list-number* **group-list** *access-list-number* command to filter incoming RP announcement messages.

   Example

   ```
   Device(config)# ip pim rp-announce-filter rp-list 10 group-list 14
   ```

   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. If permitted, these addresses are 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.

3. Use the **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] command to create a standard access list, repeating the command as many times as necessary.

   Example

   ```
   Device(config)# access-list 10 permit 10.8.1.0 255.255.224.0
   ```

   - For *access-list-number*, enter the access list number specified in Step 2.
   - The **deny** keyword denies access if the conditions are matched.
   - The **permit** keyword permits access if the conditions are matched.
   - Create an access list that specifies from which routers and multilayer switches the mapping agent accepts candidate RP announcements (RP-list ACL).
   - Create an access list that specifies the range of multicast groups from which to accept or deny (group-list ACL).
   - For *source*, enter the multicast group address range for which the RP should be used.
   - (Optional) For *source-wildcard*, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore.

   The access list is always terminated by an implicit deny statement for everything.

4. Use the **end** command to return to privileged EXEC mode.

   Example

   ```
   Device(config)# end
   ```

5. Use the **show running-config** command to verify your entries.

   Example

   ```
   Device# show running-config
   ```

Configuring PIMv2 BSR

Describes the process for configuring PIMv2 Bootstrap Router (BSR), including optional tasks such as defining PIM domain borders, IP multicast boundaries, and configuring candidate BSRs and RPs.

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

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

Define the PIM domain border

Configure the PIM bootstrap message boundary for the PIM domain on interfaces that connect to other bordering PIM domains.

Define a PIM bootstrap message boundary to prevent the device from sending or receiving PIMv2 BSR messages on specific interfaces that connect to other PIM domains.

Perform this procedure only when establishing boundaries between different PIM domains.

Before you begin

Follow these steps to define the PIM domain border:

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **interface** *interface-id* command to specify the interface to be configured and enter interface configuration mode.

   Example

   ```
   Device(config)# interface gigabitethernet 1/3
   ```

3. Use the **ip pim bsr-border** command to define the PIM bootstrap message boundary for the PIM domain.

   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. The device will not send or receive PIMv2 BSR messages on this interface.

> **Note**
>
> To remove the PIM border, use the **no ip pim bsr-border** interface configuration command.

4. Use the **end** command to return to privileged EXEC mode.

   Example

   ```
   Device(config)# end
   ```

5. Use the **show running-config** command to verify your entries.

   Example

   ```
   Device# show running-config
   ```

Define the IP multicast boundary

Configure a multicast boundary to prevent Auto-RP messages from entering the PIM domain.

You define a multicast boundary to prevent Auto-RP messages from entering the PIM domain. Create an access list that denies packets destined for 224.0.1.39 and 224.0.1.40. These addresses carry Auto-RP information.

Procedure

---

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **access-list** *access-list-number* **deny** *source* [*source-wildcard*] command to create a standard access list, repeating the command as many times as necessary.

   Example

   ```
   Device(config)# access-list 12 deny 224.0.1.39
   Device(config)# access-list 12 deny 224.0.1.40
   ```

   - The *access-list-number* value can be any number from 1 to 99.
   - The **deny** keyword denies access if the conditions are matched.
   - For the *source*, enter the multicast addresses 224.0.1.39 and 224.0.1.40. These addresses carry Auto-RP information.
   - (Optional) For the *source-wildcard* parameter, enter the wildcard bits in dotted decimal notation. These bits are 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.

3. Use the **interface** *interface-id* command to specify the interface to be configured and enter interface configuration mode.

   Example

   ```
   Device(config)# interface gigabitethernet 1/3
   ```

4. Use the **ip multicast boundary** *access-list-number* command to configure the boundary, specifying the access list you created in Step 2.

   Example

   ```
   Device(config-if)# ip multicast boundary 12
   ```

5. Use the **end** command to return to privileged EXEC mode.

   Example

   ```
   Device(config)# end
   ```

6. Use the **show running-config** command to verify your entries.

   Example

   ```
   Device# show running-config
   ```

Configure candidate BSRs

Configure candidate BSRs on your device to participate in BSR election.

Configure candidate BSRs to enable your device to participate in the Bootstrap Router (BSR) election process in a PIM sparse mode network.

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

This procedure is optional.

Before you begin

Follow these steps to configure candidate BSRs:

Procedure

---

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **ip pim bsr-candidate** *interface-id hash-mask-length* [*priority*] command to configure your device to be a candidate BSR.

Example

```
Device(config)# ip pim bsr-candidate gigabitethernet 1/3 28 100
```

- For *interface-id*, enter the interface on this device from which the BSR address is derived to make it a candidate. Ensure that this interface is enabled with PIM. Valid interfaces include physical ports, port channels, and VLANs.
- For *hash-mask-length*, specify the mask length (up to 32 bits) that is applied to the group address with an AND operation before the hash function runs. All groups sharing the same seed hash use the same RP. For instance, if you specify 24, only the first 24 bits of the group addresses are considered.
- (Optional) For *priority*, enter a priority value from 0 to 255. The BSR with the higher priority is preferred. If devices share the same priority, the device with the highest IP address becomes the BSR. If you do not specify a value, the default is 0.

3. Use the **end** command to return to privileged EXEC mode.

Example

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

4. Use the **show running-config** command to verify your entries.

Example

```
Device# show running-config
```

Configure the candidate RPs

Configure the candidate RPs to enable IP multicast routing in your network.

Configuring candidate RPs enables devices to advertise themselves as potential rendezvous points for IP multicast groups in a PIM sparse mode network.

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

This procedure is optional.

Before you begin

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

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

Follow these steps to configure the candidate RPs:

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **ip pim rp-candidate** *interface-id* [**group-list** *access-list-number*] command to configure your device to be a candidate RP.

   Example

   ```
   Device(config)# ip pim rp-candidate gigabitethernet 1/5 group-list 10
   ```

   - 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 you do not specify a group-list, the device becomes a candidate RP for all groups.

3. Use the **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] command to create a standard access list, repeating the command as many times as necessary.

   Example

   ```
   Device(config)# access-list 10 permit 239.0.0.0 0.255.255.255
   ```

   - For *access-list-number*, enter the access list number you specified earlier.
   - The **deny** keyword denies access if the conditions are matched. The **permit** keyword permits access if the conditions are matched.
   - For *source*, enter the number of the network or host from which the packet is being sent.
   - (Optional) For *source-wildcard*, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore.

   The access list is always terminated by an implicit deny statement for everything.

4. Use the **end** command to return to privileged EXEC mode.

   Example

   ```
   Device(config)# end
   ```

5. Use the **show running-config** command to verify your entries.

   Example

   ```
   Device# show running-config
   ```

Configure Sparse Mode with Auto-RP

Configures auto-rendezvous point (Auto-RP) to enable multicast routing in a network environment. This procedure supports optional integration with anycast RP.

Before you begin

- All access lists that are needed when Auto-RP is configured should be configured prior to beginning the configuration task.
- Start the procedure in global configuration mode using the **configure terminal** command.

> **Note**
>
> - When configuring Auto-RP, you must either configure your interfaces in Sparse-Dense Mode or enable the Auto-RP listener if using strict Sparse Mode.

Procedure

1. Use the **ip multicast-routing** command to enable IP multicast routing.

   Example

   ```
   Device(config)# ip multicast-routing
   ```

2. Use the **interface** *type number* command to select an interface that is connected to hosts on which PIM can be enabled.

   Example

   ```
   Device(config)# interface Gigabitethernet 1/0
   ```

3. Use the **ip pim sparse-mode** command to enable PIM sparse mode on an interface.

   Example

   ```
   Device(config-if)# ip pim sparse-mode
   ```

   When configuring Auto-RP in sparse mode, you must also configure the Auto-RP listener in the next step.

   Run the **exit** command after this step to return to global configuration mode.

4. Use the **ip pim send-rp-announce** {*interface-type interface-number* | *ip-address*} **scope** *ttl-value* [**group-list** *access-list*] [**interval** *seconds*] command to configure the device to send RP announcements from the candidate RP.

   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.

Note

Repeat Steps 1 through 4 on all PIM interfaces.

5. Use the **ip pim send-rp-discovery** [*interface-type interface-number*] **scope** *ttl-value* [**interval** *seconds*] command to configure the device to be an RP mapping agent.

   Perform this step on RP mapping agent devices or on combined RP/RP mapping agent devices.

Note

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

- Use the optional *interface-type* and *interface-number* arguments to define which IP address is to be used as the source address of the RP mapping agent.
- Use the **scope** keyword and *ttl-value* argument to specify the Time-to-Live (TTL) value in the IP header of Auto-RP discovery messages.
- Use the optional **interval** keyword and *seconds* argument to specify the interval at which Auto-RP discovery messages are sent.

Note

Lowering the interval at which Auto-RP discovery messages are sent from the default value of 60 seconds results in more frequent floodings of the group-to-RP mappings. In some network environments, the disadvantages of lowering the interval (more control packet overhead) may outweigh the advantages (more frequent group-to-RP mapping updates).

- The example shows limiting the Auto-RP discovery messages to 31 hops on loopback interface 1.

6. Use the **ip pim rp-announce-filter rp-list** *access-list* **group-list** *access-list* command to filter incoming RP announcement messages sent from candidate RPs (C-RPs) to the RP mapping agent.

   Example

   ```
   Device(config)# ip pim rp-announce-filter rp-list 1 group-list 2
   ```

   Perform this step on the RP mapping agent only.

7. Use the **interface** *type number* command to select an interface that is connected to hosts on which PIM can be enabled.

   Example

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

8. Use the **ip multicast boundary** *access-list* [**filter-autorp**] command to configure an administratively scoped boundary.
   - Perform this step on the interfaces that are boundaries to other devices.
   - The access list is not shown in this task.
   - An access list entry that uses the **deny** keyword creates a multicast boundary for packets that match that entry.

   Use the **end** command to return to privileged EXEC mode.

9. (Optional) Use the **show** commands to verify the configuration.
   a) Use the **show ip pim autorp** command to display Auto-RP information.

      Example

      ```
      Device# show ip pim autorp
      ```

   b) Use the **show ip pim rp** [**mapping**] [*rp-address*] command to display RPs known in the network and show how the device learned about each RP.

      Example

      ```
      Device# show ip pim rp mapping
      ```

   c) Use the **show ip igmp groups** [*group-name* | *group-address* | *interface-type interface-number*] [**detail**] command to display multicast groups with directly connected receivers learned through IGMP.

      Example

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

      - A receiver must be active on the network at the time that this command is issued in order for receiver information to be present on the resulting display.

   d) Use the **show ip mroute** [*group-address* | *group-name*] [*source-address* | *source-name*] [*interface-type interface-number*] [**summary**] [**count**] [**active** *kbps*] command to display the contents of the IP multicast routing table.

Example

```
Device# show ip mroute cbone-audio
```

Delay the use of PIM shortest-path tree

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

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

Before you begin

Follow these steps to delay the use of PIM shortest-path tree:

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] command to create a standard access list.

   Example

   ```
   Device(config)# access-list 16 permit 225.0.0.0 0.255.255.255
   ```

   - 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 apply to the source. Enter ones in the bit positions that you want to ignore.

   An implicit deny statement always terminates the access list.

3. Use the **ip pim spt-threshold** {*kbps* | **infinity**} [**group-list** *access-list-number*] command to specify the threshold required before moving to the shortest-path tree (SPT).

   Example

   ```
   Device(config)# ip pim spt-threshold infinity group-list 16
   ```

   - For *kbps*, specify the traffic rate in kilobits per second. The default is 0 kbps.

> **Note**
>
> Device hardware only accepts 0 kbps, although the supported range is 0 to 4294967.

- Specify **infinity** to ensure that all sources for the specified group use the shared tree and never switch to the source tree.
- (Optional) For **group-list** *access-list-number*, specify the access list created in Step 2. If the value is 0 or if the group list is not used, the threshold applies to all groups.

4. Use the **end** command to return to privileged EXEC mode.

Example

```
Device(config)# end
```

5. Use the **show running-config** command to verify your entries.

Example

```
Device# show running-config
```

Modify the PIM router-query message interval

Configure the frequency at which the device sends PIM router-query messages.

This task allows you to configure the frequency at which PIM routers and multilayer switches send PIM router-query messages to determine the designated router (DR) for each LAN segment.

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

With PIM DM operation, the DR is relevant only when IGMPv1 is in use. Because IGMPv1 does not have a querier election process, the elected DR serves as the IGMP querier. For PIM-SM operation, the DR is the device directly connected to the multicast source. This device sends PIM register messages to notify the RP to forward multicast traffic from a source down the shared tree. In this situation, the DR is the device with the highest IP address.

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

Example

```
Device# configure terminal
```

2. Use the **interface** *interface-id* command to specify the interface to be configured and enter interface configuration mode.

Example

```
Device(config)# interface gigabitethernet 1/3
```

3. Use the **ip pim query-interval** *seconds* command to configure the frequency at which the device sends PIM router-query messages.

   Example

   ```
   Device(config-if)# ip pim query-interval 45
   ```

   The default is 30 seconds. The range is 1 to 65535.

4. Use the **end** command to return to privileged EXEC mode.

   Example

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

5. Use the **show ip igmp interface** [*interface-id*] command to verify your entries.

   Example

   ```
   Device# show ip igmp interface
   ```

Verifying PIM operations

Describes how to verify that Protocol Independent Multicast (PIM) is operating correctly by checking neighbor relationships, routing table entries, and multicast forwarding state information.

IP multicast operation verification in PIM-SM or PIM-SSM networks

Describes the process for verifying IP multicast operation in PIM-SM and PIM-SSM network environments.

To verify IP multicast operation in a PIM-SM or PIM-SSM network environment, begin the process on the last hop router, then continue on routers along the SPT until reaching the first hop router. This approach helps ensure that IP multicast traffic is routed properly through the network.

Perform optional tasks to verify IP multicast operation in a PIM-SM or PIM-SSM network. These tasks help you locate a faulty hop when sources or receivers do not operate as expected.

> **Note**
>
> If packets do not reach their expected destinations, consider disabling IP multicast fast switching. This action places the router in process switching mode. If packets begin to reach their destinations after disabling IP multicast fast switching, the issue likely relates to fast switching.

Verify IP multicast on the first hop router

Verify IP multicast operations on the first hop router.

This task verifies IP multicast operations on the first hop router to confirm that multicast routes and active source traffic are being handled correctly.

To confirm that multicast forwarding information is present for the source and group, use this verification procedure on the first hop router.

Procedure

1. Use the **show ip mroute** [*group-address*] command to confirm that the F flag has been set for mroutes on the first hop router.

   Example

   ```
   Device# show ip mroute 239.1.2.3
   (*, 239.1.2.3), 00:18:10/stopped, RP 172.16.0.1, flags: SPF
     Incoming interface: Serial1/0, RPF nbr 172.31.200.2
     Outgoing interface list: Null

   (10.0.0.1, 239.1.2.3), 00:18:10/00:03:22, flags: FT
     Incoming interface: GigabitEthernet0/0, RPF nbr 0.0.0.0
     Outgoing interface list:
       Serial1/0, Forward/Sparse, 00:18:10/00:03:19
   ```

2. Use the **show ip mroute active** [*kb/s*] command to display information about active multicast sources sending to groups.

   Example

   ```
   Device# show ip mroute active
   Active IP Multicast Sources - sending >= 4 kbps

   Group: 239.1.2.3, (?)
      Source: 10.0.0.1 (?)
        Rate: 20 pps/4 kbps(1sec), 4 kbps(last 30 secs), 4 kbps(life avg)
   ```

   The output of this command provides information about the multicast packet rate for active sources.

> **Note**
>
> By default, the output of the **show ip mroute** command with the **active** keyword displays information about active sources sending traffic to groups at a rate of 4 kb/s or higher. To view information about sources sending lower-rate traffic, specify 1 for the *kb/s* argument. This setting displays all active source traffic at rates of 1 kb/s or more.

Verify IP multicast on routers along the SPT

Verify IP multicast operations on routers along the SPT in a PIM-SM or PIM-SSM network.

This task verifies IP multicast operations on routers along the shortest path tree (SPT) in a PIM-SM or PIM-SSM network to ensure proper multicast routing and forwarding.

Use this verification procedure on routers along the SPT. This confirms that IP multicast is functioning correctly in your PIM-SM or PIM-SSM network environment.

Procedure

1. Use the **show ip mroute** [*group-address*] command to confirm the RPF neighbor towards the source for a particular group or groups.

   Example

   ```
   Device# show ip mroute 239.1.2.3
   (*, 239.1.2.3), 00:17:56/00:03:02, RP 172.16.0.1, flags: S
     Incoming interface: Null, RPF nbr 0.0.0.0
     Outgoing interface list:
       GigabitEthernet0/0, Forward/Sparse, 00:17:56/00:03:02

   (10.0.0.1, 239.1.2.3), 00:15:34/00:03:28, flags: T
     Incoming interface: Serial1/0, RPF nbr 172.31.200.1
     Outgoing interface list:
       GigabitEthernet0/0, Forward/Sparse, 00:15:34/00:03:02
   ```

2. Use the **show ip mroute active** command to display information about active multicast sources sending to groups.

   Example

   ```
   Device# show ip mroute active
   Active IP Multicast Sources - sending >= 4 kbps

   Group: 239.1.2.3, (?)
      Source: 10.0.0.1 (?)
        Rate: 20 pps/4 kbps(1sec), 4 kbps(last 30 secs), 4 kbps(life avg)
   ```

   The output of this command provides information about the multicast packet rate for active sources.

> **Note**
>
> By default, the output of the **show ip mroute** command with the **active** keyword displays information about active sources sending traffic to groups at a rate greater than or equal to 4 kb/s. To display information about active sources sending low-rate traffic to groups (that is, traffic less than 4 kb/s), specify a value of 1 for the *kb/s* argument. Specifying a value of 1 for this argument displays information about active sources sending traffic to groups at a rate equal to or greater than 1 kb/s, which effectively displays information about all possible active source traffic.

Verify IP multicast operation on the last hop router

Verify IP multicast operations on the last hop router.

This task verifies IP multicast operations on the last hop router to confirm receiver membership, RP mapping, multicast routing, and multicast forwarding behavior.

On the last hop router, confirm that multicast groups with directly connected receivers are learned and multicast traffic is forwarded correctly.

Procedure

1. Use the **show ip igmp groups** command to verify IGMP memberships on the last hop router.

   Example

   ```
   Device# show ip igmp groups
   IGMP Connected Group Membership
   Group Address    Interface                Uptime    Expires   Last Reporter
   239.1.2.3        GigabitEthernet1/0     00:05:14  00:02:14  10.1.0.6
   224.0.1.39       GigabitEthernet0/0     00:09:11  00:02:08  172.31.100.1
   ```

   This command confirms the multicast groups with receivers that are directly connected to the last hop router and learned through IGMP.

2. Use the **show ip pim rp mapping** command to confirm that group-to-RP mappings are being populated correctly on the last hop router.

   Example

   ```
   Device# show ip pim rp mapping
   PIM Group-to-RP Mappings

   Group(s) 224.0.0.0/4
     RP 172.16.0.1 (?), v2v1
       Info source: 172.16.0.1 (?), elected via Auto-RP
            Uptime: 00:09:11, expires: 00:02:47
   ```

> **Note**
>
> Ignore this step if you are verifying a last hop router in a PIM-SSM network. The **show ip pim rp mapping** command does not work with routers in a PIM-SSM network because PIM-SSM does not use RPs. If configured correctly, PIM-SSM groups do not appear in the output of this command.

3. Use the **show ip mroute** command to verify that the mroute table is being populated properly on the last hop router.

   Example

```
Device# show ip mroute
(*, 239.1.2.3), 00:05:14/00:03:04, RP 172.16.0.1, flags: SJC
  Incoming interface: GigabitEthernet0/0, RPF nbr 172.31.100.1
  Outgoing interface list:
    GigabitEthernet1/0, Forward/Sparse, 00:05:10/00:03:04

(10.0.0.1, 239.1.2.3), 00:02:49/00:03:29, flags: T
  Incoming interface: GigabitEthernet0/0, RPF nbr 172.31.100.1
  Outgoing interface list:
    GigabitEthernet1/0, Forward/Sparse, 00:02:49/00:03:04

(*, 224.0.1.39), 00:10:05/stopped, RP 0.0.0.0, flags: DC
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1/0, Forward/Sparse, 00:05:15/00:00:00
    GigabitEthernet0/0, Forward/Sparse, 00:10:05/00:00:00

(172.16.0.1, 224.0.1.39), 00:02:00/00:01:33, flags: PTX
  Incoming interface: GigabitEthernet0/0, RPF nbr 172.31.100.1
```

4. Use the **show ip interface** [*type number*] command to verify that multicast fast switching is enabled on the outgoing interface on the last hop router.

   Example

```
Device# show ip interface GigabitEthernet 0/0/0
GigabitEthernet0/0/0 is up, line protocol is up
  Internet address is 172.31.100.2/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Multicast reserved groups joined: 224.0.0.1 224.0.0.22 224.0.0.13
      224.0.0.5 224.0.0.6
  Outgoing access list is not set
  Inbound  access list is not set
  Proxy ARP is enabled
  Local Proxy ARP is disabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  IP fast switching is enabled
```

```
  IP fast switching on the same interface is disabled
  IP Flow switching is disabled
  IP CEF switching is disabled
  IP Fast switching turbo vector
  IP multicast fast switching is enabled
  IP route-cache flags are Fast
  Router Discovery is disabled
  IP output packet accounting is disabled
  IP access violation accounting is disabled
  TCP/IP header compression is disabled
  RTP/IP header compression is disabled
  Policy routing is disabled
  Network address translation is disabled
  WCCP Redirect outbound is disabled
  WCCP Redirect inbound is disabled
  WCCP Redirect exclude is disabled
  BGP Policy Mapping is disabled
```

Using the **no ip mroute-cache** interface command disables IP multicast fast switching. If fast switching is disabled, packets are forwarded through the process-switched path.

5. Use the **show ip mfib** command to display the forwarding entries and interfaces in the IP Multicast Forwarding Information Base (MFIB).

   Example

```
Device# show ip mfib
```

6. Use the **show ip pim interface count** command to confirm that multicast traffic is being forwarded on the last hop router.

   Example

```
Device# show ip pim interface count

State: * - Fast Switched, H - Hardware Switching Enabled
Address          Interface                FS Mpackets In/Out
172.31.100.2     GigabitEthernet0/0     *   4122/0
10.1.0.1         GigabitEthernet1/0     *   0/3193
```

7. Use the **show ip mroute count** command to confirm that multicast traffic is being forwarded on the last hop router.

   Example

```
Device# show ip mroute count
IP Multicast Statistics
6 routes using 4008 bytes of memory
3 groups, 1.00 average sources per group
Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kilobits per second
Other counts: Total/RPF failed/Other drops(OIF-null, rate-limit etc)

Group: 239.1.2.3, Source count: 1, Packets forwarded: 3165, Packets received:
 3165
  RP-tree: Forwarding: 0/0/0/0, Other: 0/0/0
  Source: 10.0.0.1/32, Forwarding: 3165/20/28/4, Other: 0/0/0

Group: 224.0.1.39, Source count: 1, Packets forwarded: 21, Packets received:
 120
```

```
  Source: 172.16.0.1/32, Forwarding: 21/1/48/0, Other: 120/0/99

Group: 224.0.1.40, Source count: 1, Packets forwarded: 10, Packets received:
 10
  Source: 172.16.0.1/32, Forwarding: 10/1/48/0, Other: 10/0/0
```

8. Use the **show ip mroute active** [*kb/s*] command to display information about active multicast sources sending traffic to groups on the last hop router.

   Example

```
Device# show ip mroute active
Active IP Multicast Sources - sending >= 4 kbps

Group: 239.1.2.3, (?)
   Source: 10.0.0.1 (?)
     Rate: 20 pps/4 kbps(1sec), 4 kbps(last 50 secs), 4 kbps(life avg)
```

   The output of this command provides information about the multicast packet rate for active sources.

   Note

   By default, the output of the **show ip mroute** command with the **active** keyword displays information about active sources sending traffic to groups at a rate greater than or equal to 4 kb/s. To display information about active sources sending low-rate traffic to groups, specify a value of 1 for the *kb/s* argument. Specifying a value of 1 for this argument displays information about active sources sending traffic to groups at a rate equal to or greater than 1 kb/s, which effectively displays information about all possible active source traffic.

Using PIM-enabled routers to test IP multicast reachability

Describes how PIM-enabled routers can test IP multicast reachability by pinging multicast groups for administrative and debugging purposes.

Using PIM-enabled routers to test IP multicast reachability is an administrative and debugging technique. It enables network administrators to verify multicast connectivity by pinging multicast groups to which all PIM-enabled routers and access servers belong. As a result, all routers respond.

Testing process

If all the PIM-enabled routers and access servers that you administer are members of a multicast group, pinging that group causes all routers to respond, which can be a useful administrative and debugging tool.

To use PIM-enabled routers to test IP multicast reachability, perform these tasks:

Configure routers to respond to multicast pings

Configure a router to respond to multicast pings on all interfaces participating in the multicast network.

This task enables routers to respond to multicast ping requests, which is essential for troubleshooting and verifying multicast network connectivity.

Perform this task on every interface of each router participating in the multicast network. This ensures proper multicast communication and allows for effective network diagnostics.

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **interface** *type number* command to enter interface configuration mode.

   Example

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

   For the *type* and *number* arguments, specify an interface that is directly connected to hosts or is facing hosts.

3. (Optional) Use the **ip igmp join-group** *group-address* command to configure an interface on the router to join the specified group.

   Example

   ```
   Device(config-if)# ip igmp join-group 225.2.2.2
   ```

   Configure the same group address for the *group-address* argument on every interface of the router participating in the multicast network.

   > **Note**
   >
   > With this method, the router accepts the multicast packets in addition to forwarding them. Accepting the multicast packets prevents the router from fast switching.
   >
   > Repeat Step 2 and Step 3 for every interface of the router involved in the multicast network.

4. Use the **end** command to return to privileged EXEC mode.

   Example

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

Ping routers configured to respond to multicast pings

Configure a ping test to routers that respond to multicast pings. This test verifies IP multicast reachability in the network.

This task is used to test IP multicast reachability in a network.

This task initiates a ping test from a router to routers that are configured to respond to multicast pings.

Before you begin

Follow these steps on a router to initiate a ping test to the routers configured to respond to multicast pings:

Procedure

1. Use the **enable** command to enter privileged EXEC mode.

   Example

   ```
   Device> enable
   ```

   Enter your password if prompted.

2. Use the **ping** *group-address* command to ping an IP multicast group address.

   Example

   ```
   Device# ping 225.2.2.2
   ```

   A successful response indicates that the group address is functioning.

Monitoring and troubleshooting PIM

Describes the tools and techniques for monitoring PIM protocol operations and troubleshooting common issues such as neighbor relationships, multicast forwarding problems, and RP connectivity failures.

Monitor PIM information

Lists privileged EXEC commands to monitor PIM configurations.

Monitor PIM configurations using privileged EXEC commands.

Table 6: PIM monitoring commands

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

Command	Purpose
show ip pim autorp	Displays global auto-RP information.
show ip pim interface	Displays information about interfaces configured for Protocol Independent Multicast (PIM).
show ip pim neighbor	Displays the PIM neighbor information.
show ip pim rp[*group-name* | *group-address*]	Displays RP routers associated with a sparse-mode multicast group.
show ip pim tunnel [**tunnel** | **verbose**]	Displays information about PIM tunnel interfaces.
show ip pim vrf { **word** { **all-vrfs** | **autorp** | **boundary** | **bsr-router** | **interface** | **mdt** | **neighbor** | **rp** | **rp-hash** | **tunnel** } }	Displays the VPN routing/forwarding instance.
show ip igmp groups detail	Displays the interested clients that have joined the specific multicast source group.

Monitor the RP mapping and BSR information

Provides commands to verify the consistency of group-to-RP mappings and monitor BSR information in privileged EXEC mode.

Monitor RP mapping and BSR information to verify the consistency of group-to-RP mappings and check BSR status in PIM networks.

Table 7: RP mapping monitoring commands

Use privileged EXEC mode to verify the consistency of group-to-RP mappings.

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.

Table 8: BSR monitoring commands

Use privileged EXEC commands to monitor BSR information.

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

Troubleshooting PIMv1 and PIMv2 interoperability problems

Describes the systematic approach to troubleshooting PIMv1 and PIMv2 interoperability issues by verifying RP mapping consistency and checking DR-RP interactions.

When debugging interoperability problems between PIMv1 and PIMv2, check the items in this order:

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

Monitor the RP mapping and BSR information

Provides commands to verify the consistency of group-to-RP mappings and monitor BSR information in privileged EXEC mode.

Monitor RP mapping and BSR information to verify the consistency of group-to-RP mappings and check BSR status in PIM networks.

Table 9: RP mapping monitoring commands

Use privileged EXEC mode to verify the consistency of group-to-RP mappings.

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.

Table 10: BSR monitoring commands

Use privileged EXEC commands to monitor BSR information.

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

Configuration Examples for PIM

Manually assign an RP to multicast groups

Provides an example of configuring the address of the RP for a specific multicast group.

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
```

Configure Auto-RP

Provides an example of configuring Auto-RP to send RP announcements out all PIM-enabled interfaces.

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

Provides a configuration example for sparse mode with Auto-RP implementation.

This section presents a configuration for sparse mode using Auto-RP.

This example configures sparse mode with Auto-RP:

```
ip multicast-routing
ip pim autorp listener
access-list 1 permit 239.254.2.0 0.0.0.255
access-list 1 permit 239.254.3.0 0.0.0.255
access-list 10 permit 224.0.1.39
access-list 10 permit 224.0.1.40
access-list 10 permit 239.254.2.0 0.0.0.255
access-list 10 permit 239.254.3.0 0.0.0.255
```

Example: define the IP multicast boundary to deny Auto-RP information

Provides an example of defining the IP multicast boundary to deny Auto-RP information.

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

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

Example: filter incoming RP announcement messages

Provides a sample configuration on an Auto-RP mapping agent that prevents candidate RP announcements from unauthorized candidate RPs.

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 only from the devices 172.16.5.1 and 172.16.2.1, and only for multicast groups in the range 224.0.0.0 to 239.255.255.255. It does not accept RP announcements from any other devices. The mapping agent also rejects announcements from 172.16.5.1 or 172.16.2.1 if the groups are in the 239.0.0.0 to 239.255.255.255 address range, which is designated as the administratively scoped range.

Example: prevent join messages to false RPs

Provides an example of preventing join messages to false RPs when all interfaces are in sparse mode.

If all interfaces operate in sparse mode, configure a default RP to support the two well-known groups 224.0.1.39 and 224.0.1.40. Auto-RP uses these groups to collect and distribute RP-mapping information. If you enable the **ip pim accept-rp auto-rp**command, you must also configure another **ip pim accept-rp** command to accept the RP.

```
Device(config)# access-list 1 permit 224.0.1.39
Device(config)# access-list 1 permit 224.0.1.40
```

Example: configure candidate BSRs

Provides an example of configuring a candidate BSR.

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

```
Device(config)# interface gigabitethernet 1/2
Device(config-if)# ip address 172.21.24.18 255.255.255.0
Device(config-if)# exit
Device(config)# ip pim bsr-candidate gigabitethernet 1/2 30 10
```

Example: configure candidate RPs

Provides an example of configuring candidate RPs

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

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

4 Configuring SSM

Topics:

- Prerequisites for configuring SSM
- Restrictions for configuring SSM
- SSM
- How to configure SSM
- Monitor SSM
- Where to go next for SSM

Prerequisites for configuring SSM

Outlines the prerequisites for configuring source-specific multicast (SSM) and SSM mapping.

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

- Before you configure SSM mapping, you must perform the following tasks:
  - Enable IP multicast routing.
  - Enable PIM sparse mode.
  - Configure SSM.
- Before you configure static SSM mapping, you must configure access control lists (ACLs) that define the group ranges to be mapped to source addresses.
- Before you can configure and use SSM mapping with DNS lookups, you need to add records to a running DNS server. If you do not already have a DNS server running, you need to install one.

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

Restrictions for configuring SSM

Outlines the restrictions and limitations when configuring Source Specific Multicast (SSM) and SSM mapping.

Be aware of these restrictions when configuring SSM:

- To run SSM with IGMPv3, SSM must be supported in the Cisco IOS router, the host where the application is running, and the application itself.
- Existing applications in a network that predate SSM will not work within the SSM range unless modified to support (S, G) channel subscriptions. Enabling SSM in a network can create problems for these applications if they use addresses within the designated SSM range.
- IGMP Snooping–IGMPv3 uses new membership report messages that might not be correctly recognized by older IGMP snooping devices.
- You must still perform some address management when using SSM with Layer 2 switching. CGMP, IGMP snooping, and RGMP support only group-specific, not (S, G) channel-specific filtering. When receivers in a switched network request different (S, G) channels that share a group, they do not benefit from these mechanisms. In this case, all receivers receive (S, G) channel traffic and must filter unwanted packets on input. Because SSM allows re-use of SSM group addresses among independent applications, decreased traffic filtering can occur. To minimize address re-use between different applications within the SSM range, assign random IP addresses for each application. For example, a service offering multiple television channels should use a different group for each television (S, G) channel. This approach ensures that receivers subscribing to different channels in the same service never encounter traffic aliasing in Layer 2 environments.
- In PIM-SSM, the last hop router will continue to periodically send (S, G) join messages if appropriate (S, G) subscriptions are on the interfaces. Therefore, as long as receivers send (S, G) subscriptions, the shortest path tree (SPT) state from the receivers to the source will be maintained, even if the source is not sending traffic for longer periods of time (or even never).

The opposite situation occurs with PIM-SM, where (S, G) state is maintained only if the source is sending traffic and receivers are joining the group. If a source stops sending traffic for more than 3 minutes in PIM-SM, the (S, G) state is deleted and only reestablished after packets from the source arrive again through the RPT (rendezvous point tree). Because PIM-SSM does not include a mechanism to notify receivers when a source is active, the network must maintain the (S, G) state in PIM-SSM while receivers continue to request the channel.

Be aware of these restrictions when configuring SSM mapping:

- The SSM Mapping feature does not provide the full benefits of SSM. SSM mapping takes a group G join from a host and links the group to an application associated with one or more sources; consequently, it supports only one application per group G. However, full SSM applications may still share a group that is 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.

SSM

Explains an extension of IP multicast where datagram traffic is forwarded to receivers from only those multicast sources that the receivers have explicitly joined.

Source-specific multicast (SSM) is an IP multicast extension that

- forwards datagram traffic to receivers from only those multicast sources that the receivers have explicitly joined,
- creates only SSM distribution trees for multicast groups configured for SSM, and
- does not create shared trees.

Additional information

To view a complete description of the SSM commands, see the IP Multicast Command Reference. To find documentation for other commands in this chapter, use the command reference master index or search online.

SSM components

Describes the components that support Source-Specific Multicast (SSM) implementation on the device.

SSM components are networking technologies that support a datagram delivery model for one-to-many applications. These components enable IP multicast solutions for audio and video broadcast environments and implement Source-Specific Multicast functionality on Cisco devices.

Supported SSM components

The device supports these components that enable SSM implementation:

- Protocol independent multicast source-specific mode (PIM-SSM): PIM-SSM is the routing protocol that supports the implementation of SSM and is derived from PIM sparse mode (PIM-SM).
- Internet Group Management Protocol version 3 (IGMPv3)

SSM and Internet Standard Multicast (ISM)

Describes two IP multicast service models and their operational differences in source discovery and traffic delivery mechanisms.

SSM and Internet Standard Multicast (ISM) are IP multicast service models that

- provide different approaches to multicast traffic delivery and source management
- use distinct signaling mechanisms for group membership and source subscription, and
- offer varying levels of network control over multicast source discovery.

Service model characteristics

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.

ISM service and group membership

The ISM service delivers IP datagrams from any source to a group of receivers, known as the multicast host group. Each multicast datagram has an arbitrary IP unicast source address (S) and uses the multicast group address (G) as the 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.

SSM channel model and IGMPv3 signaling

In SSM, delivery of datagrams is based on (S, G) channels. In both SSM and ISM, no signaling is required to become a source. However, in SSM, receivers must subscribe or unsubscribe to (S, G) channels to receive or not receive traffic from specific sources. In other words, receivers can receive traffic only from (S, G) channels to which they are subscribed, whereas in ISM, receivers need not know the IP addresses of sources from which they receive their traffic. The proposed standard for channel subscription signaling uses IGMP. It includes membership report modes supported only in IGMP version 3.

SSM IP address range

Describes the IP multicast address range that can be configured for SSM delivery model.

An SSM IP address range is a configured subset of the IP multicast group address range that applies the SSM delivery model to specific addresses, allowing SSM to coexist with the ISM service.

SSM address range configuration

Cisco IOS software allows SSM configuration for the IP multicast address range of 224.0.0.0 through 239.255.255.255. When an SSM range is defined, existing IP multicast receiver applications do not receive any traffic when they try to use an address in the SSM range (unless the application is modified to use an explicit (S, G) channel subscription).

SSM operations

Describes SSM operations in multicast networks.

SSM operations are multicast network processes that

- enable deployment in established PIM-SM networks without the need for interdomain protocols such as MSDP, Auto-RP, or bootstrap router, provided that only SSM service is needed,
- require that only last-hop routers support SSM. Other routers operate PIM-SM in the SSM range, and
- switch PIM operations to PIM-SSM mode when you configure the **ip pim ssm** global configuration command.

SSM deployment and configuration effects

An established network, in which IP multicast service is based on PIM-SM, can support SSM services. SSM can also be deployed alone in a network without the full range of protocols required for interdomain PIM-SM (for example, MSDP, Auto-RP, or bootstrap router [BSR]) if only SSM service is needed.

If SSM is deployed in a network already configured for PIM-SM, only the last-hop routers support SSM. Routers that are not directly connected to receivers do not require support for SSM. In general, these routers must run PIM-SM in the SSM range and might require extra access control configuration to suppress MSDP signaling, registration, or PIM-SM shared tree operations in the SSM range.

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

- For groups within the SSM range, (S, G) channel subscriptions are accepted through IGMPv3 include-mode membership reports.
- When operating within the SSM address range, PIM switches to PIM-SSM, which is derived from PIM-SM. In this mode, the router only generates PIM (S, G) join and prune messages. It does not generate (S, G) rendezvous point tree (RPT) or (*, G) RPT messages. The router ignores or rejects incoming RPT-related messages. It sends immediate register-stop responses to incoming PIM register messages.

  PIM-SSM is backward compatible with PIM-SM, except on last-hop routers. Routers that are not last-hop routers can continue to run PIM-SM for SSM groups, such as when they do not support SSM.
- No MSDP source-active (SA) messages within the SSM range are accepted, generated, or forwarded.

SSM mapping

Describes a mechanism that enables routers to translate IGMPv1 or IGMPv2 membership reports into IGMPv3 source-specific memberships for multicast groups.

SSM mapping is a multicast translation mechanism that

- translates IGMPv1 or IGMPv2 membership reports into IGMPv3 channel memberships for well-known sources,
- enables last hop routers to determine source addresses through statically configured tables or DNS servers, and
- allows routers to handle source-specific multicast in environments with legacy IGMP versions.

SSM mapping operation

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

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

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

SSM mapping enables the last hop router to determine the source addresses either by a statically configured table on the router or through a DNS server. When the statically configured table or the DNS mapping changes, the router leaves the current sources associated with the joined groups.

Static SSM mapping

Explains static SSM mapping configuration for determining multicast sources.

Static SSM mapping is a multicast configuration method. It enables the last hop router to use a static map to determine the sources sending to groups. This method requires configuring ACLs to define group ranges. The **ip igmp ssm-map static** global configuration command maps groups to sources.

Static SSM mapping use cases

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

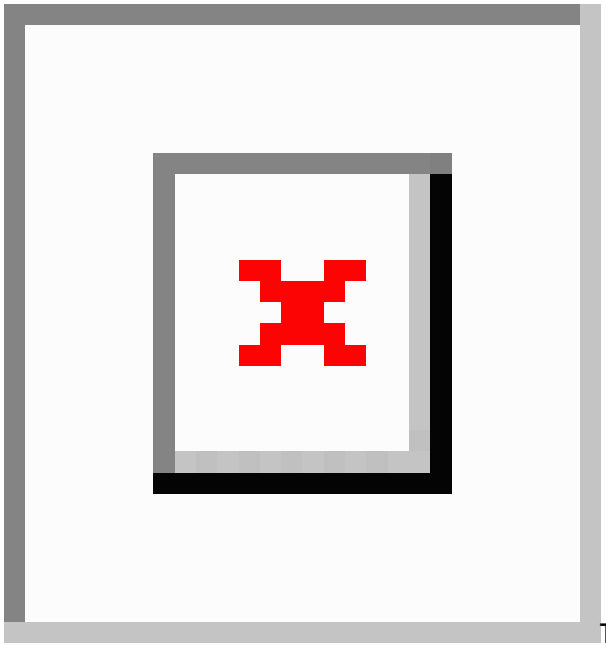
Describes a method that enables last hop routers to perform reverse DNS lookups to determine sources sending to groups.

DNS-based SSM mapping is a configuration method that enables the last hop router to perform a reverse DNS lookup to determine sources sending to groups. It constructs a domain name that includes the group address and performs a reverse lookup into the DNS. The method supports up to 20 sources for each group, and the router joins all sources configured for a group.

SSM mapping mechanism benefits

The SSM mapping mechanism that enables the last hop router to join multiple sources for a group can provide source redundancy for a TV broadcast. In this context, the last hop router provides redundancy using SSM mapping to simultaneously join two video sources for the same TV channel. However, to prevent the last hop router from duplicating the video traffic, the video sources must use a server-side switchover mechanism. One video source is active. The other video source acts as a backup and remains 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.

Figure 8: DNS-Based SSM mapping



The figure displays DNS-based SSM mapping.

DNS records configuration

To look up one or more source addresses for a group that includes G1, G2, G3, and G4, you must configure these DNS records on the DNS server:

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

See your DNS server documentation for more information about configuring DNS resource records.

How to configure SSM

Describes how to configure Source Specific Multicast (SSM) on Cisco devices to enable efficient multicast traffic delivery by allowing receivers to specify both the multicast group and source addresses.

Configure SSM

Configure SSM to enable Source Specific Multicast functionality on your network device.

Configure Source Specific Multicast (SSM) to enable efficient multicast delivery where receivers explicitly specify both the multicast group and the source from which they want to receive traffic.

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 first, then reference the access list in the **ip pim ssm** command.

Follow these steps to configure SSM:

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **ip pim ssm** [**default** | **range** *access-list*] command to define the SSM range of IP multicast addresses.

   Example

   ```
   Device(config)# ip pim ssm range 20
   ```

3. Use the **interface** *type number* command to select an interface that is connected to hosts on which IGMPv3 can be enabled and enter interface configuration mode.

   Example

   ```
   Device(config)# interface gigabitethernet 1/3
   ```

4. Use the **ip pim sparse-mode** command to enable PIM on an interface.

Example

```
Device(config-if)# ip pim sparse-mode
```

5. Use the **ip igmp version 3** command to enable IGMPv3 on this interface.

Example

```
Device(config-if)# ip igmp version 3
```

6. Use the **end** command to return to privileged EXEC mode.

Example

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

7. Use the **show running-config** command to verify your entries.

Example

```
Device# show running-config
```

Configuring source specific multicast mapping

Describes how to configure Source Specific Multicast (SSM) mapping to support SSM transition for legacy systems that cannot support IGMPv3, enabling SSM benefits for video delivery and applications without IGMPv3 host stack support.

he Source Specific Multicast (SSM) mapping feature enables SSM transition when the end system cannot support SSM for administrative or technical reasons. You can use SSM mapping to leverage SSM for video delivery to legacy STBs that do not support IGMPv3 or for applications that do not use the IGMPv3 host stack.

Configure static SSM mapping

Configure static SSM mapping to enable Source-Specific Multicast functionality on your device.

Static SSM mapping allows you to map multicast groups to specific source addresses, enabling proper handling of IGMPv1 and IGMPv2 membership reports in SSM deployments.

SSM mapping provides a mechanism to handle legacy IGMP versions in SSM environments by associating multicast groups with their corresponding source addresses through static configuration or DNS queries.

Before you begin

Follow these steps to configure static SSM mapping:

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

Example

```
Device# configure terminal
```

2. Use the **ip igmp ssm-map enable** command to enable SSM mapping for groups in the configured SSM range.

Example

```
Device(config)# ip igmp ssm-map enable
```

Note

By default, this command enables DNS-based SSM mapping.

3. (Optional) Use the **no ip igmp ssm-map query dns** command to disable DNS-based SSM mapping.

Example

```
Device(config)# no ip igmp ssm-map query dns
```

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.

4. Use the **ip igmp ssm-map static** *access-list source-address* command to configure static SSM mapping.

Example

```
Device(config)# ip igmp ssm-map static 11 172.16.8.11
```

- 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 the router receives an IGMPv1 or IGMPv2 membership report for a group in the SSM range and additional SSM mappings are configured, the device determines the associated source addresses for the group by examining 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 you need more than one mapping.

5. Use the **end** command to return to privileged EXEC mode.

Example

```
Device(config)# end
```

6. Use the **show running-config** command to verify your entries.

Example

```
Device# show running-config
```

Configure DNS-based SSM mapping

Configure DNS-based SSM mapping to enable multicast source discovery through DNS queries.

DNS-based SSM mapping allows routers to discover multicast sources through DNS queries, providing an automated method for source-specific multicast operations.

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

Before you begin

Follow these steps to configure DNS-based SSM mapping:

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **ip igmp ssm-map enable** command to enable SSM mapping for groups in a configured SSM range.

   Example

   ```
   Device(config)# ip igmp ssm-map enable
   ```

3. (Optional) Use the **ip igmp ssm-map query dns** command to enable DNS-based SSM mapping.

   Example

   ```
   Device(config)# ip igmp ssm-map query dns
   ```

   - 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.

4. (Optional) Use the **ip domain multicast** *domain-prefix* command to change the domain prefix used for DNS-based SSM mapping.

Example

```
Device(config)# ip domain multicast ssm-map.cisco.com
```

- By default, the software uses the ip-addr.arpa domain prefix.

5. Use the **ip name-server** *server-address1* [*server-address2*...*server-address6*] command to specify one or more name servers to use for name and address resolution.

   Example

   ```
   Device(config)# ip name-server 10.48.81.21
   ```

   If you require additional DNS servers for redundancy, repeat the configuration for each server.

6. Use the **end** command to return to privileged EXEC mode.

   Example

   ```
   Device(config)# end
   ```

7. Use the **show running-config** command to verify your entries.

   Example

   ```
   Device# show running-config
   ```

Configure static traffic forwarding with SSM mapping

Configure static traffic forwarding with SSM mapping on the last hop router.

This task configures SSM mapping to statically forward multicast traffic for certain groups using DNS-based SSM mapping to determine the source addresses of the channels.

Use this procedure when you need to configure static forwarding of SSM traffic on a last hop router. Static forwarding of traffic with SSM mapping works using DNS-based SSM mapping or statically configured SSM mapping.

Before you begin

The interface must be either a routed port (configured with **no switchport**) or an SVI (created with **interface vlan***VLAN-id*). These interfaces must have IP addresses assigned to them.

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

Procedure

---

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **interface** *interface-id* command to select an interface on which to statically forward traffic for a multicast group using SSM mapping and enter interface configuration mode.

Example

```
Device(config)# interface gigabitethernet 1/3
```

3. Use the **ip igmp static-group** *group-address* **source ssm-map** command to configure SSM mapping to statically forward an (S, G) channel from the interface.

   Example

   ```
   Device(config-if)# ip igmp static-group 239.1.2.1 source ssm-map
   ```

   Use this command if you want to statically forward SSM traffic for certain groups. DNS-based SSM mapping determines the source addresses of the channels.

4. Use the **end** command to return to privileged EXEC mode.

   Example

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

5. Use the **show running-config** command to verify your entries.

   Example

   ```
   Device# show running-config
   ```

Monitor SSM

Lists commands for monitoring Source Specific Multicast (SSM) functionality.

Monitor SSM operations using privileged EXEC commands to display IGMPv3 channel subscriptions and multicast group information.

Table 11: Commands for monitoring SSM

Use the privileged EXEC commands in this table to monitor 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.

Monitor SSM mapping

Lists the privileged EXEC commands to monitor SSM mapping.

This table lists commands you can use to monitor SSM mapping.

Table 12: SSM mapping monitoring commands

Command	Purpose
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.

Where to go next for SSM

Describes the next configuration steps available for SSM, including IGMP, PIM, IP Multicast Routing, Service Discovery Gateway, and other related multicast features you can set up after implementing SSM.

You can configure the following:

- IGMP
- PIM
- IP Multicast Routing
- Service Discovery Gateway

5 Implementing IPv6 Multicast

Topics:

- IPv6 multicast routing
- Configuring a BSR
- Implementing IPv6 multicast

IPv6 multicast routing

Describes how to implement IPv6 multicast routing on the switch.

IPv6 multicast routing is a network communication scheme that allows a host to send a single data stream to a subset of all hosts (group transmission) simultaneously. It provides a third transmission method beyond unicast (single host) and broadcast (all hosts) communication. This approach enables efficient data distribution to multiple recipients using the IPv6 protocol.

IPv6 multicast transmission types

Traditional IP communication supports different transmission schemes:

- Unicast transmission: allows a host to send packets to a single host
- Broadcast transmission: allows a host to send packets to all hosts
- Multicast transmission: allows a host to send a single data stream to a subset of all hosts simultaneously

IPv6 multicast

Describes how IPv6 multicast enables efficient data delivery to multiple receivers through group-based communication across networks.

IPv6 multicast is a network communication method that

- delivers data to a potentially unlimited number of receivers using only one copy of the multicast data on each subnet,
- uses multicast group addresses to identify packets for group members, and
- enables hosts to join and leave groups dynamically through the Multicast Listener Discovery (MLD) protocol.

IPv6 multicast characteristics

An IPv6 multicast group is an arbitrary group of receivers that want to receive a particular data stream. This group has no physical or geographical boundaries-receivers can be located anywhere on the Internet or in any private network.

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

Key characteristics of IPv6 multicast:

- 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 level, duration, and membership of a multicast group can all vary according to the group and over time.
- A group that has members may have no activity

Receivers that are interested in receiving data flowing to a particular group must join the group by signaling their local switch. This signaling is achieved with the Multicast Listener Discovery (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.

Senders select a multicast address for the group's receivers and use it as the destination address in datagrams sent to members. Multicast packets are delivered to a group using best-effort reliability, just like IPv6 unicast packets.

> **Note**
>
> Multicast is not supported on port-channels.

IPv6 multicast routing implementation

Describes the protocols that implement IPv6 multicast routing in Cisco IOS-XE software.

IPv6 multicast routing implementation is a network protocol framework. It uses MLD for discovering multicast listeners on directly attached links, employs PIM-SM between switches for tracking multicast packet forwarding, and supports PIM-SSM for source-specific multicast operations.

Supported protocols

The Cisco IOS-XE software supports these 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.

  There are two versions of MLD: MLD version 1 is based on version 2 of the Internet Group Management Protocol (IGMP) for IPv4, and MLD version 2 is based on version 3 of the IGMP for IPv4.

  IPv6 multicast for Cisco IOS software uses both MLD version 2 and MLD version 1. MLD version 2 is fully backward-compatible with MLD version 1 (described in RFC 2710).

  Hosts that support only MLD version 1 will interoperate with a switch running MLD version 2. Mixed LANs with both MLD version 1 and MLD version 2 hosts are likewise supported.
- PIM-SM is used between switches so that they can track which multicast packets to forward to each other and to their directly connected LANs.
- PIM in Source Specific Multicast (PIM-SSM) is similar to PIM-SM. It has the additional ability to report interest in receiving packets from specific source addresses, or from all except specific source addresses, to an IP multicast address.

IPv6 multicast listener discovery protocol

Describes the protocol used by IPv6 switches to discover multicast listeners and control multicast traffic flow in campus networks.

The IPv6 Multicast Listener Discovery (MLD) Protocol is a network protocol. It discovers the presence of multicast listeners on directly attached links. It identifies which multicast addresses are of interest to neighboring nodes, and it automatically controls and limits multicast traffic flow throughout networks using special multicast queriers and hosts.

MLD protocol implementation

To start implementing multicasting in the campus network, users must first define who receives the multicast. The MLD protocol is used by IPv6 switches to discover the presence of multicast listeners on their directly attached links. For example, nodes that want to receive multicast packets. It also discovers which multicast addresses are of interest to those neighboring nodes. It is used for discovering local group and source-specific group membership.

Multicast queriers and hosts

Describes multicast queriers and hosts as network devices that participate in multicast group communication using MLD protocols.

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

A multicast host is a receiver that reports host memberships.

Multicast group characteristics

A set of queriers and hosts is called a multicast group. Queriers and hosts use MLD reports to join and leave multicast groups and to begin receiving group traffic.

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

MLD access group

Describes receiver access control in IPv6 multicast switches.

An MLD access group restricts the list of multicast groups a receiver can join and regulates sources used to join SSM channels in IPv6 multicast switches.

Explicit tracking of receivers

Explains the explicit tracking feature that allows a switch to track the behavior of hosts within its IPv6 network and enables the fast leave mechanism to be used with MLD version 2 host reports.

Explicit tracking of receivers is a switch feature that 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.

Protocol independent multicast

Describes a multicast routing protocol that enables switches to track and forward multicast packets independently of unicast routing protocols.

Protocol Independent Multicast (PIM) is a multicast routing protocol that enables switches to track which multicast packets to forward to each other and to their directly connected LANs. It works independently of the unicast routing protocol to send or receive multicast route updates. It uses the existing unicast table content to perform the Reverse Path Forwarding (RPF) check instead of building and maintaining its own separate routing table.

Protocol capabilities

You can configure IPv6 multicast to use PIM-SM operation.

PIM-Sparse mode

Describes an intradomain multicast routing protocol that uses unicast routing to provide reverse-path information for multicast tree building.

PIM-Sparse Mode is an IPv6 multicast routing protocol that

- uses unicast routing to provide reverse-path information for multicast tree building without depending on any particular unicast routing protocol,
- operates 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, and
- distributes information about active sources by forwarding data packets on the shared tree and initially uses shared trees that require the use of a Rendezvous Point (RP).

PIM-SM operation details

PIM-SM uses PIM joins sent hop by hop toward the root node of the tree to request multicast traffic. In a shared tree, the root node is the RP. In a shortest path tree (SPT), the root node is the first-hop switch directly connected to the multicast source. The RP keeps track of multicast groups. The host's first-hop switch registers hosts that send multicast packets with the RP.

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. They are replicated wherever the RP tree branches and eventually reach 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

Describes the Bootstrap Router protocol mechanism that dynamically distributes group-to-RP mapping information throughout a PIM-SM domain for IPv6 multicast networks.

IPv6 BSR RP mapping is a dynamic protocol mechanism that

- enables PIM switches in a domain to map each multicast group to the correct RP address using the Bootstrap Router protocol
- provides adaptive distribution of group-to-RP mapping information rapidly throughout a domain, and
- detects unreachable RPs and modifies mapping tables to ensure continued multicast functionality.

RP mapping requirements

Every PIM-SM multicast group needs to be associated with the IP or IPv6 address of an RP.

When a new multicast sender starts sending, its local DR will encapsulate these data packets in a PIM register message and send them to the RP for that multicast group.

When a new multicast receiver joins, its local DR will send a PIM join message to the RP for that multicast group. When any PIM switch sends a (*, G) join message, the PIM switch needs to know which is the next switch toward the RP so that G (Group) can send a message to that switch.

Also, when a PIM switch is forwarding data packets using (*, G) state, the PIM switch must also determine the correct incoming interface for packets destined for G, ensuring that it rejects any packets arriving on other interfaces.

BSR operation involves these components:

- Candidate bootstrap switches (C-BSRs): A small set of switches from a domain configured as candidates, with a single BSR selected for that domain
- Candidate RPs (C-RPs): A set of switches within a domain configured as candidate RPs, typically the same switches configured as C-BSRs
- C-RP-Adv messages: Periodic unicast messages sent by candidate RPs to the BSR, advertising their willingness to be an RP with optional group addresses and mask length fields
- Bootstrap messages (BSMs): Periodic messages originated by the BSR that include C-RPs and their corresponding group prefixes, distributed hop-by-hop throughout the domain

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

Routable address hello option

Describes a PIM hello message option that includes all addresses on an interface to help detect upstream switch addresses when a switch has multiple addresses on a link.

A routable address hello option is a PIM hello message option that

- includes all addresses on the interface on which the PIM hello message is advertised,
- allows the PIM protocol to avoid situations where the address of a PIM neighbor is not the same as the address of the next-hop switch, and
- enables proper upstream switch detection when a switch has multiple addresses on a link.

Routable address hello option operation

An IPv6 interior gateway protocol is used to build the unicast routing table. A specific procedure detects the upstream switch address. This procedure assumes that the address of a Protocol Independent Multicast (PIM) neighbor is always identical to the address of the next-hop switch. However, this assumption is valid only when both addresses refer to the same switch.

A typical situation occurs when the address of an RP shares a subnet prefix with downstream switches. The RP switch address must be domain-wide and, therefore, cannot be a link-local address.

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. This option includes all possible addresses of a PIM switch on the link, so it always contains the RPF calculation result if it refers to the PIM switch that supports this option.

Due to size restrictions on PIM messages and the requirement that the routable address hello option fit within a single PIM hello message, only 16 addresses can be configured on the interface.

Rendezvous point

Describes a single common root in IPv6 PIM sparse mode that serves AS the distribution point for multicast traffic.

A rendezvous point is a network device that

- serves as a single common root at a chosen point of a shared distribution tree,
- receives multicast data from directly connected sources through PIM-designated routers for distribution along the shared tree, and
- is statically configured in each device and can serve as an RP for multiple groups.

RP operation and configuration details

IPv6 PIM provides embedded RP support. Embedded RP support enables the device to obtain RP information from the multicast group destination address, rather than from a statically configured RP. For devices that are the RP, the device must be statically configured as the RP.

How the device learns embedded RP addresses

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

Static RP vs. embedded RP selection

To select a static RP over an embedded RP, the specific embedded RP group range or mask must be configured in the access list of the static RP. When PIM is configured in sparse mode, you must also choose one or more devices to operate as an RP.

> **Note**
>
> The **ipv6 PIM RP embedded** command is enabled by default.

Data forwarding mechanics by PIM DRs

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

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

Functions and constraints of the RP address

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.

PIM accept register filtering

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

> **Note**
>
> Dynamic RPs have higher preference over Static RP's

Static mroutes

Explains how IPv6 static mroutes behave similarly to IPv4 static mroutes to influence RPF checks.

A static mroute is a multicast routing configuration that

- influences the RPF check in IPv6 networks similar to IPv4 static mroutes,
- shares the same database as IPv6 static routes, and
- extends static route support for RPF checks.

Static mroute features

Static mroutes provide these capabilities:

- Support equal-cost multipath mroutes
- Support unicast-only static routes

MRIB

Describes the Multicast Routing Information Base (MRIB) as a protocol-independent repository for multicast routing entries.

The Multicast Routing Information Base (MRIB) is a protocol-independent repository that

- stores multicast routing entries instantiated by multicast routing protocols,
- provides independence between routing protocols and the Multicast Forwarding Information Base (MFIB), and
- acts as a coordination and communication point among its clients.

MRIB client services and coordination

Routing clients use the services provided by the MRIB to instantiate routing entries and retrieve changes made to routing entries by other clients. In addition to routing clients, MRIB has forwarding clients (MFIB instances) and special clients such as MLD.

MFIB retrieves its forwarding entries from MRIB and notifies the MRIB of any events related to packet reception. Routing clients can explicitly request these notifications, or MFIB can generate them spontaneously.

MRIB allows for the coordination of multiple routing clients in establishing multicast connectivity within the same multicast session. MRIB also allows for the coordination between MLD and routing protocols.

MFIB

Describes a platform-independent and routing-protocol-independent library that provides IPv6 multicast forwarding capabilities.

The MFIB is a platform-independent and routing-protocol-independent library that

- provides a Cisco IOS-XE platform with an interface to read the IPv6 multicast forwarding table and receive notifications when the forwarding table changes,
- provides information with clearly defined forwarding semantics designed to make it easy for the platform to translate to its specific hardware or software forwarding mechanisms, and
- maintains next-hop address information based on the information in the IPv6 routing table.

MFIB operation and characteristics

When routing or topology changes occur in the network, the IPv6 routing table is updated, and those changes are reflected in the MFIB. 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.

> **Note**
>
> The maximum transmission unit (MTU) for V6 PIM register tunnels created by MFIB is 1452 bytes. As a result, you cannot use jumbo frames over a V6 PIM tunnel.

Distributed MFIB

Describes a multicast forwarding mechanism used on distributed platforms for IPv6 packet switching.

Distributed MFIB (dMFIB) is a multicast forwarding mechanism that

- switches multicast IPv6 packets on distributed platforms,
- contains platform-specific information on replication across line cards, and
- implements core forwarding logic common to all forwarding environments.

Functions

dMFIB implements these functions:

- Relays data-driven protocol events generated in the line cards to PIM.
- Provides an MFIB platform application program interface (API) to propagate MFIB changes to platform-specific code responsible for programming the hardware acceleration engine. The API also includes entry points to switch a packet in software (when the packet triggers a data-driven event). Additionally, it enables uploading traffic statistics to the software.

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

IPv6 multicast process switching and fast switching

Describes the unified MFIB approach for providing both fast switching and process switching support for PIM-SM and PIM-SSM in IPv6 multicast.

IPv6 multicast process switching and fast switching are packet forwarding methods that

- use a unified MFIB to provide support for PIM-SM and PIM-SSM in IPv6 multicast,
- offer different levels of performance and scalability for IPv6 packet forwarding, and
- utilize optimized data structures and precomputed information to improve forwarding efficiency.

Process switching and fast switching characteristics

Process switching requires the Route Processor IOS daemon to 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 RP IOSd also computes the cyclic redundancy check (CRC). This switching method is the least scalable method for switching IPv6 packets.

IPv6 multicast fast switching

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

Role of the MFIB and adjacency tables

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.

Population of the adjacency table

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

Multipath loading and redundancy mechanisms

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

Configuring a BSR

Describes how to configure the Bootstrap Router (BSR), which distributes group-to-RP mapping information rapidly throughout a domain.

Follow these steps to configure the Bootstrap Router (BSR). The BSR distributes group-to-RP mapping information rapidly throughout a domain.

Configure a BSR and verify BSR information

Configure a Bootstrap Router (BSR) for IPv6 PIM and verify BSR information.

This task configures a switch as a candidate Bootstrap Router (BSR) for IPv6 Protocol Independent Multicast (PIM) and verifies the BSR information to ensure proper configuration.

Use this procedure to set up a BSR for IPv6 PIM multicast routing. Verify that the BSR is functioning correctly.

Before you begin

To configure and verify BSR information, perform these steps:

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **ipv6 pim bsr candidate bsr ipv6-address [hash-mask-length] [priority priority-value]** command to configure a switch to be a candidate BSR.

Example

```
Device(config)# ipv6 pim bsr candidate bsr 2001:db8:3000:3000::42 124 priority
 10
```

3. Use the **interface** *type number* command to specify an interface type and number and enter interface configuration mode.

   Example

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

4. Use the **ipv6 pim bsr border** command to configure a border for all bootstrap message (BSMs) of any scope on a specified interface.

   Example

   ```
   Device(config-if)# ipv6 pim bsr border
   ```

5. Use the **end** command to return to privileged EXEC mode.

   Example

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

6. Use the **show ipv6 pim bsr {election | rp-cache | candidate-rp}** command to display information related to PIM BSR protocol processing.

   Example

   ```
   Device# show ipv6 pim bsr election
   ```

Send PIM RP advertisements to the BSR

Configure PIM RP advertisements to send to the BSR for IPv6 multicast routing.

This task configures PIM RP (Rendezvous Point) advertisements to be sent to the BSR (Bootstrap Router) for IPv6 multicast routing.

Use this procedure when you need to configure a device to send PIM RP advertisements to the BSR in an IPv6 multicast network.

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **ipv6 pim bsr candidate rp ipv6-address [group-list access-list-name] [priority priority-value] [interval seconds]** command to send PIM RP advertisements to the BSR.

Example

```
Device(config)# ipv6 pim bsr candidate rp 2001:db8:3000:3000::42 priority 0
```

3. Use the **interface** *type number* command to specify an interface type and number and enter interface configuration mode.

   Example

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

4. Use the **ipv6 pim bsr border** command to configure a border for all BSMs of any scope on a specified interface.

   Example

   ```
   Device(config-if)# ipv6 pim bsr border
   ```

Configure BSR for use within scoped zones

Configure BSR for use within scoped zones to enable multicast routing within specific network boundaries.

Configure the Bootstrap Router (BSR) so that it operates within scoped zones. This allows you to manage multicast routing control and network boundaries.

When you configure BSR within scoped zones, you can control multicast traffic boundaries and manage Rendezvous Point (RP) advertisements within specific network segments.

Before you begin

Follow these steps to configure BSR for use within scoped zones:

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **ipv6 pim bsr candidate rp ipv6-address [hash-mask-length] [priority priority-value]** command to configure a switch to be a candidate BSR.

   Example

   ```
   Device(config)# ipv6 pim bsr candidate bsr 2001:db8:1:1:4
   ```

3. Use the **ipv6 pim bsr candidate rp ipv6-address [group-list access-list-name] [priority priority-value] [interval seconds]** command to configure the candidate RP to send PIM RP advertisements to the BSR.

Example

```
Device(config)# ipv6 pim bsr candidate rp 2001:db8:1:1:1 group-list list scope
 6
```

4. Use the **interface** *type number* command to specify an interface type and number and enter interface configuration mode.

   Example

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

5. Use the **ipv6 multicast boundary scope scope-value** command to configure a multicast boundary on the interface for a specified scope.

   Example

   ```
   Device(config-if)# ipv6 multicast boundary scope 6
   ```

Configure BSR switches to announce scope-to-RP mappings

Configure IPv6 BSR switches to announce scope-to-RP mappings directly instead of learning them from candidate-RP messages.

Configure a BSR switch to announce scope-to-RP mappings so that an RP that does not support BSR is imported into the BSR. This feature also enables a known remote RP on the local candidate BSR switch to learn about an RP positioned outside the enterprise's BSR domain.

IPv6 BSR switches can be statically configured to announce scope-to-RP mappings directly instead of learning them from candidate-RP messages. You might want to configure a BSR switch to announce scope-to-RP mappings to import an RP that does not support BSR into the BSR domain. Enabling this feature allows the known remote RP on the local candidate BSR switch to learn about an RP positioned outside the enterprise’s BSR domain.

To configure BSR switches to announce scope-to-RP mappings, use these steps:

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **ipv6 pim bsr announced rp ipv6-address [group-list access-list-name] [priority priority-value]** command to announce scope-to-RP mappings directly from the BSR for the specified candidate RP.

   Example

   ```
   Device(config)# ipv6 pim bsr announced rp 2001:db8:3000:3000::42 priority 0
   ```

Implementing IPv6 multicast

Describes the process of configuring and deploying IPv6 multicast networking to enable efficient one-to-many communication across IPv6 networks, including protocol setup and routing considerations.

Enable IPv6 multicast routing

Configure IPv6 multicast routing to enable multicast forwarding for PIM and MLD on all enabled interfaces of the switch.

Enable IPv6 multicast routing to support multicast forwarding for PIM and MLD protocols on all IPv6-enabled interfaces of the switch.

IPv6 multicast routing allows the switch to forward multicast traffic and participate in multicast distribution trees using Protocol Independent Multicast (PIM) and Multicast Listener Discovery (MLD) protocols.

To enable IPv6 multicast routing, perform these steps:

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **ipv6 multicast-routing** command to enable multicast routing on all IPv6-enabled interfaces and enable multicast forwarding for PIM and MLD on all enabled interfaces of the switch.

   Example

   ```
   Device(config)# ipv6 multicast-routing
   ```

Customizing and verifying the MLD protocol

Describes how to customize Multicast Listener Discovery (MLD) protocol settings and verify proper operation on IPv6 networks to ensure efficient multicast traffic management and troubleshoot connectivity issues.

Customizing and verifying MLD on an interface

Configure MLD settings on an interface to customize multicast group behavior and verify the configuration.

This task enables you to customize MLD (Multicast Listener Discovery) settings on a specific interface to control multicast group membership and verify the configuration is working correctly.

MLD allows IPv6 multicast routers to discover multicast listeners on directly attached links. Customizing MLD settings helps control multicast traffic behavior and optimize network performance.

To customize and verify MLD on an interface, complete these steps:

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

Example

```
Device# configure terminal
```

2. Use the **interface** *type number* command to specify an interface type and number and enter interface configuration mode.

   Example

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

3. Use the **ipv6 mld join-group [group-address] [include | exclude] {source-address | source-list [acl]}** command to configure MLD reporting for a specified group and source.

   Example

   ```
   Device(config-if)# ipv6 mld join-group ff04::10
   ```

   Note

   join-group and static-group should be used only in a debugging environment, as the packets matching this group will reach CPU for processing.

4. Use the **ipv6 mld access-group access-list-name** command to allow the user to perform IPv6 multicast receiver access control.

   Example

   ```
   Device(config-if)# ipv6 access-list acc-grp-1
   ```

5. Use the **ipv6 mld static-group [group-address] [include | exclude] {source-address | source-list [acl]}** command to statically forward traffic for the multicast group onto a specified interface. This command causes the interface to behave as if an MLD joiner were present.

   Example

   ```
   Device(config-if)# ipv6 mld static-group ff04::10 include 100::1
   ```

6. Use the **ipv6 mld query-max-response-time seconds** command to configure the timeout value before the switch takes over as the querier for the interface.

   Example

   ```
   Device(config-if)# ipv6 mld query-timeout 130
   ```

7. Use the **exit** command to return to privileged EXEC mode.

   Example

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

Enter this command twice to exit interface configuration mode and enter privileged EXEC mode.

8. Use the **show** commands to verify the configuration.
   a) Use the **show ipv6 mld groups [link-local] [group-name | group-address] [interface-type interface-number] [detail | explicit]** command to display the multicast groups that are directly connected to the switch and that were learned through MLD.

      Example

      ```
      Device# show ipv6 mld groups gigabitethernet 1/0/1
      ```

   b) Use the **show ipv6 mld groups summary** command to display the number of (*, G) and (S, G) membership reports present in the MLD cache.

      Example

      ```
      Device# show ipv6 mld groups summary
      ```

   c) Use the **show ipv6 mld interface [type number]** command to display multicast-related information about an interface.

      Example

      ```
      Device# show ipv6 mld interface gigabitethernet 1/0/1
      ```

MLD group limits

Describes the independent operation of per-interface and global MLD limits on network switches.

MLD group limits are network configuration constraints that operate independently at both per-interface and global levels. Users must configure these limits manually, since no default values are set. If membership reports exceed either limit, those reports are ignored.

MLD group limits configuration behavior

Both per-interface and global MLD limits can be configured on the same switch. By default, the system does not set any global or per-interface MLD limits. Users must manually configure the desired limits.

Implement MLD group limits globally

Configure global MLD state limits to control the number of MLD states on the device.

This task allows you to limit the number of MLD (Multicast Listener Discovery) states globally on the device to control multicast group membership and resource usage.

MLD group limits help prevent excessive multicast group memberships that could impact device performance. Global limits provide network-wide control over MLD state resources.

Follow these steps to implement MLD group limits globally:

Procedure

---

1. Use the **configure terminal** command to enter global configuration mode.

Example

```
Device# configure terminal
```

2. Use the **ipv6 mld state-limit number** command to limit the number of MLD states globally.

Example

```
Device(config)# ipv6 mld state-limit 300
```

Implementing MLD group limits per interface

Configure MLD group limits per interface to control the number of MLD states allowed on each interface.

Implement MLD group limits per interface to control multicast traffic and prevent excessive MLD states from consuming device resources.

MLD group limits help manage multicast traffic by restricting the number of MLD states that can be established on a specific interface. This prevents potential resource exhaustion and ensures optimal network performance.

Follow these steps to implement MLD group limits per interface:

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

Example

```
Device# configure terminal
```

2. Use the **interface** *type number* command to specify an interface type and number and enter interface configuration mode.

Example

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

3. Use the **ipv6 mld limit number [except] access-list** command to limit the number of MLD states on a per-interface basis.

Example

```
Device(config-if)# ipv6 mld limit 100
```

Configure explicit tracking of receivers to track host behavior

Configure explicit tracking of receivers to enable switches to track host behavior within IPv6 networks and use fast leave mechanisms with MLD version 2 host reports.

Enable a switch to track the behavior of hosts within its IPv6 network. Allow the fast leave mechanism to work with MLD version 2 host reports.

The explicit tracking feature allows a switch to monitor the behavior of hosts within its IPv6 network. It also enables the fast leave mechanism for use with MLD version 2 host reports.

Follow these steps to configure explicit tracking of receivers to track host behavior:

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **interface** *type number* command to specify an interface type and number and enter interface configuration mode.

   Example

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

3. Use the **ipv6 mld explicit-tracking access-list-name** command to enable explicit tracking of hosts.

   Example

   ```
   Device(config-if)# ipv6 mld explicit-tracking list1
   ```

Reset the MLD traffic counters

Configure the device to reset all MLD traffic counters and verify the results.

Reset the MLD (Multicast Listener Discovery) traffic counters to clear accumulated statistics and provide a fresh baseline for monitoring MLD traffic performance.

MLD traffic counters accumulate statistics over time and may need to be reset for troubleshooting or monitoring purposes. This procedure clears all MLD traffic counters and allows you to verify the reset operation.

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **clear ipv6 mld traffic** command to reset all MLD traffic counters.

   Example

   ```
   Device# clear ipv6 mld traffic
   ```

3. Use the **show ipv6 mld traffic** command to display the MLD traffic counters.

Example

```
Device# show ipv6 mld traffic
```

Clear the MLD interface counters

Configure the system to clear MLD interface counters for network troubleshooting and monitoring.

This task clears the MLD interface counters to reset monitoring statistics and troubleshoot network issues.

Use this procedure when you need to reset MLD interface counters for monitoring or troubleshooting purposes.

Before you begin

Follow these steps to clear the MLD interface counters:

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **clear ipv6 mld counters interface-type** command to clear the MLD interface counters.

   Example

   ```
   Device# clear ipv6 mld counters ethernet1/0
   ```

Configuring PIM

Explains how to configure Protocol Independent Multicast (PIM) on network devices, including the necessary commands and procedures for setting up multicast routing functionality.

This section explains how to configure PIM.

Configure PIM-SM and display PIM-SM information for a group range

Configure PIM-SM rendezvous point addressing for IPv6 multicast groups and display PIM-SM information using various show and debug commands.

This task configures Protocol Independent Multicast Sparse Mode (PIM-SM) settings for IPv6 multicast group management and provides commands to view PIM-SM operational information and troubleshoot PIM protocol activity.

PIM-SM is used in IPv6 multicast environments to efficiently manage multicast traffic by designating rendezvous points (RPs) for specific group ranges. This configuration is essential for proper multicast routing and group membership management in IPv6 networks.

To configure PIM-SM and view PIM-SM information for a group range, use the steps provided.

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **ipv6 pim rp-address ipv6-address [group-access-list]** command to configure the address of a PIM RP for a particular group range.

   Example

   ```
   Device(config)# ipv6 pim rp-address 2001:db8::01:800:200e:8c6c acc-grp-1
   ```

3. Use the **exit** command to return to privileged EXEC mode.

   Example

   ```
   Device(config)# exit
   ```

4. Use the **show ipv6 pim interface [state-on] [state-off] [type-number]** command to display information about interfaces configured for PIM.

   Example

   ```
   Device# show ipv6 pim interface
   ```

5. Use the **show ipv6 pim group-map [group-name | group-address] | [group-range | group-mask] [info-source {bsr | default | embedded-rp | static}]** command to display an IPv6 multicast group mapping table.

   Example

   ```
   Device# show ipv6 pim group-map
   ```

6. Use the **show ipv6 pim neighbor [detail] [interface-type interface-number | count]** command to display the PIM neighbors discovered by the Cisco IOS software.

   Example

   ```
   Device# show ipv6 pim neighbor
   ```

7. Use the **show ipv6 pim range-list [config] [rp-address | rp-name]** command to display information about IPv6 multicast range lists.

   Example

   ```
   Device# show ipv6 pim range-list
   ```

8. Use the **show ipv6 pim tunnel [interface-type interface-number]** command to display information about the PIM register encapsulation and de-encapsulation tunnels on an interface.

Example

```
Device# show ipv6 pim tunnel
```

Configure PIM options

Configure Protocol Independent Multicast (PIM) options on a switch to control multicast traffic behavior and interface parameters.

This task configures PIM options to control when a PIM leaf switch joins the Shortest Path Tree (SPT), set designated router priority, and adjust hello message intervals.

Use this task when you need to customize PIM behavior on your switch to optimize multicast traffic handling and interface communication parameters.

Use these steps to configure PIM options:

Procedure

---

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **ipv6 pim spt-threshold infinity [group-list access-list-name]** command to configure when a PIM leaf switch joins the SPT for the specified groups.

   Example

   ```
   Device(config)# ipv6 pim spt-threshold infinity group-list acc-grp-1
   ```

3. Use the **interface type number** command to specify an interface type and number and enter interface configuration mode.

   Example

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

4. Use the **ipv6 pim dr-priority value** command to configure the DR priority on a PIM switch.

   Example

   ```
   Device(config-if)# ipv6 pim dr-priority 3
   ```

5. Use the **ipv6 pim hello-interval seconds** command to configure the frequency of PIM hello messages on an interface.

   Example

   ```
   Device(config-if)# ipv6 pim hello-interval 45
   ```

6. Use the **ipv6 pim join-prune-interval seconds** command to configure periodic join and prune announcement intervals for a specified interface.

   Example

   ```
   Device(config-if)# ipv6 pim join-prune-interval 75
   ```

7. Use the **exit** command to return to privileged EXEC mode.

   Example

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

   Enter this command twice to exit interface configuration mode and enter privileged EXEC mode.

8. Use the **show ipv6 pim join-prune statistic [interface-type]** command to display the average join-prune aggregation for the most recently aggregated packets for each interface.

   Example

   ```
   Device# show ipv6 pim join-prune statistic
   ```

Reset the PIM traffic counters

Configure the system to clear PIM traffic counters for troubleshooting and verification purposes.

Resetting PIM traffic counters helps you troubleshoot PIM malfunctions and verify that the expected number of PIM packets are received and sent.

If PIM malfunctions, or if you need to verify that the expected number of PIM packets are received and sent, you can clear PIM traffic counters. After clearing the traffic counters, enter the **show ipv6 pim traffic** command to confirm that PIM is functioning correctly and that PIM packets are being received and sent as expected.

Procedure

---

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **clear ipv6 pim traffic** command to reset the PIM traffic counters.

   Example

   ```
   Device# clear ipv6 pim traffic
   ```

3. Use the **show ipv6 pim traffic** command to display the PIM traffic counters.

Example

```
Device# show ipv6 pim traffic
```

Clear the PIM topology table to reset the MRIB connection

Clear the PIM topology table to reset the MRIB connection and verify MRIB information.

This task clears the PIM topology table to reset the MRIB connection and allows verification of MRIB information in certain situations.

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

Follow these steps to clear the PIM topology table to reset the MRIB connection:

Procedure

---

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **clear ipv6 pim [vrf vrfname] topology [group-name | group-address]** command to clear the PIM topology table.

   Example

   ```
   Device# clear ipv6 pim topology ff04::10
   ```

3. Use the **show ipv6 mrib [vrf vrfname] client [filter] [name {client-name | client-name : client-id}]** command to display multicast-related information about an interface.

   Example

   ```
   Device# show ipv6 mrib client
   ```

4. Use the **show ipv6 mrib [vrf vrfname] route {link-local | summary | [sourceaddress-or-name | *] [groupname-or-address [prefix-length]]]** command to display the MRIB route information.

   Example

   ```
   Device# show ipv6 mrib route
   ```

5. Use the **show ipv6 pim [vrf vrfname] topology [groupname-or-address [sourceaddress-or-name] | link-local | route-count [detail]]** command to display PIM topology table information for a specific group or all groups.

Example

```
Device# show ipv6 pim topology
```

Configure static mroutes

Configure static multicast routes (mroutes) in IPv6 as an extension of IPv6 static routes for unicast routing only, multicast RPF selection only, or both unicast routing and multicast RPF selection.

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. Alternatively, you can configure your switch to use a static multicast route for multicast RPF selection only. You can also configure a static route for both unicast routing and multicast RPF selection.

Use this procedure to establish static IPv6 routes for multicast routing on your switch.

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Device# configure terminal
   ```

2. Use the **ipv6 route {ipv6-prefix / prefix-length ipv6-address | interface-type interface-number ipv6-address} [administrative-distance] [administrative-multicast-distance | unicast | multicast] [tag tag]** command to establish static IPv6 routes.

   Example

   ```
   Device(config)# ipv6 route 2001:db8::/64 6::6 100
   ```

3. Use the **exit** command to return to privileged EXEC mode.

   Example

   ```
   Device# exit
   ```

4. Use the **show ipv6 mroute [link-local | [group-name | group-address [source-address | source-name]] [summary] [count]]** command to display the contents of the IPv6 multicast routing table.

   Example

   ```
   Device# show ipv6 mroute ff07::1
   ```

5. Use the **show ipv6 mroute [link-local | group-name | group-address] active [kbps]** command to display the active multicast streams on the switch.

   Example

   ```
   Device# show ipv6 mroute active
   ```

6. Use the **show ipv6 rpf [ipv6-prefix]** command to check RPF information for a given unicast host address and prefix.

   Example

   ```
   Device# show ipv6 rpf 2001::1:1:2
   ```

Using MFIB in IPv6 multicast

Describes how multicast forwarding is automatically enabled in IPv6 networks when IPv6 multicast routing is configured, utilizing the Multicast Forwarding Information Base (MFIB) for efficient packet distribution.

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

Verify MFIB operation in IPv6 multicast

Verify Multicast Forwarding Information Base (MFIB) operation in an IPv6 multicast environment.

This task verifies IPv6 MFIB forwarding entries, active multicast sources, interface forwarding status, and MFIB operational status.

Use this task when you need to verify IPv6 multicast forwarding behavior or troubleshoot MFIB operation on the device.

Procedure

1. Use the **show ipv6 mfib [link-local | linkscope | verbose | vrf vrf-name | group-address-name | ipv6-prefix / prefix-length | source-address-name | active | count | interface | status | summary]** command to display forwarding entries and interfaces in the IPv6 MFIB.

   Example

   ```
   Device# show ipv6 mfib
   ```

2. Use the **show ipv6 mfib [link-local | vrf vrf-name | group-name | group-address] active [kbps]** command to display the rate at which active sources are sending to multicast groups.

   Example

   ```
   Device# show ipv6 mfib active
   ```

3. Use the **show ipv6 mfib [all | linkscope | vrf vrf-name | group-name | group-address [source-name | source-address]]** command to display specific IPv6 MFIB entries.

   Example

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

4. Use the **show ipv6 mfib interface** command to display information about IPv6 multicast-enabled interfaces and their forwarding status.

   Example

   ```
   Device# show ipv6 mfib interface
   ```

5. Use the **show ipv6 mfib status** command to display general MFIB configuration and operational status.

   Example

   ```
   Device# show ipv6 mfib status
   ```

6. Use the **show ipv6 mfib summary** command to display summary information about the number of IPv6 MFIB entries and interfaces.

   Example

   ```
   Device# show ipv6 mfib summary
   ```

Reset MFIB traffic counters

Reset all active MFIB traffic counters using CLI commands.

This task resets MFIB traffic counters to clear accumulated statistics and start fresh monitoring.

Use this procedure when you need to reset MFIB traffic counters for monitoring purposes or troubleshooting.

Procedure

---

1. Use the **enable** command to enter privileged EXEC mode.

   Example

   ```
   Device> enable
   ```

   Enter your password if prompted.

2. Use the **clear ipv6 mfib counters [group-name | group-address [source-address | source-name]]** command to reset all active MFIB traffic counters.

   Example

   ```
   Device# clear ipv6 mfib counters ff04::10
   ```
