



Implementing IPv6 Multicast

- [Finding Feature Information, page 1](#)
- [Information About Implementing IPv6 Multicast Routing, page 1](#)
- [Implementing IPv6 Multicast, page 6](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release.

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

Information About Implementing IPv6 Multicast Routing

This chapter describes how to implement IPv6 multicast routing on the switch.

Traditional IP communication allows a host to send packets to a single host (unicast transmission) or to all hosts (broadcast transmission). IPv6 multicast provides a third scheme, allowing a host to send a single data stream to a subset of all hosts (group transmission) simultaneously.

IPv6 Multicast Overview

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

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

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

The multicast environment consists of senders and receivers. Any host, 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.

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

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

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

IPv6 Multicast Routing Implementation

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

- MLD is used by IPv6 switches to discover multicast listeners (nodes that want to receive multicast packets destined for specific multicast addresses) on directly attached links. 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 with the additional ability to report interest in receiving packets from specific source addresses (or from all but the specific source addresses) to an IP multicast address.

MLD Access Group

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

Explicit Tracking of Receivers

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

Protocol Independent Multicast

Protocol Independent Multicast (PIM) is used between switches so that they can track which multicast packets to forward to each other and to their directly connected LANs. PIM works independently of the unicast routing protocol to perform send or receive multicast route updates like other protocols. Regardless of which unicast

routing protocols are being used in the LAN to populate the unicast routing table, Cisco IOS PIM uses the existing unicast table content to perform the Reverse Path Forwarding (RPF) check instead of building and maintaining its own separate routing table.

You can configure IPv6 multicast to use either PIM-SM or PIM-SSM operation, or you can use both PIM-SM and PIM-SSM together in your network.

PIM-Sparse Mode

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

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

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

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

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

IPv6 BSR: Configure RP Mapping

PIM switches in a domain must be able to map each multicast group to the correct RP address. The BSR protocol for PIM-SM provides a dynamic, adaptive mechanism to distribute group-to-RP mapping information rapidly throughout a domain. With the IPv6 BSR feature, if an RP becomes unreachable, it will be detected and the mapping tables will be modified so that the unreachable RP is no longer used, and the new tables will be rapidly distributed throughout the domain.

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 needs to know which is the correct incoming interface for packets destined for G, because it needs to reject any packets that arrive on other interfaces.

A small set of switches from a domain are configured as candidate bootstrap switches (C-BSRs) and a single BSR is selected for that domain. A set of switches within a domain are also configured as candidate RPs (C-RPs); typically, these switches are the same switches that are configured as C-BSRs. Candidate RPs periodically unicast candidate-RP-advertisement (C-RP-Adv) messages to the BSR of that domain, advertising their willingness to be an RP. A C-RP-Adv message includes the address of the advertising C-RP, and an optional list of group addresses and mask length fields, indicating the group prefixes for which the candidacy is advertised. The BSR then includes a set of these C-RPs, along with their corresponding group prefixes, in bootstrap messages (BSMs) it periodically originates. BSMs are distributed hop-by-hop throughout the domain.

Bidirectional BSR support 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.

PIM-Source Specific Multicast

PIM-SSM is the routing protocol that supports the implementation of SSM and is derived from PIM-SM. However, unlike PIM-SM where data from all multicast sources are sent when there is a PIM join, the SSM feature forwards datagram traffic to receivers from only those multicast sources that the receivers have explicitly joined, thus optimizing bandwidth utilization and denying unwanted Internet broadcast traffic. Further, instead of the use of RP and shared trees, SSM uses information found on source addresses for a multicast group. This information is provided by receivers through the source addresses relayed to the last-hop switches by MLD membership reports, resulting in shortest-path trees directly to the sources.

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

MLD version 2 is required for SSM to operate. MLD allows the host to provide source information. Before SSM can run with MLD, SSM must be supported in the Cisco IOS IPv6 switch, the host where the application is running, and the application itself.

Routable Address Hello Option

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

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

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

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

Static Mroutes

IPv6 static mroutes behave much in the same way as IPv4 static mroutes used to influence the RPF check. IPv6 static mroutes share the same database as IPv6 static routes and are implemented by extending static route support for RPF checks. Static mroutes support equal-cost multipath mroutes, and they also support unicast-only static routes.

MRIB

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

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

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

MFIB

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

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

Distributed MFIB

**Note**

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

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

dMFIB implements the following functions:

- Distributes a copy of the MFIB to the line cards.
- Relays data-driven protocol events generated in the line cards to PIM.
- Provides an MFIB platform application program interface (API) to propagate MFIB changes to platform-specific code responsible for programming the hardware acceleration engine. This API also includes entry points to switch a packet in software (necessary if the packet is triggering a data-driven event) and to upload traffic statistics to the software.
- Provides hooks to allow clients residing on the RP to read traffic statistics on demand. (dMFIB does not periodically upload these statistics to the RP.)

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

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

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

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

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

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

Implementing IPv6 Multicast

Enabling IPv6 Multicast Routing

Beginning in privileged EXEC mode, follow these steps:

DETAILED STEPS

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

Customizing and Verifying the MLD Protocol

Customizing and Verifying MLD on an Interface

Beginning in privileged EXEC mode, follow these steps:

DETAILED STEPS

| | Command or Action | Purpose |
|--------|---|--|
| Step 1 | <code>configure terminal</code> | Enters global configuration mode. |
| Step 2 | <code>interface type number</code> Example: Switch(config)# <code>interface FastEthernet 1/0</code> | Specifies an interface type and number, and places the switch in interface configuration mode. |
| Step 3 | <code>ipv6 mld join-group [group-address] [include exclude] {source-address source-list [acl]}</code> Example: Switch (config-if) # <code>ipv6 mld join-group FF04::10</code> | Configures MLD reporting for a specified group and source. |
| Step 4 | <code>ipv6 mld access-group access-list-name</code> Example: Switch (config-if) # <code>ipv6 access-list acc-grp-1</code> | Allows the user to perform IPv6 multicast receiver access control. |

| | Command or Action | Purpose |
|----------------|---|--|
| Step 5 | <p>ipv6 mld static-group [<i>group-address</i>] [include exclude] {<i>source-address</i> source-list [<i>acl</i>]}</p> <p>Example:</p> <pre>Switch (config-if) # ipv6 mld static-group ff04::10 include 100::1</pre> | Statically forwards traffic for the multicast group onto a specified interface and cause the interface to behave as if a MLD joiner were present on the interface. |
| Step 6 | <p>ipv6 mld query-max-response-time <i>seconds</i></p> <p>Example:</p> <pre>Switch (config-if) # ipv6 mld query-max-response-time 20</pre> | Configures the maximum response time advertised in MLD queries. |
| Step 7 | <p>ipv6 mld query-timeout <i>seconds</i></p> <p>Example:</p> <pre>Switch (config-if) # ipv6 mld query-timeout 130</pre> | Configures the timeout value before the switch takes over as the querier for the interface. |
| Step 8 | <p>exit</p> <p>Example:</p> <pre>Switch (config-if) # exit</pre> | Enter this command twice to exit interface configuration mode and enter privileged EXEC mode. |
| Step 9 | <p>show ipv6 mld groups [link-local] [<i>group-name</i> <i>group-address</i>] [<i>interface-type interface-number</i>] [detail explicit]</p> <p>Example:</p> <pre>Switch # show ipv6 mld groups FastEthernet 2/1</pre> | Displays the multicast groups that are directly connected to the switch and that were learned through MLD. |
| Step 10 | <p>show ipv6 mld groups summary</p> <p>Example:</p> <pre>Switch # show ipv6 mld groups summary</pre> | Displays the number of (*, G) and (S, G) membership reports present in the MLD cache. |
| Step 11 | <p>show ipv6 mld interface [<i>type number</i>]</p> <p>Example:</p> <pre>Switch # show ipv6 mld interface FastEthernet 2/1</pre> | Displays multicast-related information about an interface. |
| Step 12 | <p>debug ipv6 mld [<i>group-name</i> <i>group-address</i> <i>interface-type</i>]</p> | Enables debugging on MLD protocol activity. |

| | Command or Action | Purpose |
|----------------|---|---|
| | Example: Switch # <code>debug ipv6 mld</code> | |
| Step 13 | debug ipv6 mld explicit [<i>group-name</i> <i>group-address</i>] Example: Switch # <code>debug ipv6 mld explicit</code> | Displays information related to the explicit tracking of hosts. |
| Step 14 | copy running-config startup-config | (Optional) Save your entries in the configuration file. |

Implementing MLD Group Limits

Per-interface and global MLD limits operate independently of each other. Both per-interface and global MLD limits can be configured on the same switch. The number of MLD limits, globally or per interface, is not configured by default; the limits must be configured by the user. A membership report that exceeds either the per-interface or the global state limit is ignored.

Configuring Explicit Tracking of Receivers to Track Host Behavior

The explicit tracking feature allows a switch to track the behavior of the hosts within its IPv6 network and enables the fast leave mechanism to be used with MLD version 2 host reports.

Beginning in privileged EXEC mode, follow these steps:

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|--|--|
| Step 1 | configure terminal | Enter global configuration mode. |
| Step 2 | interface <i>type number</i> Example: Switch(config)# <code>interface FastEthernet 1/0</code> | Specifies an interface type and number, and places the switch in interface configuration mode. |
| Step 3 | ipv6 mld explicit-tracking <i>access-list-name</i> Example: Switch(config-if)# <code>ipv6 mld explicit-tracking list1</code> | Enables explicit tracking of hosts. |
| Step 4 | copy running-config startup-config | (Optional) Save your entries in the configuration file. |

Resetting the MLD Traffic Counters

Beginning in privileged EXEC mode, follow these steps.

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|--|---|
| Step 1 | clear ipv6 mld traffic Example: Switch # clear ipv6 mld traffic | Resets all MLD traffic counters. |
| Step 2 | show ipv6 mld traffic Example: Switch # show ipv6 mld traffic | Displays the MLD traffic counters. |
| Step 3 | copy running-config startup-config | (Optional) Save your entries in the configuration file. |

Clearing the MLD Interface Counters

Beginning in privileged EXEC mode, follow these steps.

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|--|---|
| Step 1 | clear ipv6 mld counters <i>interface-type</i> Example: Switch # clear ipv6 mld counters Ethernet1/0 | Clears the MLD interface counters. |
| Step 2 | copy running-config startup-config | (Optional) Save your entries in the configuration file. |

Configuring PIM

This section explains how to configure PIM.

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

Beginning in privileged EXEC mode, follow these steps:

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|---|---|
| Step 1 | configure terminal | Enter global configuration mode. |
| Step 2 | ipv6 pim rp-address <i>ipv6-address</i> [<i>group-access-list</i>] Example: Switch (config) # ipv6 pim rp-address 2001:DB8::01:800:200E:8C6C acc-grp-1 | Configures the address of a PIM RP for a particular group range. |
| Step 3 | exit Example: Switch (config) # exit | Exits global configuration mode, and returns the switch to privileged EXEC mode. |
| Step 4 | show ipv6 pim interface [<i>state-on</i>] [<i>state-off</i>] [<i>type-number</i>] Example: Switch # show ipv6 pim interface | Displays information about interfaces configured for PIM. |
| Step 5 | show ipv6 pim group-map [<i>group-name</i> <i>group-address</i>] [<i>group-range</i> <i>group-mask</i>] [<i>info-source</i> { bsr default embedded-rp static }] Example: Switch # show ipv6 pim group-map | Displays an IPv6 multicast group mapping table. |
| Step 6 | show ipv6 pim neighbor [<i>detail</i>] [<i>interface-type</i> <i>interface-number</i> <i>count</i>] Example: Switch # show ipv6 pim neighbor | Displays the PIM neighbors discovered by the Cisco IOS software. |
| Step 7 | show ipv6 pim range-list [<i>config</i>] [<i>rp-address</i> <i>rp-name</i>] Example: Switch # show ipv6 pim range-list | Displays information about IPv6 multicast range lists. |
| Step 8 | show ipv6 pim tunnel [<i>interface-type</i> <i>interface-number</i>] Example: Switch # show ipv6 pim tunnel | Displays information about the PIM register encapsulation and de-encapsulation tunnels on an interface. |

| | Command or Action | Purpose |
|----------------|--|---|
| Step 9 | debug ipv6 pim [<i>group-name</i> <i>group-address</i> interface <i>interface-type</i> bsr group mvpn neighbor] Example: Switch # debug ipv6 pim | Enables debugging on PIM protocol activity. |
| Step 10 | copy running-config startup-config | (Optional) Save your entries in the configuration file. |

Configuring PIM Options

Beginning in privileged EXEC mode, follow these steps:

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|---|--|
| Step 1 | configure terminal | Enter global configuration mode. |
| Step 2 | ipv6 pim spt-threshold infinity [<i>group-list</i> <i>access-list-name</i>] Example: Switch (config) # ipv6 pim spt-threshold infinity group-list acc-grp-1 | Configures when a PIM leaf switch joins the SPT for the specified groups. |
| Step 3 | ipv6 pim accept-register { <i>list access-list</i> route-map <i>map-name</i> } Example: Switch (config) # ipv6 pim accept-register route-map reg-filter | Accepts or rejects registers at the RP. |
| Step 4 | interface <i>type number</i> Example: Switch (config) # interface FastEthernet 1/0 | Specifies an interface type and number, and places the switch in interface configuration mode. |
| Step 5 | ipv6 pim dr-priority <i>value</i> Example: Switch (config-if) # ipv6 pim dr-priority 3 | Configures the DR priority on a PIM switch. |

| | Command or Action | Purpose |
|----------------|---|--|
| Step 6 | ipv6 pim hello-interval <i>seconds</i> Example: Switch (config-if) # ipv6 pim hello-interval 45 | Configures the frequency of PIM hello messages on an interface. |
| Step 7 | ipv6 pim join-prune-interval <i>seconds</i> Example: Switch (config-if) # ipv6 pim join-prune-interval 75 | Configures periodic join and prune announcement intervals for a specified interface. |
| Step 8 | exit Example: Switch (config-if) # exit | Enter this command twice to exit interface configuration mode and enter privileged EXEC mode. |
| Step 9 | ipv6 pim join-prune statistic [<i>interface-type</i>] Example: Switch (config-if) # show ipv6 pim join-prune statistic | Displays the average join-prune aggregation for the most recently aggregated packets for each interface. |
| Step 10 | copy running-config startup-config | (Optional) Save your entries in the configuration file. |

Resetting the PIM Traffic Counters

If PIM malfunctions or in order to verify that the expected number of PIM packets are received and sent, the user can clear PIM traffic counters. Once the traffic counters are cleared, the user can enter the `show ipv6 pim traffic` command to verify that PIM is functioning correctly and that PIM packets are being received and sent correctly.

Beginning in privileged EXEC mode, follow these steps:

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|--|----------------------------------|
| Step 1 | clear ipv6 pim traffic Example: Switch # clear ipv6 pim traffic | Resets the PIM traffic counters. |

| | Command or Action | Purpose |
|---------------|--|---|
| Step 2 | show ipv6 pim traffic Example: Switch # show ipv6 pim traffic | Displays the PIM traffic counters. |
| Step 3 | copy running-config startup-config | (Optional) Save your entries in the configuration file. |

Clearing the PIM Topology Table to Reset the MRIB Connection

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

Beginning in privileged EXEC mode, follow these steps:

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|--|---|
| Step 1 | clear ipv6 pim topology [<i>group-name</i> <i>group-address</i>] Example: Switch # clear ipv6 pim topology FF04::10 | Clears the PIM topology table. |
| Step 2 | show ipv6 mrib client [<i>filter</i>] [<i>name</i> { <i>client-name</i> <i>client-name</i> : <i>client-id</i> }] Example: Switch # show ipv6 mrib client | Displays multicast-related information about an interface. |
| Step 3 | show ipv6 mrib route { <i>link-local</i> <i>summary</i> [<i>sourceaddress-or-name</i> *] [<i>groupname-or-address</i> [<i>prefix-length</i>]]] Example: Switch # show ipv6 mrib route | Displays the MRIB route information. |
| Step 4 | show ipv6 pim topology [<i>groupname-or-address</i> [<i>sourceaddress-or-name</i>] <i>link-local</i> <i>route-count</i> [<i>detail</i>]] Example: Switch # show ipv6 pim topology | Displays PIM topology table information for a specific group or all groups. |

| | Command or Action | Purpose |
|----------------|---|---|
| Step 5 | debug ipv6 mrib client Example: Switch # <code>debug ipv6 mrib client</code> | Enables debugging on MRIB client management activity. |
| Step 6 | debug ipv6 mrib io Example: Switch # <code>debug ipv6 mrib io</code> | Enables debugging on MRIB I/O events. |
| Step 7 | debug ipv6 mrib proxy Example: Switch # <code>debug ipv6 mrib proxy</code> | Enables debugging on MRIB proxy activity between the switch processor and line cards on distributed switch platforms. |
| Step 8 | debug ipv6 mrib route [<i>group-name</i> <i>group-address</i>] Example: Switch # <code>debug ipv6 mrib route</code> | Displays information about MRIB routing entry-related activity. |
| Step 9 | debug ipv6 mrib table Example: Switch # <code>debug ipv6 mrib table</code> | Enables debugging on MRIB table management activity. |
| Step 10 | copy running-config startup-config | (Optional) Save your entries in the configuration file. |

Configuring a BSR

The tasks included here are described below.

Configuring a BSR and Verifying BSR Information

Beginning in privileged EXEC mode, follow these steps:

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|---------------------------|----------------------------------|
| Step 1 | configure terminal | Enter global configuration mode. |

| | Command or Action | Purpose |
|--------|---|--|
| Step 2 | ipv6 pim bsr candidate bsr <i>ipv6-address[hash-mask-length] [priority priority-value]</i> Example: Switch (config) # ipv6 pim bsr candidate bsr 2001:DB8:3000:3000::42 124 priority 10 | Configures a switch to be a candidate BSR. |
| Step 3 | interface <i>type number</i> Example: Switch (config) # interface FastEthernet 1/0 | Specifies an interface type and number, and places the switch in interface configuration mode. |
| Step 4 | ipv6 pim bsr border Example: Switch (config-if) # ipv6 pim bsr border | Specifies an interface type and number, and places the switch in interface configuration mode. |
| Step 5 | exit Example: Switch (config-if) # exit | Enter this command twice to exit interface configuration mode and enter privileged EXEC mode. |
| Step 6 | show ipv6 pim bsr {election rp-cache candidate-rp} Example: Switch (config-if) # show ipv6 pim bsr election | Displays information related to PIM BSR protocol processing. |
| Step 7 | copy running-config startup-config | (Optional) Save your entries in the configuration file. |

Sending PIM RP Advertisements to the BSR

Beginning in privileged EXEC mode, follow these steps:

DETAILED STEPS

| | Command or Action | Purpose |
|--------|---|---|
| Step 1 | configure terminal | Enter global configuration mode. |
| Step 2 | ipv6 pim bsr candidate rp <i>ipv6-address [group-list</i> <i>access-list-name] [priority priority-value] [interval seconds]</i> | Sends PIM RP advertisements to the BSR. |

| | Command or Action | Purpose |
|---------------|--|--|
| | <p>Example:</p> <pre>Switch(config) # ipv6 pim bsr candidate rp 2001:DB8:3000:3000::42 priority 0</pre> | |
| Step 3 | <p>interface <i>type number</i></p> <p>Example:</p> <pre>Switch(config) # interface FastEthernet 1/0</pre> | Specifies an interface type and number, and places the switch in interface configuration mode. |
| Step 4 | <p>ipv6 pim bsr border</p> <p>Example:</p> <pre>Switch(config-if) # ipv6 pim bsr border</pre> | Configures a border for all BSMS of any scope on a specified interface. |
| Step 5 | copy running-config startup-config | (Optional) Save your entries in the configuration file. |

Configuring BSR for Use Within Scoped Zones

Beginning in privileged EXEC mode, follow these steps:

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|---|---|
| Step 1 | configure terminal | Enter global configuration mode. |
| Step 2 | <p>ipv6 pim bsr candidate rp <i>ipv6-address</i> [<i>hash-mask-length</i>] [<i>priority priority-value</i>]</p> <p>Example:</p> <pre>Switch(config) # ipv6 pim bsr candidate bsr 2001:DB8:1:1:4</pre> | Configures a switch to be a candidate BSR. |
| Step 3 | <p>ipv6 pim bsr candidate rp <i>ipv6-address</i> [group-list <i>access-list-name</i>] [priority <i>priority-value</i>] [interval seconds]</p> <p>Example:</p> <pre>Switch(config) # ipv6 pim bsr candidate rp 2001:DB8:1:1:1 group-list list scope 6</pre> | Configures the candidate RP to send PIM RP advertisements to the BSR. |

| | Command or Action | Purpose |
|---------------|---|--|
| Step 4 | <code>interface type number</code> Example: <code>Switch(config-if) # interface FastEthernet 1/0</code> | Specifies an interface type and number, and places the switch in interface configuration mode. |
| Step 5 | <code>ipv6 multicast boundary scope scope-value</code> Example: <code>Switch(config-if) # ipv6 multicast boundary scope 6</code> | Configures a multicast boundary on the interface for a specified scope. |
| Step 6 | <code>copy running-config startup-config</code> | (Optional) Save your entries in the configuration file. |

Configuring BSR Switches to Announce Scope-to-RP Mappings

IPv6 BSR switches can be statically configured to announce scope-to-RP mappings directly instead of learning them from candidate-RP messages. A user might want to configure a BSR switch to announce scope-to-RP mappings so that an RP that does not support BSR is imported into the BSR. Enabling this feature also allows an RP positioned outside the enterprise's BSR domain to be learned by the known remote RP on the local candidate BSR switch.

Beginning in privileged EXEC mode, follow these steps:

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|--|--|
| Step 1 | <code>configure terminal</code> | Enter global configuration mode. |
| Step 2 | <code>ipv6 pim bsr announced rp ipv6-address [group-list access-list-name] [priority priority-value]</code> Example: <code>Switch(config)# ipv6 pim bsr announced rp 2001:DB8:3000:3000::42 priority 0</code> | Announces scope-to-RP mappings directly from the BSR for the specified candidate RP. |
| Step 3 | <code>copy running-config startup-config</code> | (Optional) Save your entries in the configuration file. |

Configuring SSM Mapping

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

You can use either DNS-based or static SSM mapping, depending on your switch configuration. If you choose to use static SSM mapping, you can configure multiple static SSM mappings. If multiple static SSM mappings are configured, the source addresses of all matching access lists will be used.



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

Beginning in privileged EXEC mode, follow these steps:

DETAILED STEPS

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

Configuring Static Mroutes

Static multicast routes (mroutes) in IPv6 can be implemented as an extension of IPv6 static routes. You can configure your switch to use a static route for unicast routing only, to use a static multicast route for multicast RPF selection only, or to use a static route for both unicast routing and multicast RPF selection.

Beginning in privileged EXEC mode, follow these steps:

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|---|---|
| Step 1 | configure terminal | Enter global configuration mode. |
| Step 2 | <pre>ipv6 route {ipv6-prefix / prefix-length ipv6-address interface-type interface-number ipv6-address} [administrative-distance] [administrative-multicast-distance unicast multicast] [tag tag]</pre> <p>Example:</p> <pre>Switch (config) # ipv6 route 2001:DB8::/64 6::6 100</pre> | Establishes static IPv6 routes. The example shows a static route used for both unicast routing and multicast RPF selection. |
| Step 3 | <pre>exit</pre> <p>Example:</p> <pre>Switch # exit</pre> | Exits global configuration mode, and returns the switch to privileged EXEC mode. |
| Step 4 | <pre>show ipv6 mroute [link-local [group-name group-address] [source-address source-name]] [summary] [count]</pre> <p>Example:</p> <pre>Switch # show ipv6 mroute ff07::1</pre> | Displays the contents of the IPv6 multicast routing table. |
| Step 5 | <pre>show ipv6 mroute [link-local group-name group-address] active [kbps]</pre> <p>Example:</p> <pre>Switch (config-if) # show ipv6 mroute active</pre> | Displays the active multicast streams on the switch. |
| Step 6 | <pre>show ipv6 rpf [ipv6-prefix]</pre> <p>Example:</p> <pre>Switch (config-if) # show ipv6 rpf 2001::1:1:2</pre> | Checks RPF information for a given unicast host address and prefix. |
| Step 7 | copy running-config startup-config | (Optional) Save your entries in the configuration file. |

Using MFIB in IPv6 Multicast

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

Verifying MFIB Operation in IPv6 Multicast

Beginning in privileged EXEC mode, follow these steps:

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|---|---|
| Step 1 | <pre>show ipv6 mfib [link-local verbose group-address-name ipv6-prefix / prefix-length source-address-name active count interface status summary]</pre> <p>Example:</p> <pre>Switch # show ipv6 mfib</pre> | Displays the forwarding entries and interfaces in the IPv6 MFIB. |
| Step 2 | <pre>show ipv6 mfib [link-local group-name group-address] active [kbps]</pre> <p>Example:</p> <pre>Switch # show ipv6 mfib active</pre> | Displays the rate at which active sources are sending to multicast groups. |
| Step 3 | <pre>show ipv6 mfib [all linkscope group-name group-address [source-name source-address]] count</pre> <p>Example:</p> <pre>Switch # show ipv6 mfib ff07::1</pre> | Displays the contents of the IPv6 multicast routing table. |
| Step 4 | <pre>show ipv6 mfib interface</pre> <p>Example:</p> <pre>Switch # show ipv6 mfib interface</pre> | Displays information about IPv6 multicast-enabled interfaces and their forwarding status. |
| Step 5 | <pre>show ipv6 mfib status</pre> <p>Example:</p> <pre>Switch # show ipv6 mfib status</pre> | Displays general MFIB configuration and operational status. |
| Step 6 | <pre>show ipv6 mfib summary</pre> <p>Example:</p> <pre>Switch # show ipv6 mfib summary</pre> | Displays summary information about the number of IPv6 MFIB entries and interfaces. |
| Step 7 | <pre>debug ipv6 mfib [group-name group-address] [adjacency db fs init interface mrib [detail] nat pak platform ppr ps signal table]</pre> | Enables debugging output on the IPv6 MFIB. |

| | Command or Action | Purpose |
|---------------|---|---|
| | Example: Switch # <code>debug ipv6 mfib FF04::10 pak</code> | |
| Step 8 | <code>copy running-config startup-config</code> | (Optional) Save your entries in the configuration file. |

Resetting MFIB Traffic Counters

Beginning in privileged EXEC mode, follow these steps:

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|---|---|
| Step 1 | <code>clear ipv6 mfib counters [group-name group-address [source-address source-name]]</code> Example: Switch # <code>clear ipv6 mfib counters FF04::10</code> | Resets all active MFIB traffic counters. |
| Step 2 | <code>copy running-config startup-config</code> | (Optional) Save your entries in the configuration file. |