CHAPTER 1

# Implementing IPv6 Multicast

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.

This document provides information on how to implement IPv6 multicast. For the command reference, refer to the Cisco IOS IPv6 Command Reference.

## Information About Implementing IPv6 Multicast

### 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.

**Note** IPv6 Multicast Routing is supported only on the IP Services image.

# 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.

Multicast Listener Discovery Protocol for IPv6

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

The difference between multicast queriers and hosts is as follows:

- A querier is a network device, such as a switch, that sends query messages to discover which network devices are members of a given multicast group.
- A host is a receiver, including switches, that send report messages to inform the querier of a host membership.

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

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

MLD has three types of messages:

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

Group-specific and multicast address-specific queries both set the multicast address field to the address being queried. A group address is a multicast address.

- Report--In a report message, the multicast address field is that of the specific IPv6 multicast address to which the sender is listening.
- Done--In a done message, the multicast address field is that of the specific IPv6 multicast address to which the source of the MLD message is no longer listening.

**Note** There is no done message in MLD version 2.

An MLD report must be sent with a valid IPv6 link-local source address, or the unspecified address (::), if the sending interface has not yet acquired a valid link-local address. Sending reports with the unspecified address is allowed to support the use of IPv6 multicast in the Neighbor Discovery Protocol.

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

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

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

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

## MLD Access Group

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.

## IPv6 Multicast User Authentication and Profile Support

IPv6 multicast by design allows any host in the network to become a receiver or a source for a multicast group. Therefore, multicast access control is needed to control multicast traffic in the network. Access control functionality consists mainly of source access control and accounting, receiver access control and accounting, and provisioning of this access control mechanism.

Multicast access control provides an interface between multicast and authentication, authorization, and accounting (AAA) for provisioning, authorizing, and accounting at the last-hop switch, receiver access control functions in multicast, and group or channel disabling capability in multicast.

When you deploy a new multicast service environment, it is necessary to add user authentication and provide a user profile download on a per-interface basis. The use of AAA and IPv6 multicast supports user authentication and downloading of the user profile in a multicast environment.

The event that triggers the download of a multicast access-control profile from the RADIUS server to the access switch is arrival of an MLD join on the access switch. When this event occurs, a user can cause the authorization cache to time out and request download periodically or use an appropriate multicast clear command to trigger a new download in case of profile changes.

Accounting occurs via RADIUS accounting. Start and stop accounting records are sent to the RADIUS server from the access switch. In order for you to track resource consumption on a per-stream basis, these accounting records provide information about the multicast source and group. The start record is sent when the last-hop switch receives a new MLD report, and the stop record is sent upon MLD leave or if the group or channel is deleted for any reason.

## IPv6 MLD Proxy

The MLD proxy feature provides a mechanism for a switch to generate MLD membership reports for all (*, G)/(S, G) entries or a user-defined subset of these entries on the switch's upstream interface. The MLD proxy feature enables a device to learn proxy group membership information, and forward multicast packets based upon that information.

If a switch is acting as RP for mroute proxy entries, MLD membership reports for these entries can be generated on user specified proxy interface.

# 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.

### Designated Switch

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

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

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

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

**Note**

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

### Rendezvous Point

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

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

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

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

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

The RP address is used by first-hop switches 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 switches to send PIM join and prune messages to the RP to inform it about group membership. You must configure the RP address on all switches (including the RP switch).

A PIM switch 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 switch is an RP.

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

### PIMv6 Anycast RP Solution Overview

The anycast RP solution in IPv6 PIM allows an IPv6 network to support anycast services for the PIM-SM RP. It allows anycast RP to be used inside a domain that runs PIM only. This feature is useful when interdomain connection is not required. Anycast RP can be used in IPv4 as well as IPv6, but it does not depend on the Multicast Source Discovery Protocol (MSDP), which runs only on IPv4.

Anycast RP is a mechanism that ISP-based backbones use to get fast convergence when a PIM RP device fails. To allow receivers and sources to rendezvous to the closest RP, the packets from a source need to get to all RPs to find joined receivers.

A unicast IP address is chosen as the RP address. This address is either statically configured or distributed using a dynamic protocol to all PIM devices throughout the domain. A set of devices in the domain is chosen to act as RPs for this RP address; these devices are called the anycast RP set. Each device in the anycast RP set is configured with a loopback interface using the RP address. Each device in the anycast RP set also needs a separate physical IP address to be used for communication between the RPs.

The RP address, or a prefix that covers the RP address, is injected into the unicast routing system inside of the domain. Each device in the anycast RP set is configured with the addresses of all other devices in the anycast RP set, and this configuration must be consistent in all RPs in the set.

## 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.

### SSM Mapping for IPv6

SSM mapping for IPv6 supports both static and dynamic Domain Name System (DNS) mapping for MLD version 1 receivers. This feature allows deployment of IPv6 SSM with hosts that are incapable of providing MLD version 2 support in their TCP/IP host stack and their IP multicast receiving application.

SSM mapping allows the switch to look up the source of a multicast MLD version 1 report either in the running configuration of the switch or from a DNS server. The switch can then initiate an (S, G) join toward the source.

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

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

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

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

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

Join and prune messages are sent for sources and RPs. They are sent hop-by-hop and are processed by each PIM switch 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 switch that is directly connected to a source and are received by the RP for the group.

### Reverse Path Forwarding

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

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

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

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

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

## 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.

## Bidirectional PIM

Bidirectional PIM allows multicast switches to keep reduced state information, as compared with unidirectional shared trees in PIM-SM. Bidirectional shared trees convey data from sources to the rendezvous point address (RPA) and distribute them from the RPA to the receivers. Unlike PIM-SM, bidirectional PIM does not switch over to the source tree, and there is no register encapsulation of data from the source to the RP.

A single designated forwarder (DF) exists for each RPA on every link within a bidirectional PIM domain (including multiaccess and point-to-point links). The only exception is the RPL on which no DF exists. The DF is the switch on the link with the best route to the RPA, which is determined by comparing MRIB-provided metrics. A DF for a given RPA forwards downstream traffic onto its link and forwards upstream traffic from its link toward the rendezvous point link (RPL). The DF performs this function for all bidirectional groups that map to the RPA. The DF on a link is also responsible for processing Join messages from downstream switches on the link as well as ensuring that packets are forwarded to local receivers discovered through a local membership mechanism such as MLD.

Bidirectional PIM offers advantages when there are many moderate or low-rate sources. However, the bidirectional shared trees may have worse delay characteristics than do the source trees built in PIM-SM (depending on the topology).

Only static configuration of bidirectional RPs is supported in IPv6.

## 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

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 VRF Lite

The IPv6 Multicast VRF Lite feature provides IPv6 multicast support for multiple virtual routing/forwarding contexts (VRFs). The scope of these VRFs is limited to the switch in which the VRFs are defined.

This feature provides separation between routing and forwarding, providing an additional level of security because no communication between devices belonging to different VRFs is allowed unless it is explicitly configured. The IPv6 Multicast VRF Lite feature simplifies the management and troubleshooting of traffic belonging to a specific VRF.

## 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.

## Multiprotocol BGP for the IPv6 Multicast Address Family

The multiprotocol BGP for the IPv6 multicast address family feature provides multicast BGP extensions for IPv6 and supports the same features and functionality as IPv4 BGP. IPv6 enhancements to multicast BGP include support for an IPv6 multicast address family and network layer reachability information (NLRI) and next hop (the next switch in the path to the destination) attributes that use IPv6 addresses.

Multicast BGP is an enhanced BGP that allows the deployment of interdomain IPv6 multicast. Multiprotocol BGP carries routing information for multiple network layer protocol address families; for example, IPv6 address family and for IPv6 multicast routes. The IPv6 multicast address family contains routes used for RPF lookup by the IPv6 PIM protocol, and multicast BGP IPV6 provides for interdomain transport of the same. Users must use multiprotocol BGP for IPv6 multicast when using IPv6 multicast with BGP because the unicast BGP learned routes will not be used for IPv6 multicast.

Multicast BGP functionality is provided through a separate address family context. A subsequent address family identifier (SAFI) provides information about the type of the network layer reachability information that is carried in the attribute. Multiprotocol BGP unicast uses SAFI 1 messages, and multiprotocol BGP multicast uses SAFI 2 messages. SAFI 1 messages indicate that the routes are only usable for IP unicast, but not IP multicast. Because of this functionality, BGP routes in the IPv6 unicast RIB must be ignored in the IPv6 multicast RPF lookup.

A separate BGP routing table is maintained to configure incongruent policies and topologies (for example, IPv6 unicast and multicast) by using IPv6 multicast RPF lookup. Multicast RPF lookup is very similar to the IP unicast route lookup.

No MRIB is associated with the IPv6 multicast BGP table. However, IPv6 multicast BGP operates on the unicast IPv6 RIB when needed. Multicast BGP does not insert or update routes into the IPv6 unicast RIB.

## NSF and SSO Support In IPv6 Multicast

Support for nonstop forwarding (NSF) and stateful switchover (SSO) is provided in IPv6 Multicast.

## Bandwidth-Based CAC for IPv6 Multicast

The bandwidth-based call admission control (CAC) for IPv6 multicast feature implements a way to count per-interface mroute state limiters using cost multipliers. This feature can be used to provide bandwidth-based CAC on a per-interface basis in network environments where the multicast flows use different amounts of bandwidth.

This feature limits and accounts for IPv6 multicast state in detail. When this feature is configured, interfaces can be limited to the number of times they may be used as incoming or outgoing interfaces in the IPv6 multicast PIM topology.

With this feature, switch administrators can configure global limit cost commands for state matching access lists and specify which cost multiplier to use when accounting such state against the interface limits. This feature provides the required flexibility to implement bandwidth-based local CAC policy by tuning appropriate cost multipliers for different bandwidth requirements.

# Implementing IPv6 Multicast

- Enabling IPv6 Multicast Routing, page 1-13

- Customizing and Verifying the MLD Protocol, page 1-13
- Configuring PIM, page 1-19
- Configuring a BSR, page 1-24
- Configuring SSM Mapping, page 1-26
- Configuring Static Mroutes, page 1-27
- Using MFIB in IPv6 Multicast, page 1-28

# Enabling IPv6 Multicast Routing

Beginning in privileged EXEC mode, follow these steps:

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

# Customizing and Verifying the MLD Protocol

- Customizing and Verifying MLD on an Interface, page 1-13
- Implementing MLD Group Limits, page 1-15
- Configuring Explicit Tracking of Receivers to Track Host Behavior, page 1-16
- Configuring Multicast User Authentication and Profile Support, page 1-16
- Enabling MLD Proxy in IPv6, page 1-18
- Resetting the MLD Traffic Counters, page 1-19
- Clearing the MLD Interface Counters, page 1-19

## Customizing and Verifying MLD on an Interface

Beginning in privileged EXEC mode, follow these steps:

| | Command | Purpose |
|---|---|---|
| Step 1 | **configure terminal** | Enter global configuration mode. |
| Step 2 | **interface** *type number*<br>**Example:**<br>`Switch(config)# interface FastEthernet 1/0` | Specifies an interface type and number, and places the switch in interface configuration mode. |

| | Command | Purpose |
|---|---|---|
| **Step 3** | **ipv6 mld join-group** [*group-address*] [**include** \| **exclude**] {*source-address* \| **source-list** [*acl*]}<br>**Example:**<br>`Switch(config-if)# ipv6 mld join-group FF04::10` | Configures MLD reporting for a specified group and source. |
| **Step 4** | **ipv6 mld access-group** *access-list-name*<br>**Example:**<br>`Switch(config-if)# ipv6 access-list acc-grp-1` | Allows the user to perform IPv6 multicast receiver access control. |
| **Step 5** | **ipv6 mld static-group** *group-address* ] [**include**\| **exclude**] {*source-address* \| source-list [*acl*]}<br>**Example:**<br>`Switch(config-if)# ipv6 mld static-group ff04::10 include 100::1` | Statically forwards traffic for the multicast group onto a specified interface and cause the interface to behave as if a MLD joiner were present on the interface. |
| **Step 6** | **ipv6 mld query-max-response-time** *seconds*<br>**Example:**<br>`Switch(config-if)# ipv6 mld query-max-response-time 20` | Configures the maximum response time advertised in MLD queries. |
| **Step 7** | **ipv6 mld query-timeout** *seconds*<br>**Example:**<br>`Switch(config-if)# ipv6 mld query-timeout 130` | Configures the timeout value before the switch takes over as the querier for the interface. |
| **Step 8** | **exit**<br>**Example:**<br>`Switch(config-if)# exit` | Enter this command twice to exit interface configuration mode and enter privileged EXEC mode. |
| **Step 9** | **show ipv6 mld** [**vrf** *vrf-name*] **groups** [**link-local**] [*group-name* \| *group-address*] [*interface-type interface-number*] [**detail** \| **explicit**]<br>**Example:**<br>`Switch# show ipv6 mld groups FastEthernet 2/1` | Displays the multicast groups that are directly connected to the switch and that were learned through MLD. |
| **Step 10** | **show ipv6 mld groups summary**<br>**Example:**<br>`Switch# show ipv6 mld groups summary` | Displays the number of (*, G) and (S, G) membership reports present in the MLD cache. |
| **Step 11** | **show ipv6 mld** [**vrf** *vrf-name*] **interface** [*type number*]<br>**Example:**<br>`Switch# show ipv6 mld interface FastEthernet 2/1` | Displays multicast-related information about an interface. |

| | Command | Purpose |
|---|---|---|
| Step 12 | **debug ipv6 mld** [*group-name* \| *group-address* \| *interface-type*]<br>**Example:**<br>`Switch# debug ipv6 mld` | Enables debugging on MLD protocol activity. |
| Step 13 | **debug ipv6 mld explicit** [*group-name* \| *group-address*]<br>**Example:**<br>`Switch# debug ipv6 mld explicit` | Displays information related to the explicit tracking of hosts. |
| Step 14 | **copy running-config startup-config** | (Optional) Save your entries in the configuration file. |

## 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.

### Implementing MLD Group Limits Globally

Beginning in privileged EXEC mode, follow these steps:

| | Command | Purpose |
|---|---|---|
| Step 1 | **configure terminal** | Enter global configuration mode. |
| Step 2 | **ipv6 mld** [vrf *vrf-name*] **state**-limit *number*<br>**Example:**<br>`Switch(config)# ipv6 mld state-limit 300` | Limits the number of MLD states globally. |
| Step 3 | **copy running-config startup-config** | (Optional) Save your entries in the configuration file. |

### Implementing MLD Group Limits per Interface

Beginning in privileged EXEC mode, follow these steps:

| | Command | Purpose |
|---|---|---|
| Step 1 | **configure terminal** | Enter global configuration mode. |
| Step 2 | **interface** *type number*<br>**Example:**<br>`Switch(config)# interface FastEthernet 1/0` | Specifies an interface type and number, and places the switch in interface configuration mode. |
| Step 3 | **ipv6 mld limit** *number* [**except** *access-list*<br>**Example:**<br>`Switch(config-if)# ipv6 mld limit 100` | Limits the number of MLD states on a per-interface basis. |
| Step 4 | **copy running-config startup-config** | (Optional) Save your entries in the configuration file. |

## 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:

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

## Configuring Multicast User Authentication and Profile Support

Before you configure multicast user authentication and profile support, you should be aware of the following restrictions:

- The port, interface, VC, or VLAN ID is the user or subscriber identity. User identity by hostname, user ID, or password is not supported.
- Enabling AAA Access Control for IPv6 Multicast
- Specifying Method Lists and Enabling Multicast Accounting
- Disabling the Switch from Receiving Unauthenticated Multicast Traffic
- Resetting Authorization Status on an MLD Interface

### Enabling AAA Access Control for IPv6 Multicast

Beginning in privileged EXEC mode, follow these steps:

| | Command | Purpose |
|---|---|---|
| Step 1 | **configure terminal** | Enter global configuration mode. |
| Step 2 | **aaa new-model**<br>**Example:**<br>`Switch(config)# aaa new-model` | Enables the AAA access control system. |
| Step 3 | **copy running-config startup-config** | (Optional) Save your entries in the configuration file. |

### Specifying Method Lists and Enabling Multicast Accounting

Perform this task to specify the method lists used for AAA authorization and accounting and how to enable multicast accounting on specified groups or channels on an interface.

Beginning in privileged EXEC mode, follow these steps:

| | Command | Purpose |
|---|---|---|
| Step 1 | **configure terminal** | Enter global configuration mode. |
| Step 2 | **aaa authorization multicast default** [*method3* \| *method4*]<br>**Example:**<br>`Switch(config)# aaa authorization multicast default` | Enables AAA authorization and sets parameters that restrict user access to an IPv6 multicast network. |
| Step 3 | **aaa accounting multicast default** [**start-stop** \| **stop-only**] [**broadcast**] [*method1*] [*method2*] [*method3*] [*method4*]<br>**Example:**<br>`Switch(config)# aaa accounting multicast default` | Enables AAA accounting of IPv6 multicast services for billing or security purposes when you use RADIUS. |
| Step 4 | **interface** *type number*<br>**Example:**<br>`Switch(config)# interface FastEthernet 1/0` | Specifies an interface type and number, and places the switch in interface configuration mode. |
| Step 5 | **ipv6 multicast aaa account receive** *access-list-name* [**throttle** *throttle-number*]<br>**Example:**<br>`Switch(config-if)# ipv6 multicast aaa account receive list1` | Enables AAA accounting on specified groups or channels. |
| Step 6 | **copy running-config startup-config** | (Optional) Save your entries in the configuration file. |

### Disabling the Switch from Receiving Unauthenticated Multicast Traffic

In some situations, access control may be needed to prevent multicast traffic from being received unless the subscriber is authenticated and the channels are authorized as per access control profiles. That is, there should be no traffic at all unless specified otherwise by access control profiles.

Perform this task to disable the switch from receiving multicast traffic to be received from unauthenticated groups or unauthorized channels.

Beginning in privileged EXEC mode, follow these steps:

| | Command | Purpose |
|---|---|---|
| Step 1 | **configure terminal** | Enter global configuration mode. |
| Step 2 | **ipv6 multicast** [vrf *vrf-name*] **group-range**[*access-list-name*]<br>**Example:**<br>`Switch(config)# ipv6 multicast group-range` | Disables multicast protocol actions and traffic forwarding for unauthorized groups or channels on all the interfaces in a switch. |
| Step 3 | **copy running-config startup-config** | (Optional) Save your entries in the configuration file. |

## Enabling MLD Proxy in IPv6

Beginning in privileged EXEC mode, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | **configure terminal** | Enter global configuration mode. |
| **Step 2** | **ipv6 mld host-proxy** [*group-acl*]<br>**Example:**<br>`Switch(config)# ipv6 mld host-proxy proxy-group` | Enables the MLD proxy feature. |
| **Step 3** | **ipv6 mld host-proxy interface** [*group-acl*]<br>**Example:**<br>`Switch(config)# ipv6 mld host-proxy interface Ethernet 0/0` | Enables the MLD proxy feature on a specified interface on an RP. |
| **Step 4** | **show ipv6 mld host-proxy** [*interface-type interface-number*] **group** [*group-address*]]<br>**Example:**<br>`Switch(config)# show ipv6 mld host-proxy Ethernet0/0` | Displays IPv6 MLD host proxy information. |
| **Step 5** | **copy running-config startup-config** | (Optional) Save your entries in the configuration file. |

### Resetting Authorization Status on an MLD Interface

If no interface is specified, authorization is reset on all MLD interfaces.

Beginning in privileged EXEC mode, follow these steps:

| | Command | Purpose |
|---|---|---|
| **Step 1** | **clear ipv6 multicast aaa authorization** [*interface-type interface-number*]<br>**Example:**<br>`Switch# clear ipv6 multicast aaa authorization FastEthernet 1/0` | Clears parameters that restrict user access to an IPv6 multicast network. |
| **Step 2** | **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:

| | Command | Purpose |
|---|---|---|
| **Step 1** | clear ipv6 mld [vrf vrf-name] traffic<br>**Example:**<br>`Switch# clear ipv6 mld traffic` | Resets all MLD traffic counters. |
| **Step 2** | **show ipv6 mld** [vrf *vrf-name*] **traffic**<br>**Example:**<br>`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:

| | Command | Purpose |
|---|---|---|
| **Step 4** | **clear ipv6 ml**d [vrf *vrf-name*] **counters** *interface-type*<br>**Example:**<br>`Switch# clear ipv6 mld counters Ethernet1/0` | Clears the MLD interface counters. |
| **Step 5** | **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:

| | Command | Purpose |
|---|---|---|
| **Step 1** | **configure terminal** | Enter global configuration mode. |
| **Step 2** | ipv6 pim [**vrf** *vrf-name*] **rp-address** *ipv6-address* [*group-access-list*] [**bidir**]<br>**Example:**<br>`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. |

| | Command | Purpose |
|---|---|---|
| Step 3 | **exit**<br>**Example:**<br>`Switch(config)# exit` | Exits global configuration mode, and returns the switch to privileged EXEC mode |
| Step 4 | **show ipv6 pim** [**vrf** *vrf-name*] **interface** [state-on] [**state-off**] [*type number*]<br>**Example:**<br>`Switch# show ipv6 pim interface` | Displays information about interfaces configured for PIM. |
| Step 5 | **show ipv6 pim** [**vrf** *vrf-name*] **group-map** [*group-name* \| group-address] \| [*group-range* \| *group-mask*] [**info-source** {**bsr** \| **default** \| **embedded-rp** \| **static**}]<br>**Example:**<br>`Switch# show ipv6 pim group-map` | Displays an IPv6 multicast group mapping table. |
| Step 6 | **show ipv6 pim** [vrf *vrf-name*] **neighbor** [**detail**] [*interface-type interface-number* \| **count**]<br>**Example:**<br>`Switch# show ipv6 pim neighbor` | Displays the PIM neighbors discovered by the Cisco IOS software. |
| Step 7 | **show ipv6 pim** [**vrf** *vrf-name*] **range-list**[**config**] [*rp-address* \| *rp-name*]<br>**Example:**<br>`Switch# show ipv6 pim range-list` | Displays information about IPv6 multicast range lists. |
| Step 8 | **show ipv6 pim** [**vrf** *vrf-name*] **tunnel** [*interface-type interface-number*]<br>**Example:**<br>`Switch# show ipv6 pim tunnel` | Displays information about the PIM register encapsulation and de-encapsulation tunnels on an interface. |
| Step 9 | **debug ipv6 pim** [*group-name* \| *group-address* \| **interface** *interface-typ*e \| **bsr** \| **group** \| **mvpn** \| **neighbor**]<br>**Example:**<br>`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:

| | Command | Purpose |
|---|---|---|
| Step 1 | **configure terminal** | Enter global configuration mode. |
| Step 2 | **ipv6 pim** [**vrf** *vrf-name*] **spt-threshold infinity** [**group-list** *access-list-name*]<br>**Example:**<br>`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** [vrf *vrf-name*] **accept-register** {**list** *access-list* \| **route-map** *map-name*}<br>**Example:**<br>`Switch(config)# ipv6 pim accept-register route-map reg-filter` | Accepts or rejects registers at the RP. |
| Step 4 | **interface** *type number*<br>**Example:**<br>`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** *value*<br>**Example:**<br>`Switch(config-if)# ipv6 pim dr-priority 3` | Configures the DR priority on a PIM switch. |
| Step 6 | **ipv6 pim hello-interval** *seconds*<br>**Example:**<br>`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** *seconds*<br>**Example:**<br>`Switch(config-if)# ipv6 pim join-prune-interval 75` | Configures periodic join and prune announcement intervals for a specified interface. |
| Step 8 | **exit**<br>**Example:**<br>`Switch(config-if)# exit` | Enter this command twice to exit interface configuration mode and enter privileged EXEC mode. |
| Step 9 | **show ipv6 pim** [vrf *vrf-name*] **join-prune statistic** [*interface-type*]<br>**Example:**<br>`Switch# 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. |

## Configuring Bidirectional PIM and Displaying Bidirectional PIM Information

Beginning in privileged EXEC mode, follow these steps:

| | Command | Purpose |
|---|---|---|
| Step 1 | **configure terminal** | Enter global configuration mode. |
| Step 2 | **ipv6 pim** [*vrf vrf-name*] **rp-address** *ipv6-address* [*group-access-list*] [**bidir**]<br>**Example:**<br>`Switch(config)# ipv6 pim rp-address 2001:DB8::01:800:200E:8C6C bidir` | Configures the address of a PIM RP for a particular group range. Use of the **bidir** keyword means that the group range will be used for bidirectional shared-tree forwarding. |
| Step 3 | **exit**<br>**Example:**<br>`Switch(config-if)# exit` | Exits global configuration mode, and returns the switch to privileged EXEC mode. |
| Step 4 | **show ipv6 pim** [**vrf** *vrf-name*] **df** [*interface-type interface-number*] [*rp-address*]<br>**Example:**<br>`Switch# show ipv6 pim df` | Displays the designated forwarder (DF)-election state of each interface for RP |
| Step 5 | **show ipv6 pim** [vrf *vrf-name*] **df winner**[*interface-type interface-number*] [*rp-address*]<br>**Example:**<br>`Switch# show ipv6 pim df winner ethernet 1/0 200::1` | Displays the DF-election winner on each interface for each RP. |
| Step 6 | **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:

| | Command | Purpose |
|---|---|---|
| Step 1 | **clear ipv6 pim** [**vrf** *vrf-name*] **traffic**<br>**Example:**<br>`Switch# clear ipv6 pim traffic` | Resets the PIM traffic counters. |
| Step 2 | **show ipv6 pim** [**vrf** *vrf-name*] **traffic**<br>**Example:**<br>`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:

| | Command | Purpose |
|---|---|---|
| **Step 1** | **clear ipv6 pim** [**vrf** *vrf-name*] **topology** [*group-name* \| *group-address*]<br>**Example:**<br>`Switch# clear ipv6 pim topology FF04::10` | Clears the PIM topology table. |
| **Step 2** | **show ipv6 mrib** [**vrf** *vrf-name*] **client** [**filter**] [**name** {*client-name* \| *client-name* **: client-id**}]<br>**Example:**<br>`Switch# show ipv6 mrib client` | Displays multicast-related information about an interface. |
| **Step 3** | **show ipv6 mrib** [**vrf** *vrf-name*] **route [link-local\| summary** \| [*sourceaddress-or-name* \| *] [*groupname-or-address* [*prefix-length*]]]<br>**Example:**<br>`Switch# show ipv6 mrib route` | Displays the MRIB route information. |
| **Step 4** | **show ipv6 pim** [**vrf** *vrf-name*] **topology** [*groupname-or-address* [*sourcename-or-address*] \| **link-local** \| **route-count** [**detail**]]<br>**Example:**<br>`Switch# show ipv6 pim topology` | Displays PIM topology table information for a specific group or all groups. |
| **Step 5** | **debug ipv6 mrib** [**vrf** *vrf-name*] **client**<br>**Example:**<br>`Switch# debug ipv6 mrib client` | Enables debugging on MRIB client management activity. |
| **Step 6** | **debug ipv6 mrib** [vrf *vrf-name*] **io**<br>**Example:**<br>`Switch# debug ipv6 mrib io` | Enables debugging on MRIB I/O events. |
| **Step 7** | **debug ipv6 mrib proxy**<br>**Example:**<br>`Switch# debug ipv6 mrib proxy` | Enables debugging on MRIB proxy activity between the switch processor and line cards on distributed switch platforms. |
| **Step 8** | **debug ipv6 mrib** [vrf *vrf-name*] **route** [*group-name* \| *group-address*]<br>**Example:**<br>`Switch# debug ipv6 mrib route` | Displays information about MRIB routing entry-related activity. |
| **Step 9** | **debug ipv6 mrib** [**vrf** *vrf-name*] **table**<br>**Example:**<br>`Switch# debug ipv6 mrib table` | 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:

| | Command | Purpose |
|---|---|---|
| Step 1 | **configure terminal** | Enter global configuration mode. |
| Step 2 | **ipv6 pim** [**vrf** *vrf-name*] **bsr candidate bsr** *ipv6-address*[*hash-mask-length*] [**priority** *priority-value*]<br>**Example:**<br>`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** *type number*<br>**Example:**<br>`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 borde**r<br>**Example:**<br>`Switch(config-if)# ipv6 pim bsr border` | Configures a border for all BSMs of any scope on a specified interface. |
| Step 5 | **exit**<br>**Example:**<br>`Switch(config-if)# exit` | Enter this command twice to exit interface configuration mode and enter privileged EXEC mode. |
| Step 6 | **show ipv6 pim** [**vrf** *vrf-name*] bsr {**election** \| **rp-cache** \| **candidate-rp**}<br>**Example:**<br>`Switch# 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:

| | Command | Purpose |
|---|---|---|
| Step 1 | **configure terminal** | Enter global configuration mode. |
| Step 2 | **ipv6 pim** [**vrf** *vrf-name*] **bsr candidate rp** *ipv6-address* [**group-list** *access-list-name*] [**priority** *priority-value*] [**interval** *seconds*] [**scope** *scope-value*] [**bidir**]<br>**Example:**<br>`Switch(config)# ipv6 pim bsr candidate rp 2001:DB8:3000:3000::42 priority 0` | Sends PIM RP advertisements to the BSR. |
| Step 3 | **interface** *type number*<br>**Example:**<br>`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**<br>**Example:**<br>`Switch(config-if)# ipv6 pim bsr border` | 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:

| | Command | Purpose |
|---|---|---|
| Step 1 | **configure terminal** | Enter global configuration mode. |
| Step 2 | **ipv6 pim** [**vrf** *vrf-name*] **bsr candidate bsr** *ipv6-address* [*hash-mask-length*] [**priority** *priority-value*]<br>**Example:**<br>`Switch(config)# ipv6 pim bsr candidate bsr 2001:DB8:1:1:4` | Configures a switch to be a candidate BSR. |
| Step 3 | **ipv6 pim** [**vrf** *vrf-name*] **bsr candidate rp** *ipv6-address* [**group-list** *access-list-name*] [**priority** *priority-value*] [**interval** *seconds*] [**scope** *scope-value*] [**bidir**]<br>**Example:**<br>`Switch(config)# ipv6 pim bsr candidate rp 2001:DB8:1:1:1 group-list list scope 6` | Configures the candidate RP to send PIM RP advertisements to the BSR. |
| Step 4 | **interface** *type number*<br>**Example:**<br>`Switch(config)# interface FastEthernet 1/0` | Specifies an interface type and number, and places the switch in interface configuration mode. |

| | Command | Purpose |
|---|---|---|
| **Step 5** | **ipv6 multicast boundary scope** *scope-value*<br>**Example:**<br>`Switch(config-if)# ipv6 multicast boundary scope 6` | Configures a multicast boundary on the interface for a specified scope. |
| **Step 6** | **copy running-config startup-config** | (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:

| | Command | Purpose |
|---|---|---|
| **Step 1** | **configure terminal** | Enter global configuration mode. |
| **Step 2** | **ipv6 pim** [**vrf** *vrf-name*] **bsr announced rp** *ipv6-address* [**group-list** *access-list-name*] [**priority** *priority-value*] [**bidir**] [**scope** *scope-value*]<br>**Example:**<br>`Switch(config)# ipv6 pim bsr announced rp 2001:DB8:3000:3000::42 priority 0` | Announces scope-to-RP mappings directly from the BSR for the specified candidate RP. |
| **Step 3** | **copy running-config startup-config** | (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:

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

| | Command | Purpose |
|---|---|---|
| Step 1 | **configure terminal** | Enter global configuration mode. |
| Step 2 | **ipv6 route** *ipv6-prefix* / *prefix-length ipv6-address* \| *interface-type interface-number ipv6-address*]} [*administrative-distance*] [*administrative-multicast-distance* \| **unicast**\| **multicast**] [**tag** *tag*<br>**Example:**<br>`Switch(config)# ipv6 route 2001:DB8::/64 6::6 100` | Establishes static IPv6 routes. The example shows a static route used for both unicast routing and multicast RPF selection. |

| | Command | Purpose |
|---|---|---|
| **Step 3** | **exit**<br>**Example:**<br>`Switch(config-if)# exit` | Exits global configuration mode, and returns the switch to privileged EXEC mode. |
| **Step 4** | **show ipv6 mroute** [**vrf** *vrf-name*] [link-local \| [*group-name* \| *group-address* [*source-address* \| *source-name*]] [**summary**] [**count**]<br>**Example:**<br>`Switch# show ipv6 mroute ff07::1` | Displays the contents of the IPv6 multicast routing table. |
| **Step 5** | **show ipv6 mroute** [**vrf** *vrf-name*] [**link-local** \| *group-name* \| *group-address*] **active**[*kbps*]<br>**Example:**<br>`Switch# show ipv6 mroute active` | Displays the active multicast streams on the switch. |
| **Step 6** | **show ipv6 rpf** [**vrf** *vrf-name*] *ipv6-prefix*<br>**Example:**<br>`Switch# show ipv6 rpf 2001:DB8::1:1:2` | 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:

| | Command | Purpose |
|---|---|---|
| **Step 1** | **show ipv6 mfib** [**vrf** *vrf-name*] [**link-local** \| **verbose** \| *group-address-name* \| *ipv6-prefix* / *prefix-length* \| *source-address-name*\| **active** \| **count** \| **interface** \| **status** \| **summary**]<br>**Example:**<br>`Switch# show ipv6 mfib` | Displays the forwarding entries and interfaces in the IPv6 MFIB. |
| **Step 2** | **show ipv6 mfib** [**vrf** *vrf-name*] [**link-local**\| *group-name* \| *group-address*] **active** [*kbps*]<br>**Example:**<br>`Switch# show ipv6 mfib active` | Displays the rate at which active sources are sending to multicast groups. |
| **Step 3** | **show ipv6 mfib** [**vrf** *vrf-name*] [**all** \| **linkscope**\| *group-name* \| *group-address* [*source-name* \| *source-address*]] **count**<br>**Example:**<br>`Switch# show ipv6 mfib count` | Displays summary traffic statistics from the MFIB about the group and source. |

| | Command | Purpose |
|---|---|---|
| **Step 4** | **show ipv6 mfib interface**<br>**Example:**<br>`Switch# show ipv6 mfib interface` | Displays information about IPv6 multicast-enabled interfaces and their forwarding status. |
| **Step 5** | **show ipv6 mfib status**<br>**Example:**<br>`Switch# show ipv6 mfib status` | Displays general MFIB configuration and operational status. |
| **Step 6** | **show ipv6 mfib** [**vrf** *vrf-name*] **summary**<br>**Example:**<br>`Switch# show ipv6 mfib summary` | Displays summary information about the number of IPv6 MFIB entries and interfaces. |
| **Step 7** | **debug ipv6 mfib** [**vrf** *vrf-name*] [*group-name*\| *group-address*] [**adjacency** \| **db** \| **fs** \| **init** \| **interface** \| **mrib** [**detail**] \| **nat** \| **pak** \| **platform** \| **ppr** \| **ps** \| **signal** \| **table**]<br>**Example:**<br>`Switch# debug ipv6 mfib FF04::10 pak` | Enables debugging output on the IPv6 MFIB. |
| **Step 8** | **copy running-config startup-config** | (Optional) Save your entries in the configuration file. |

## Resetting MFIB Traffic Counters

Beginning in privileged EXEC mode, follow these steps:

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