Implementing IPv6 Multicast

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 Multicast Listener Discovery (MLD) protocol.

Switches use the MLD protocol to learn whether members of a group are present on their directly attached subnets. Hosts join multicast groups by sending MLD report messages. 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 Multicast is not supported on port-channels.

IPv6 Multicast Routing Implementation

The Cisco IOS-XE 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.

Note MLDv2 is not fully supported as part of this release. MLDv2 Join/Report alone, is supported as part of the V6 SSM feature.

- 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 packetsfrom specific source addresses (or from all but the specific source addresses) to an IP multicast address.

IPv6 Multicast Listener Discovery Protocol

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.

Multicast Queriers and Hosts

A multicast 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 multicast host is a receiver which reports host memberships.

A set of queriers and hosts is called a multicast group. Queriers and hosts use MLD reports to join and leave multicast groups and to begin receiving group traffic.

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

MLD Access Group

The MLD access group provides receiver access control in 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 PIM-SM operation.

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 Rendezvous Point (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 Bootstrap Router (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 is 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.

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.

A typical situation which can lead to this is 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.

Rendezvous Point

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

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

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

Note The **ipv6 pim rp embedded** command is enabled by default.

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

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

The RP address is used by first-hop devices to send PIM register messages on behalf of a host sending a packet to the group. The RP address is also used by last-hop devices to send PIM join and prune messages to the RP to inform it about group membership. You must configure the RP address on all devices (including the RP device).

A PIM device can be an RP for more than one group. Only one RP address can be used at a time within a PIM domain for a certain group. The conditions specified by the access list determine for which groups the device is an RP.

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

Note Dynamic RPs have higher preference over Static RP's

Configuring a BSR

Follow these steps to configure the Bootstrap Router (BSR). The BSR distributes group-to-RP mapping information rapidly throughout a domain.

Configuring a BSR and Verifying BSR Information

To configure and verify BSR information, perform this procedure:

Procedure

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

	Command or Action	Purpose
	`> enable`	
Step 2	**configure terminal** **Example:** `# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 pim bsr candidate bsr** *ipv6-address*[*hash-mask-length*] [**priority** *priority-value*] **Example:** `(config)# ipv6 pim bsr candidate bsr 2001:DB8:3000:3000::42 124 priority 10`	Configures a switch to be a candidate BSR.
Step 4	**interface** *type number* **Example:** `(config)# interface GigabitEthernet 1/0/1`	Specifies an interface type and number, and places the switch in interface configuration mode.
Step 5	**ipv6 pim bsr border** **Example:** `(config-if)# ipv6 pim bsr border`	Configures a border for all bootstrap message (BSMs) of any scope on a specified interface.
Step 6	**exit** **Example:** `(config-if)# exit`	Enter this command twice to exit interface configuration mode and enter privileged EXEC mode.

Step 7	**show ipv6 pim** bsr {**election** | **rp-cache** | **candidate-rp**}
	Example:
	`(config-if)# show ipv6 pim bsr election`
		Displays information related to PIM BSR protocol processing.

Step 8	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Sending PIM RP Advertisements to the BSR

To sending PIM RP advertisements to the BSR, perform this procedure:

Procedure

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

	Command or Action	Purpose
	`> enable`	
Step 2	**configure terminal** **Example:** `# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 pim bsr candidate rp** *ipv6-address* [**group-list** *access-list-name*] [**priority** *priority-value*] [**interval** seconds] **Example:** `(config)# ipv6 pim bsr candidate rp 2001:DB8:3000:3000::42 priority 0`	Sends PIM RP advertisements to the BSR.
Step 4	**interface** *type number* **Example:** `(config)# interface GigabitEthernet 1/0/1`	Specifies an interface type and number, and places the switch in interface configuration mode.
Step 5	**ipv6 pim bsr border** **Example:** `(config-if)# ipv6 pim bsr border`	Configures a border for all BSMs of any scope on a specified interface.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Configuring BSR for Use Within Scoped Zones

To configure BSR for use within scoped zones, perform this procedure:

Procedure

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

	Command or Action	Purpose
Step 3	**ipv6 pim bsr candidate rp** *ipv6-address* [*hash-mask-length*] [**priority** priority-value] **Example:** `(config)# ipv6 pim bsr candidate bsr 2001:DB8:1:1:4`	Configures a switch to be a candidate BSR.
Step 4	**ipv6 pim bsr candidate rp** *ipv6-address* [**group-list** *access-list-name*] [**priority** *priority-value*] [**interval** seconds] **Example:** `(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 5	**interface** *type number* **Example:** `(config-if)# interface GigabitEthernet 1/0/1`	Specifies an interface type and number, and places the switch in interface configuration mode.
Step 6	**ipv6 multicast boundary scope** *scope-value* **Example:** `(config-if)# ipv6 multicast boundary scope 6`	Configures a multicast boundary on the interface for a specified scope.
Step 7	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

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.

To configure BSR switches to announce Scope-to-RP mappings, perform this procedure:

Procedure

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

	Command or Action	Purpose
	`# configure terminal`	
Step 3	**ipv6 pim bsr announced rp** *ipv6-address* [**group-list** *access-list-name*] [**priority** *priority-value*] **Example:** `(config)# ipv6 pim bsr announced rp 2001:DB8:3000:3000::42 priority 0`	Announces scope-to-RP mappings directly from the BSR for the specified candidate RP.
Step 4	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

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-XE 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.

Note V6 PIM register tunnels MTU created by MFIB are limited to a max of 1452 bytes. Therefore, jumbo frames over V6 PIM tunnel are not supported.

Distributed MFIB

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

MFIB implements the following functions:

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

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

IPv6 Multicast Process Switching and Fast Switching

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

IPv6 multicast fast switching 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

To enable IPv6 multicast routing, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal**	Enter global configuration mode.
Step 3	**ipv6 multicast-routing** **Example:** `(config)# ipv6 multicast-routing`	Enables multicast routing on all IPv6-enabled interfaces and enables multicast forwarding for PIM and MLD on all enabled interfaces of the switch.
Step 4	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Customizing and Verifying the MLD Protocol

Customizing and Verifying MLD on an Interface

To customize and verify MLD on an interface, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `(config)# interface GigabitEthernet 1/0/1`	Specifies an interface type and number, and places the switch in interface configuration mode.
Step 4	**ipv6 mld join-group** [*group-address*] [**include** | **exclude**] {*source-address* | **source-list** [*acl*]}	Configures MLD reporting for a specified group and source.

	Command or Action	Purpose
	Example: `(config-if)# ipv6 mld join-group FF04::10`	**Note** join-group and static-group should be used only in a debugging environment, as the packets matching this group will reach CPU for processing.
Step 5	**ipv6 mld access-group** *access-list-name* **Example:** `(config-if)# ipv6 access-list acc-grp-1`	Allows the user to perform IPv6 multicast receiver access control.

Step 6	**ipv6 mld static-group** [*group-address*] [**include** | **exclude**] {*source-address* | source-list [*acl*]}
	Example:
	`(config-if)# ipv6 mld static-group ff04::10 include 100::1`
		Statically forwards traffic for the multicast group onto a specified interface and cause the interface to behave as if a MLD joiner were present on the interface.

Step 7	**ipv6 mld query-max-response-time** *seconds* **Example:** `(config-if)# ipv6 mld query-timeout 130`	Configures the timeout value before the switch takes over as the querier for the interface.
Step 8	**exit** **Example:** `(config-if)# exit`	Enter this command twice to exit interface configuration mode and enter privileged EXEC mode.
Step 9	**show ipv6 mld groups [link-local]** [*group-name* | *group-address*] [*interface-type interface-number*] [**detail** | **explicit**] **Example:** `# show ipv6 mld groups GigabitEthernet 1/0/1`	Displays the multicast groups that are directly connected to the switch and that were learned through MLD.
Step 10	**show ipv6 mld groups summary** **Example:** `# 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 interface** [*type number*] **Example:** `# show ipv6 mld interface GigabitEthernet 1/0/1`	Displays multicast-related information about an interface.
Step 12	**debug ipv6 mld** [*group-name* | *group-address* | *interface-type*] **Example:**	Enables debugging on MLD protocol activity.

	Command or Action	Purpose
	`# debug ipv6 mld`	
Step 13	**debug ipv6 mld explicit** [*group-name* | *group-address* **Example:** `# 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

To implement MLD group limits globally, perform this procedure:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 mld state-limit** *number*
4. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Switch Controller Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Switch Controller Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 mld state-limit** *number* **Example:** `Switch Controller Device(config)# ipv6 mld state-limit 300`	Limits the number of MLD states globally.
Step 4	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Implementing MLD Group Limits per Interface

To implement MLD group limits per interface, perform this procedure:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface type** *number*
4. **ipv6 mld limit** *number* [**except**]*access-list*
5. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `# configure terminal`	Enters global configuration mode.
Step 3	**interface type** *number* **Example:** `(config)# interface GigabitEthernet 1/0/1`	Specifies an interface type and number, and places the switch in interface configuration mode.
Step 4	**ipv6 mld limit** *number* [**except**]*access-list* **Example:** `(config-if)# ipv6 mld limit 100`	Limits the number of MLD states on a per-interface basis.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

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.

To configuring explicit tracking of receivers to track host behavior, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal**	Enter global configuration mode.
Step 3	**interface** *type number* **Example:** `(config)# interface GigabitEthernet 1/0/1`	Specifies an interface type and number, and places the switch in interface configuration mode.
Step 4	**ipv6 mld explicit-tracking** *access-list-name* **Example:** `(config-if)# ipv6 mld explicit-tracking list1`	Enables explicit tracking of hosts.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Resetting the MLD Traffic Counters

To reset the MLD traffic counters, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `# configure terminal`	Enters global configuration mode.
Step 3	**clear ipv6 mld traffic** **Example:** `# clear ipv6 mld traffic`	Resets all MLD traffic counters.
Step 4	**show ipv6 mld traffic** **Example:** `# show ipv6 mld traffic`	Displays the MLD traffic counters.

	Command or Action	Purpose
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Clearing the MLD Interface Counters

To clearing the MLD interface counters, perform this procedure

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `# configure terminal`	Enters global configuration mode.
Step 3	**clear ipv6 mld counters** *interface-type* **Example:** `# clear ipv6 mld counters Ethernet1/0`	Clears the MLD interface counters.
Step 4	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Configuring PIM

This section explains how to configure PIM.

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

To configuring PIM-SM and view PIM-SM information for a group range, perform this procedure:

Procedure

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

	Command or Action	Purpose
	`# configure terminal`	
Step 3	ipv6 pim **rp-address** *ipv6-address*[*group-access-list*] **Example:** `(config)# ipv6 pim rp-address 2001:DB8::01:800:200E:8C6C acc-grp-1`	Configures the address of a PIM RP for a particular group range.
Step 4	**exit** **Example:** `(config)# exit`	Exits global configuration mode, and returns the switch to privileged EXEC mode.
Step 5	**show ipv6 pim interface** [state-on] [**state-off**] [*type-number*] **Example:** `# show ipv6 pim interface`	Displays information about interfaces configured for PIM.
Step 6	**show ipv6 pim group-map** [*group-name* | group-address] | [*group-range* | *group-mask*] [**info-source {bsr** | **default** | **embedded-rp** | **static**}] **Example:** `# show ipv6 pim group-map`	Displays an IPv6 multicast group mapping table.
Step 7	**show ipv6 pim neighbor** [**detail**] [*interface-type interface-number* | **count**] **Example:** `# show ipv6 pim neighbor`	Displays the PIM neighbors discovered by the Cisco IOS software.
Step 8	**show ipv6 pim range-list** [**config**] [*rp-address* | *rp-name*] **Example:** `# show ipv6 pim range-list`	Displays information about IPv6 multicast range lists.
Step 9	**show ipv6 pim tunnel** [*interface-type interface-number*] **Example:** `# show ipv6 pim tunnel`	Displays information about the PIM register encapsulation and de-encapsulation tunnels on an interface.
Step 10	**debug ipv6 pim** [*group-name* | *group-address* | **interface** *interface-type* | **bsr** | **group** | **mvpn** | **neighbor**] **Example:**	Enables debugging on PIM protocol activity.

	Command or Action	Purpose
	`# debug ipv6 pim`	
Step 11	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Configuring PIM Options

To configure PIM options, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Switch Controller Device > enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Switch Controller Device # configure terminal`	Enters global configuration mode.
Step 3	**ipv6 pim spt-threshold infinity** [**group-list** *access-list-name*] **Example:** `Switch Controller Device (config)# ipv6 pim spt-threshold infinity group-list acc-grp-1`	Configures when a PIM leaf switch joins the SPT for the specified groups.
Step 4	**interface** *type number* **Example:** `Switch Controller Device (config)# interface GigabitEthernet 1/0/1`	Specifies an interface type and number, and places the switch in interface configuration mode.
Step 5	**ipv6 pim dr-priority** *value* **Example:** `Switch Controller Device (config-if)# ipv6 pim dr-priority 3`	Configures the DR priority on a PIM switch.
Step 6	**ipv6 pim hello-interval** *seconds* **Example:** `Switch Controller Device (config-if)# ipv6 pim hello-interval 45`	Configures the frequency of PIM hello messages on an interface.

	Command or Action	Purpose
Step 7	**ipv6 pim join-prune-interval** *seconds* **Example:** `Switch Controller Device (config-if)# ipv6 pim join-prune-interval 75`	Configures periodic join and prune announcement intervals for a specified interface.
Step 8	**exit** **Example:** `Switch Controller Device (config-if)# exit`	Enter this command twice to exit interface configuration mode and enter privileged EXEC mode.
Step 9	**ipv6 pim join-prune statistic** [*interface-type*] **Example:** `Switch Controller Device (config-if)# show ipv6 pim join-prune statistic`	Displays the average join-prune aggregation for the most recently aggregated packets for each interface.
Step 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.

To resetting the PIM traffic counters, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `# configure terminal`	Enters global configuration mode.
Step 3	**clear ipv6 pim traffic** **Example:** `# clear ipv6 pim traffic`	Resets the PIM traffic counters.

	Command or Action	Purpose
Step 4	**show ipv6 pim traffic** **Example:** `# show ipv6 pim traffic`	Displays the PIM traffic counters.
Step 5	**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.

To clear the PIM topology table to reset the MRIB connection, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `# configure terminal`	Enters global configuration mode.
Step 3	**clear ipv6 pim** [*vrf vrfname*]**topology** [*group-name* | *group-address*] **Example:** `# clear ipv6 pim topology FF04::10`	Clears the PIM topology table.
Step 4	**show ipv6 mrib** [*vrf vrfname*]**client** [**filter**] [**name** {*client-name* | *client-name* **: client-id**}] **Example:** `# show ipv6 mrib client`	Displays multicast-related information about an interface.
Step 5	**show ipv6 mrib** [*vrf vrfname*]**route** {**link-local** | **summary** | [*sourceaddress-or-name* | *] [*groupname-or-address*[*prefix-length*]]] **Example:** `# show ipv6 mrib route`	Displays the MRIB route information.

	Command or Action	Purpose
Step 6	**show ipv6 pim** [*vrf vrfname*]**topology** [*groupname-or-address* [*sourceaddress-or-name*] | **link-local** | **route-count** [**detail**]] **Example:** `# show ipv6 pim topology`	Displays PIM topology table information for a specific group or all groups.
Step 7	**debug ipv6 mrib** [*vrf vrfname*]**io** **Example:** `# debug ipv6 mrib io`	Enables debugging on MRIB I/O events.
Step 8	**debug ipv6 mrib** [*vrf vrfname*]**route** [*group-name* | *group-address*] **Example:** `# debug ipv6 mrib route`	Displays information about MRIB routing entry-related activity.
Step 9	**debug ipv6 mrib** [*vrf vrfname*]**table** **Example:** `# 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 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.

To configure static mroutes, perform this procedure:

Procedure

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

	Command or Action	Purpose
Step 3	**ipv6 route** {*ipv6-prefix / prefix-length ipv6-address* | *interface-type interface-number ipv6-address*]} [*administrative-distance*] [*administrative-multicast-distance* | *unicast* | *multicast*] [**tag** *tag*] **Example:** `(config)# ipv6 route 2001:DB8::/64 6::6 100`	Establishes static IPv6 routes. The example shows a static route used for both unicast routing and multicast RPF selection.
Step 4	**exit** **Example:** `# exit`	Exits global configuration mode, and returns the switch to privileged EXEC mode.
Step 5	**show ipv6 mroute** [link-local | [*group-name* | *group-address* [*source-address* | *source-name*]] [**summary**] [**count**] **Example:** `# show ipv6 mroute ff07::1`	Displays the contents of the IPv6 multicast routing table.
Step 6	**show ipv6 mroute** [**link-local** | *group-name* | *group-address*] **active** [*kbps*] **Example:** `(config-if)# show ipv6 mroute active`	Displays the active multicast streams on the switch.
Step 7	**show ipv6 rpf** [*ipv6-prefix*] **Example:** `(config-if)#  show ipv6 rpf 2001::1:1:2`	Checks RPF information for a given unicast host address and prefix.
Step 8	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Using MFIB in IPv6 Multicast

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

Verifying MFIB Operation in IPv6 Multicast

To verify MFIB operation in IPv6 multicast

Procedure

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

	Command or Action	Purpose
	`> enable`	
Step 2	**show ipv6 mfib** [| **verbose**| *vrf vrfname* | *group-address-name* | *ipv6-prefix / prefix-length* | *source-address-name* | **count** | **interface** | **status** | **summary**] **Example:** `# show ipv6 mfib`	Displays the forwarding entries and interfaces in the IPv6 MFIB.
Step 3	**show ipv6 mfib** [all | linkscope | vrf vrfname | group-name | group-address [source-name | source-address]] **count** **Example:** `# show ipv6 mfib ff07::1`	Displays the contents of the IPv6 multicast routing table.
Step 4	**show ipv6 mfib interface** **Example:** `# show ipv6 mfib interface`	Displays information about IPv6 multicast-enabled interfaces and their forwarding status.
Step 5	**show ipv6 mfib status** **Example:** `# show ipv6 mfib status`	Displays general MFIB configuration and operational status.
Step 6	**show ipv6 mfib summary** **Example:** `# show ipv6 mfib summary`	Displays summary information about the number of IPv6 MFIB entries and interfaces.
Step 7	**debug ipv6 mfib** [*vrf vrfname* |*group-name* | *group-address*] [**adjacency** | **db** | **fs** | **init** | **interface** | **mrib** [**detail**] | **nat** | **pak** | **platform** | **ppr** | **ps** | **signal** | **table**] **Example:** `# debug ipv6 mfib FF04::10 pak`	Enables debugging output on the IPv6 MFIB.

Resetting MFIB Traffic Counters

To reset MFIB traffic counters, perform this procedure:

Procedure

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

	Command or Action	Purpose
	Example: `> enable`	Enter your password if prompted.
Step 2	**clear ipv6 mfib counters** [*group-name* | **group-address** [*source-address* | *source-name*]] **Example:** `# clear ipv6 mfib counters FF04::10`	Resets all active MFIB traffic counters.