IPv6 Multicast Routing

Feature history for IPv6 multicast

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

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

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

Understand IPv6 multicast

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

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

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

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

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

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

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

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

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

IPv6 multicast routing implementation

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

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

  MLD version 1 is based on version 2 of the Internet Group Management Protocol (IGMP) for IPv4. MLD version 2 is based on version 3 of IGMP for IPv4. Cisco IOS software uses both MLD version 2 and MLD version 1 for IPv6 multicast. MLD version 2 is fully backward-compatible with MLD version 1 (described in RFC 2710). Hosts that support only MLD version 1 will interoperate with a switch running MLD version 2. LANs with both MLD version 1 and version 2 hosts are supported.

- PIM-SM operates between switches to track multicast packets for forwarding to directly connected LANs.

- PIM in Source Specific Multicast (PIM-SSM) is similar to PIM-SM. It adds the ability to report interest in packets from specific source addresses or exclude specific source addresses to an IP multicast address.

IPv6 multicast listener discovery protocol

To implement multicasting in the campus network, you need to define the multicast recipients. The MLD protocol is used by IPv6 switches to discover the presence of multicast listeners (for example, nodes that want to receive multicast packets) on their directly attached links, and to discover specifically which multicast addresses are of interest to those neighboring nodes. MLD protocol can be used to discover local group and source-specific group membership.

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

Multicast queriers and hosts

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

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

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

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

MLD access group

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

Explicit tracking

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

Protocol independent multicast

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

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

Configure IPv6 multicast to use PIM-SM, PIM-SSM, or both PIM-SM and PIM-SSM together in your network.

PIM-sparse mode

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

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

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

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

A multicast data sender sends data destined for a multicast group. The designated switch (DR) of the sender takes those data packets, unicast-encapsulates them, and sends them directly to the RP. The RP receives these encapsulated data packets, de-encapsulates them, and forwards them onto the shared tree.

The packets then follow the (*, G) multicast tree state in the switches on the RP tree, being replicated wherever the RP tree branches, and eventually reaching all the receivers for that multicast group. The process of encapsulating data packets to the RP is called registering, and the encapsulation packets are called PIM register packets.

IPv6 BSR RP mapping

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

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

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

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

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

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

PIM-source specific multicast

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

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

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

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

Routable address hello option

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

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

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

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

PIM IPv6 stub routing

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

For IPv6 traffic in a PIM stub routing network, the only allowable route to the user is through a switch configured with PIM stub routing. PIM passive interfaces are connected to Layer 2 access domains, such as VLANs, or to interfaces that are connected to other Layer 2 devices. Only directly connected multicast receivers and sources are allowed in the Layer 2 access domains. The PIM passive interfaces do not send or process any received PIM control packets.

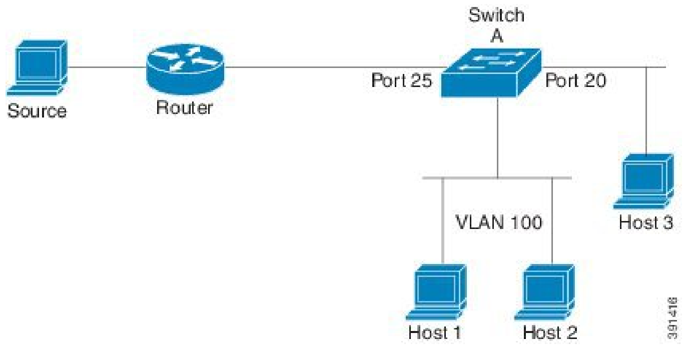
Configure the distribution and remote routers to use IPv6 multicast routing. Set the switch as a PIM stub router. Transit traffic between distribution routers cannot be routed using the switch. You also need to configure a routed uplink port on the switch. The switch uplink port cannot be used with SVIs.

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

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

The configuration involves Switch A routed uplink port 25 connected to the router, with PIM stub routing enabled on the VLAN 100 interfaces and on Host 3. This configuration allows directly connected hosts to receive traffic from the multicast source.

Figure 1: PIM stub router configuration



IPv6 multicast process switching and fast switching

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

To enhance packet forwarding performance, use IPv6 multicast fast switching instead of process switching. In IPv6 multicast switching, information that is usually stored in a route cache is stored in several data structures. The data structures provide optimized lookup for efficient packet forwarding.

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

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

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

Multiprotocol BGP for the IPv6 multicast address family

Multiprotocol BGP for IPv6 multicast has extensions similar to those of IPv4 BGP. These IPv6 enhancements to multicast BGP include support for an IPv6 multicast address family, network layer reachability information (NLRI), and next-hop attributes that use IPv6 addresses.

Multicast BGP is an enhanced BGP that allows the deployment of interdomain IPv6 multicast. Multiprotocol BGP carries routing information, such as IPv6 address family and IPv6 multicast routes. The IPv6 multicast address family contains routes used for RPF lookup by the IPv6 PIM protocol. Additionally, multicast BGP IPv6 provides interdomain transport for these routes. Use multiprotocol BGP for IPv6 multicast if you are using IPv6 multicast with BGP. Unicast BGP learned routes won't be used for IPv6 multicast.

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

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

No MRIB is associated with the IPv6 multicast BGP table. However, Pv6 multicast BGP processes the unicast IPv6 RIB when necessary. Multicast BGP does not insert or update routes into the IPv6 unicast RIB.

Embedded RP

IPv6 PIM provides embedded RP support. Embedded RP support allows the device to learn RP information using the multicast group destination address instead of the statically configured RP. If a device serves as the RP, configure it statically as the RP.

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

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

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

- The first-hop device encapsulates data in register packets and unicasts it directly to the RP while operating as the DR.
- The RPF forwarding algorithm, as described in the PIM-Sparse Mode section, dictates multicast forwarding if the RP has joined the source tree.

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

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

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

Static mroutes

IPv6 static mroutes function in a similar manner to IPv4 static mroutes, influencing the RPF check. IPv6 static mroutes share the same database as IPv6 static routes. They also extend static route support for RPF checks. Static mroutes support equal-cost multipath mroutes and unicast-only static routes.

MRIB

The Multicast Routing Information Base (MRIB) is a protocol-independent repository of multicast routing entries instantiated by multicast routing protocols (routing clients). ts primary function is to allow routing protocols to operate independently from the Multicast Forwarding Information Base (MFIB). It also acts as a coordination and communication point among its clients.

Routing clients use the services provided by the MRIB to instantiate routing entries and retrieve changes made to routing entries by other clients. Besides routing clients, MRIB accommodates both forwarding clients (MFIB instances) and special clients like MLD. MFIB retrieves forwarding entries from MRIB and notifies MRIB about events related to packet reception. Routing clients can request notifications, or MFIB can generate them independently.

To coordinate multiple routing clients for multicast connectivity within the same session, utilize MRIB. Additionally, MRIB facilitates coordination between MLD and routing protocols.

MFIB

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

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

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

Configure IPv6 multicast

This section explains how to configure IPv6 multicast.

Enable IPv6 multicast routing

Perform this procedure to enable IPv6 multicast routing:

Procedure

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

Customize and verify the MLD protocol

This section explains how to configure and verify the MLD protocol.

Customize and verify MLD on an interface

Perform this procedure to customize and verify MLD on an interface:

Procedure

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

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 1/0/1`	Specifies an interface type and number, and places the switch in interface configuration mode.

Step 4	**ipv6 mld join-group** [*group-address*] [**include** | **exclude**] {*source-address* | **source-list** [*acl*]}
	Example:
	`Device(config-if)# ipv6 mld join-group FF04::10`
		Configures MLD reporting for a specified group and source.

Step 5	**ipv6 mld access-group** *access-list-name* **Example:** `Device(config-if)# ipv6 access-list acc-grp-1`	Allows the user to perform IPv6 multicast receiver access control.

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

Step 7	**ipv6 mld query-max-response-time** *seconds* **Example:** `Device(config-if)# ipv6 mld query-timeout 130`	Configures the timeout value before the switch takes over as the querier for the interface.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Enter this command twice to exit interface configuration mode and enter privileged EXEC mode.
Step 9	**show ipv6 mld groups [link-local]** [*group-name* | *group-address*] [*interface-type interface-number*] [**detail** | **explicit**] **Example:** `Device# show ipv6 mld groups GigabitEthernet 1/0/1`	Displays the multicast groups that are directly connected to the switch and that were learned through MLD.
Step 10	**show ipv6 mld groups summary** **Example:**	Displays the number of (*, G) and (S, G) membership reports present in the MLD cache.

	Command or Action	Purpose
	`Device# show ipv6 mld groups summary`	
Step 11	**show ipv6 mld interface** [*type number*] **Example:** `Device# show ipv6 mld interface GigabitEthernet 1/0/1`	Displays multicast-related information about an interface.
Step 12	**debug ipv6 mld** [*group-name* | *group-address* | *interface-type*] **Example:** `Device# debug ipv6 mld`	Enables debugging on MLD protocol activity.
Step 13	**debug ipv6 mld explicit** [*group-name* | *group-address*] **Example:** `Device# debug ipv6 mld explicit`	Displays information related to the explicit tracking of hosts.
Step 14	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Implement MLD group limits

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

Implement MLD group limits globally

Perform this procedure to implement MLD group limits globally:

Procedure

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

	Command or Action	Purpose
Step 3	**ipv6 mld** [*vrf vrf-name*] **state-limit** *number* **Example:** `Device(config)# ipv6 mld state-limit 300`	Limits the number of MLD states globally.
Step 4	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Implement MLD group limits per interface

Perform this procedure to implement MLD group limits per interface:

Procedure

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

Configure explicit tracking of receivers to track host behavior

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

Perform this procedure to configuring explicit tracking of receivers to track host behavior:

Procedure

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

Reset the MLD traffic counters

Perform this procedure to reset the MLD traffic counters:

Procedure

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

	Command or Action	Purpose
Step 3	**clear ipv6 mld traffic** **Example:** `Device# clear ipv6 mld traffic`	Resets all MLD traffic counters.
Step 4	**show ipv6 mld traffic** **Example:** `Device# show ipv6 mld traffic`	Displays the MLD traffic counters.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Clear the MLD interface counters

Perform this procedure to clearing the MLD interface counters:

Procedure

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

Configuring PIM

This section explains how to configure PIM.

Configure PIM-SM and display PIM-SM information for a group range

Perform this procedure to configuring PIM-SM and view PIM-SM information for a group range:

Procedure

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

	Command or Action	Purpose
Step 8	**show ipv6 pim range-list** [**config**] [*rp-address* | *rp-name*] **Example:** `Device# show ipv6 pim range-list`	Displays information about IPv6 multicast range lists.
Step 9	**show ipv6 pim tunnel** [*interface-type interface-number*] **Example:** `Device# show ipv6 pim tunnel`	Displays information about the PIM register encapsulation and de-encapsulation tunnels on an interface.
Step 10	**debug ipv6 pim** [*group-name* | *group-address* | **interface** *interface-type* | **bsr** | **group** | **mvpn** | **neighbor**] **Example:** `Device# debug ipv6 pim`	Enables debugging on PIM protocol activity.
Step 11	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Configure PIM options

Perform this procedure to configure PIM options:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 pim spt-threshold infinity** [**group-list** *access-list-name*] **Example:** `Device(config)# ipv6 pim spt-threshold infinity group-list acc-grp-1`	Configures when a PIM leaf switch joins the SPT for the specified groups.

	Command or Action	Purpose

Step 4	**ipv6 pim accept-register** {**list** *access-list* | **route-map** *map-name*}
	Example:
	`Device(config)# ipv6 pim accept-register route-map reg-filter`
		Accepts or rejects registers at the RP.

Step 5	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 1/0/1`	Specifies an interface type and number, and places the switch in interface configuration mode.
Step 6	**ipv6 pim dr-priority** *value* **Example:** `Device(config-if)# ipv6 pim dr-priority 3`	Configures the DR priority on a PIM switch.
Step 7	**ipv6 pim hello-interval** *seconds* **Example:** `Device(config-if)# ipv6 pim hello-interval 45`	Configures the frequency of PIM hello messages on an interface.
Step 8	**ipv6 pim join-prune-interval** *seconds* **Example:** `Device(config-if)# ipv6 pim join-prune-interval 75`	Configures periodic join and prune announcement intervals for a specified interface.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Enter this command twice to exit interface configuration mode and enter privileged EXEC mode.
Step 10	**ipv6 pim join-prune statistic** [*interface-type*] **Example:** `Device(config-if)# show ipv6 pim join-prune statistic`	Displays the average join-prune aggregation for the most recently aggregated packets for each interface.
Step 11	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Reset PIM traffic counters

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

Perform this procedure to reset the PIM traffic counters:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**clear ipv6 pim traffic** **Example:** `Device# clear ipv6 pim traffic`	Resets the PIM traffic counters.
Step 4	**show ipv6 pim traffic** **Example:** `Device# show ipv6 pim traffic`	Displays the PIM traffic counters.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Clear PIM topology table to reset MRIB connection

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

Perform this procedure to clear the PIM topology table to reset the MRIB connection:

Procedure

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

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**clear ipv6 pim topology** [*group-name* | *group-address*] **Example:** `Device# clear ipv6 pim topology FF04::10`	Clears the PIM topology table.
Step 4	**show ipv6 mrib client** [**filter**] [**name** {*client-name* | *client-name* **: client-id**}] **Example:** `Device# show ipv6 mrib client`	Displays multicast-related information about an interface.
Step 5	**show ipv6 mrib route** {**link-local** | **summary** | [*sourceaddress-or-name* | *] [*groupname-or-address* [*prefix-length*]]] **Example:** `Device# show ipv6 mrib route`	Displays the MRIB route information.
Step 6	**show ipv6 pim topology** [*groupname-or-address* [*sourceaddress-or-name*] | **link-local** | **route-count** [**detail**]] **Example:** `Device# show ipv6 pim topology`	Displays PIM topology table information for a specific group or all groups.
Step 7	**debug ipv6 mrib client** **Example:** `Device# debug ipv6 mrib client`	Enables debugging on MRIB client management activity.
Step 8	**debug ipv6 mrib io** **Example:** `Device# debug ipv6 mrib io`	Enables debugging on MRIB I/O events.

	Command or Action	Purpose
Step 9	**debug ipv6 mrib proxy** **Example:** `Device# debug ipv6 mrib proxy`	Enables debugging on MRIB proxy activity between the switch processor and line cards on distributed switch platforms.
Step 10	**debug ipv6 mrib route** [*group-name* | *group-address*] **Example:** `Device# debug ipv6 mrib route`	Displays information about MRIB routing entry-related activity.
Step 11	**debug ipv6 mrib table** **Example:** `Device# debug ipv6 mrib table`	Enables debugging on MRIB table management activity.
Step 12	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Configure PIM IPv6 stub routing

The PIM Stub routing feature supports multicast routing between the distribution layer and the access layer. It supports two types of PIM interfaces: uplink PIM interfaces and PIM passive interfaces. A routed interface configured with the PIM passive mode does not pass or forward PIM control traffic. It only passes and forwards MLD traffic.

PIM IPv6 stub routing configuration guidelines

- Before configuring PIM stub routing, you must have IPv6 multicast routing configured on both the stub router and the central router. PIM mode (sparse-mode) must be configured on the uplink interface of the stub router.
- The PIM stub router prevents the routing of transit traffic between distribution routers. Unicast (EIGRP) stub routing enforces this behavior. Configure unicast stub routing to support PIM stub router behavior. For more information, see the *EIGRP Stub Routing* section.
- Only directly connected multicast (MLD) receivers and sources are allowed in the Layer 2 access domains. The PIM protocol is not supported in access domains.
- The redundant PIM stub router topology is not supported.

Default IPv6 PIM routing configuration

This table displays the default IPv6 PIM routing configuration.

Table 1: Default multicast routing configuration

Feature	Default Setting
Multicast routing	Disabled on all interfaces.

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

Enable IPV6 PIM stub routing

Perform this procedure to enable IPV6 PIM stub routing:

Before you begin

PIM stub routing is disabled in IPv6 by default.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 multicast pim-passive-enable** **Example:** `Device(config-if)#  ipv6 multicast pim-passive-enable`	Enables IPv6 Multicast PIM routing on the switch.

	Command or Action	Purpose
Step 4	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet 9/0/6`	Specifies the interface on which you want to enable PIM stub routing, and enters interface configuration mode. The specified interface must be one of the following: • A routed port: A physical port that has been configured as a Layer 3 port by entering the **no switchport** interface configuration command. You will also need to enable IP PIM sparse mode on the interface, and join the interface as a statically connected member to an MLD static group. • An SVI: A VLAN interface created by using the **interface vlan** *vlan-id* global configuration command. You will also need to enable IP PIM sparse mode on the VLAN, join the VLAN as a statically connected member to an MLD static group, and then enable MLD snooping on the VLAN, the MLD static group, and physical interface. These interfaces must have IPv6 addresses assigned to them.
Step 5	**ipv6 pim** **Example:** `Device(config-if)# ipv6 pim`	Enables the PIM on the interface.

Step 6	**ipv6 pim** {**bsr** | {**dr-priority** | *value*} | {**hello-interval** | *seconds*} | {**join-prune-interval** | *seconds*} | **passive**}
	Example:
	`Device(config-if)# ipv6 pim bsr|dr-priority|hello-interval|join-prune-interval|passive`
		Configures the various PIM stub features on the interface.
		Enter **bsr** to configure BSR on a PIM switch
		Enter **dr-priority** to configure the DR priority on a PIM switch.
		Enter **hello-interval** to configure the frequency of PIM hello messages on an interface.
		Enter **join-prune-interval** to configure periodic join and prune announcement intervals for a specified interface.
		Enter **passive** to configure the PIM in the passive mode.

	Command or Action	Purpose

Step 7	**end**
	Example:
	`Device(config-if)# end`
		Returns to privileged EXEC mode.

Disable embedded RP support in IPv6 PIM

A user might want to disable embedded RP support on an interface if all of the devices in the domain do not support embedded RP.

Note This task disables PIM completely, not just embedded RP support in IPv6 PIM.

Perform this procedure to disable embedded RP support in IPv6 PIM:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**no ipv6 pim** [**vrf** *vrf-name*] **rp embedded** **Example:** `Device(config)# no ipv6 pim rp embedded`	Disables embedded RP support in IPv6 PIM.
Step 4	**interface** *type number* **Example:** `Device(config)# interface FastEthernet 1/0`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 5	**no ipv6 pim** **Example:** `Device(config-if)# no ipv6 pim`	Turns off IPv6 PIM on a specified interface.

Monitor IPv6 PIM stub routing

Table 2: PIM stub configuration show commands

Command	Purpose
show ipv6 pim interface	Displays the PIM stub that is enabled on each interface.
show ipv6 mld groups	Displays the interested clients that have joined the specific multicast source group.
show ipv6 mroute	Verifies that the multicast stream forwards from the source to the interested clients.

Configuring a BSR

This section explains how to configure BSR.

Configure a BSR and verify BSR information

Perform this procedure to configure and verify BSR Information:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 pim bsr candidate bsr** *ipv6-address* [*hash-mask-length*] [**priority** *priority-value*] **Example:** `Device(config)# ipv6 pim bsr candidate bsr 2001:DB8:3000:3000::42 124 priority 10`	Configures a switch to be a candidate BSR.
Step 4	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 1/0/1`	Specifies an interface type and number, and places the switch in interface configuration mode.

	Command or Action	Purpose
Step 5	**ipv6 pim bsr border** **Example:** `Device(config-if)# ipv6 pim bsr border`	Specifies an interface type and number, and places the switch in interface configuration mode.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Enter this command twice to exit interface configuration mode and enter privileged EXEC mode.

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

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

Send PIM RP advertisements to the BSR

Perform this procedure to send PIM RP advertisements to the BSR:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 pim bsr candidate rp** *ipv6-address* [**group-list** *access-list-name*] **[priority** *priority-value*] [**interval** seconds] **Example:** `Device(config)# ipv6 pim bsr candidate rp 2001:DB8:3000:3000::42 priority 0`	Sends PIM RP advertisements to the BSR.

	Command or Action	Purpose
Step 4	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 1/0/1`	Specifies an interface type and number, and places the switch in interface configuration mode.
Step 5	**ipv6 pim bsr border** **Example:** `Device(config-if)# ipv6 pim bsr border`	Configures a border for all BSMs of any scope on a specified interface.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Configure BSR for use within scoped zones

Perform this procedure to configure BSR for use within scoped zones:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 pim bsr candidate rp** *ipv6-address* [*hash-mask-length*] [**priority** *priority-value*] **Example:** `Device(config)# ipv6 pim bsr candidate bsr 2001:DB8:1:1:4`	Configures a switch to be a candidate BSR.
Step 4	**ipv6 pim bsr candidate rp** *ipv6-address* [**group-list** *access-list-name*] [**priority** *priority-value*] [**interval** seconds] **Example:** `Device(config)# ipv6 pim bsr candidate rp 2001:DB8:1:1:1 group-list list scope 6`	Configures the candidate RP to send PIM RP advertisements to the BSR.

	Command or Action	Purpose
Step 5	**interface** *type number* **Example:** `Device(config-if)# interface GigabitEthernet 1/0/1`	Specifies an interface type and number, and places the switch in interface configuration mode.
Step 6	**ipv6 multicast boundary scope** *scope-value* **Example:** `Device(config-if)# ipv6 multicast boundary scope 6`	Configures a multicast boundary on the interface for a specified scope.
Step 7	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Configure BSR switches to announce scope-to-RP mappings

IPv6 BSR switches can be statically configured to announce scope-to-RP mappings directly. A user might configure a BSR switch to announce these mappings so that an RP not supporting BSR is incorporated into the BSR. This action also allows an RP located outside the enterprise's BSR domain to be detected by the local candidate BSR switch.

Perform this procedure to configure BSR switches to announce Scope-to-RP mappings:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 pim bsr announced rp** *ipv6-address* [**group-list** *access-list-name*] [**priority** *priority-value*] **Example:** `Device(config)# ipv6 pim bsr announced rp 2001:DB8:3000:3000::42 priority 0`	Announces scope-to-RP mappings directly from the BSR for the specified candidate RP.
Step 4	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Configure static mroutes

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

Perform this procedure to configure static mroutes:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 route** {*ipv6-prefix / prefix-length ipv6-address* | *interface-type interface-number ipv6-address*]} [*administrative-distance*] [*administrative-multicast-distance* | *unicast* | *multicast*] [**tag** *tag*] **Example:** `Device(config)# ipv6 route 2001:DB8::/64 6::6 100`	Establishes static IPv6 routes. The example shows a static route used for both unicast routing and multicast RPF selection.
Step 4	**exit** **Example:** `Device# exit`	Exits global configuration mode, and returns the switch to privileged EXEC mode.
Step 5	**show ipv6 mroute** [link-local | [*group-name* | *group-address* [*source-address* | *source-name*]] **[summary]** [**count**] **Example:** `Device# show ipv6 mroute ff07::1`	Displays the contents of the IPv6 multicast routing table.
Step 6	**show ipv6 mroute** [**link-local** | *group-name* | *group-address*] **active** [*kbps*] **Example:** `Device(config-if)# show ipv6 mroute active`	Displays the active multicast streams on the switch.

	Command or Action	Purpose
Step 7	**show ipv6 rpf** [*ipv6-prefix*] **Example:** `Device(config-if)#  show ipv6 rpf 2001::1:1:2`	Checks RPF information for a given unicast host address and prefix.
Step 8	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Verify MFIB operation in IPv6 multicast

Perform this procedure to verify MFIB operation in IPv6 multicast:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**show ipv6 mfib** [**verbose** | *group-address-name* | *ipv6-prefix / prefix-length* | *source-address-name* | **count** | **interface** | **status** | **summary**] **Example:** `Device# show ipv6 mfib`	Displays the forwarding entries and interfaces in the IPv6 MFIB.
Step 3	**show ipv6 mfib** [**all** | **linkscope** | *group-name* | *group-address* [*source-name* | *source-address*]] **count** **Example:** `Device# show ipv6 mfib ff07::1`	Displays the contents of the IPv6 multicast routing table.
Step 4	**show ipv6 mfib interface** **Example:** `Device# show ipv6 mfib interface`	Displays information about IPv6 multicast-enabled interfaces and their forwarding status.
Step 5	**show ipv6 mfib status** **Example:** `Device# show ipv6 mfib status`	Displays general MFIB configuration and operational status.

	Command or Action	Purpose
Step 6	**show ipv6 mfib summary** **Example:** `Device# show ipv6 mfib summary`	Displays summary information about the number of IPv6 MFIB entries and interfaces.
Step 7	**debug ipv6 mfib** [*group-name* | *group-address*] [**adjacency** | **db** | **fs** | **init** | **interface** | **mrib** [**detail**] | **nat** | **pak** | **platform** | **ppr** | **ps** | **signal** | **table**] **Example:** `Device# debug ipv6 mfib FF04::10 pak`	Enables debugging output on the IPv6 MFIB.

Reset MFIB traffic counters

Perform this procedure to reset MFIB traffic counters:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**clear ipv6 mfib counters** [*group-name* | **group-address** [*source-address* | *source-name*]] **Example:** `Device# clear ipv6 mfib counters FF04::10`	Resets all active MFIB traffic counters.