Multicast Configuration Guide for Cisco 8000 Series Routers, IOS XR Release 24.1.x, 24.2.x, 24.3.x, 24.4.x

First Published: 2023-12-08

Last Modified: 2024-12-16

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Preface

This guide describes the Cisco IOS XR Multicast configurations.

The preface contains the following sections:

- Changes to This Document, on page iii
- Communications, Services, and Additional Information, on page iii

Changes to This Document

Describes the changes in the document from the initial release of this document.

Table 1: Changes to This Document

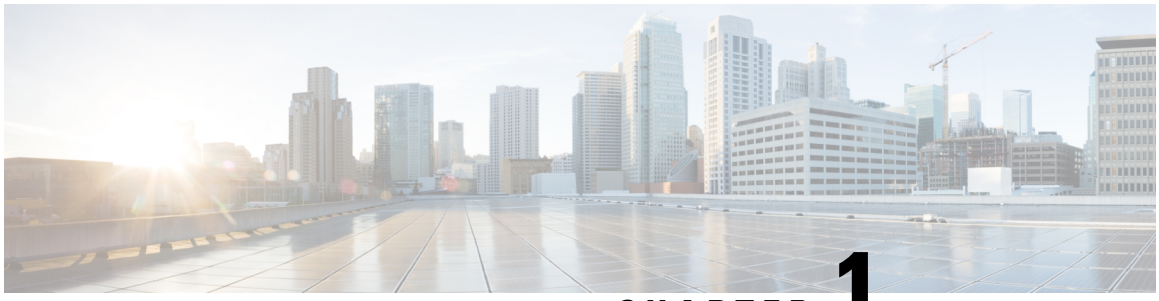
Date	Summary
December 2024	Republished for Release 24.4.1.
February 2024	Initial release of this document.

Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business results you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco DevNet.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.



CHAPTER 1

New and Changed Multicast Features

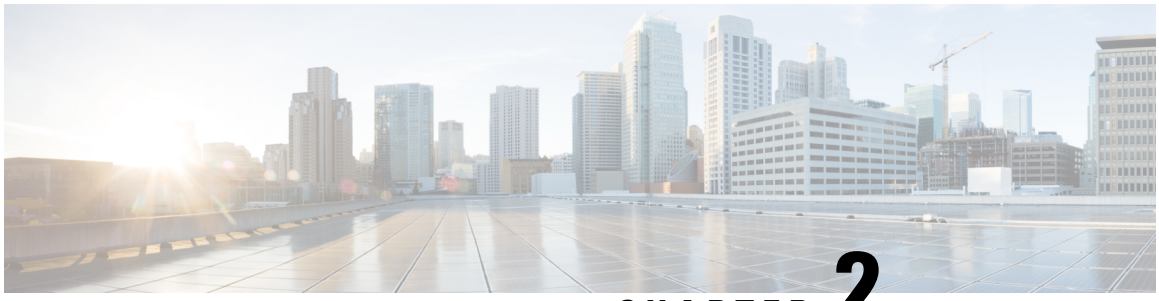
This chapter lists all the features that have been added or modified in this guide. The table also contains references to these feature documentation sections.

- Multicast Features Added or Modified in IOS XR Release 24.x.x, on page 1

Multicast Features Added or Modified in IOS XR Release 24.x.x

Table 2: New and Changed Features

Feature	Description	Changed in Release	Where Documented
LSM mLDP based MVPN bud/tail node enhancements on edge routers	Introduced in this release	Release 24.4.1	LSM mLDP based MVPN bud/tail node enhancements on edge routers
Protection-based MoFRR	Introduced in this release	Release 24.2.11	Protection-based MoFRR, on page 65
Draft-Rosen Multicast VPN for Profiles 0, 3, and 11	Introduced in this release	Release 24.2.11	Rosen-GRE Multicast VPN for Profiles 0, 3, and 11, on page 97
MVPN Ingress Replication Over Dynamic TE-Tunnels	Introduced in this release	Release 24.1.1	MVPN Ingress Replication Over Dynamic TE-Tunnels, on page 111
Profile 22 in Multicast VPN (MVPN) on Edge Routers	Introduced in this release	Release 24.1.1	Profile 22 in Multicast VPN (MVPN) on Edge Routers



CHAPTER 2

Implementing Layer 2 Multicast

- Implementing IGMP Snooping, on page 3
- Supported Features and Restrictions for IGMP Snooping, on page 4
- Information About IGMP Snooping, on page 5
- How to Configure IGMP Snooping, on page 11
- Configuration Examples for IGMP Snooping, on page 17
- MLD Snooping , on page 23
- Creating a MLD Snooping Profile, on page 31
- Deactivating MLD Snooping on a Bridge Domain, on page 32
- Configuring Static Mrouter Ports (MLD), on page 32
- Configuring Immediate-leave for MLD, on page 33
- Configuring Internal Querier for MLD, on page 34
- Configuring Static Groups for MLD, on page 35
- Configuring MLD Snooping, on page 37
- Configuring MLD Snooping on Ethernet Bundles, on page 38

Implementing IGMP Snooping

IGMP snooping provides a way to constrain multicast traffic at Layer 2.

Table 3: Feature History Table

Feature Name	Release Information	Feature Description
IGMP snooping	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100]), (8700 [ASIC: K100])(select variants only*). IGMP Snooping is used in Layer 2 multicast to optimize the distribution of multicast traffic. IGMP membership report messages are examined from hosts to determine which interfaces are connected to devices interested in receiving multicast traffic. This helps in reducing unnecessary traffic by ensuring that multicast data is only sent to ports with interested receivers, rather than flooding the entire VLAN. The benefit of IGMP snooping is bandwidth optimization that limits multicast traffic to only the necessary ports. *This feature is supported on: • 8212-48FH-M • 8712-MOD-M • 8711-32FH-M

Internet Group Management Protocol (IGMP) snooping restricts multicast flows at Layer 2 to only those segments with at least one interested receiver. This module describes how to implement IGMP snooping.

Note Multicast traffic without Spanning-Tree protocol is supported at Layer 2 for multicast traffic without snooping enabled.

Prerequisites for IGMP Snooping

Before implementing IGMP snooping, make sure that the network is configured with a Layer 2 VPN (L2VPN).

Supported Features and Restrictions for IGMP Snooping

- EVPN dual-homed Active Active (AA) IGMP State Sync using IGMP snooping profile is not supported.
- IGMP snooping is supported only under L2VPN bridge domains.

- Explicit host tracking (an IGMPv3 snooping feature) is not supported.
- IGMPv1 is not supported.
- ISSU is not supported on Layer 2 Multicast.
- IGMPv3-exclude is not supported in EVPN multi-homing or proxy scenarios.
- PIM control packets are supported when snooping is enabled.

Information About IGMP Snooping

IGMP Snooping Overview

Description of Basic Functions

IGMP snooping provides a way to constrain multicast traffic at Layer 2. By snooping the IGMP membership reports sent by hosts in the bridge domain, the IGMP snooping application can set up Layer 2 multicast forwarding tables to deliver traffic only to ports with at least one interested member, significantly reducing the volume of multicast traffic.

Configured at Layer 3, IGMP provides a means for hosts in an IPv4 multicast network to indicate which multicast traffic they are interested in and for routers to control and limit the flow of multicast traffic in the network at Layer 3.

IGMP snooping uses the information in IGMP membership report messages to build corresponding information in the forwarding tables to restrict IP multicast traffic at Layer 2. The forwarding table entries are in the form <Route, OIF List>, where:

- Route is a <*, G> route or <S, G> route, where * is any source, G is group and S is the source.
- OIF List comprises all bridge ports that have sent IGMP membership reports for the specified route plus all multicast router (mrouter) ports in the bridge domain.

Implemented in a multicast network, IGMP snooping has the following attributes:

- In its basic form, it reduces bandwidth consumption by reducing multicast traffic that would otherwise flood an entire VPLS bridge domain.
- With the use of some optional configurations, it provides security between bridge domains by filtering the IGMP reports received from hosts on one bridge port and preventing leakage towards the hosts on other bridge ports.
- Using optional configurations, reduces the traffic impact on upstream IP multicast routers by suppressing IGMP membership reports (IGMPv2) or by acting as an IGMP proxy reporter (IGMPv3) to the upstream IP multicast router.

High Availability Features

All high availability features apply to the IGMP snooping processes with no additional configuration beyond enabling IGMP snooping. The following high availability features are supported:

- Process restarts

- RP Failover
- Non-Stop Forwarding (NSF)—Forwarding continues unaffected while the control plane is restored following a process restart or route processor (RP) failover.
- Line card online insertion and removal (OIR)

Bridge Domain Support

IGMP snooping operates at the bridge domain level. When IGMP snooping is enabled on a bridge domain, the snooping functionality applies to all ports under the bridge domain, including:

- Physical ports under the bridge domain.
- Ethernet flow points (EFPs)—An EFP can be a VLAN, VLAN range, list of VLANs, or an entire interface port.
- Ethernet bundles—Ethernet bundles include IEEE 802.3ad link bundles and Cisco EtherChannel bundles. From the perspective of the IGMP snooping application, an Ethernet bundle is just another EFP. The forwarding application in the router randomly nominates a single port from the bundle to carry the multicast traffic.

Note The **efp-visibility** configuration is required when a bridge has attachment circuits as VLAN sub-interfaces from the same bundle-ether or physical interface.

IGMP snooping for bridge domains without Bridged Virtual Interface (BVI) is supported with the following design consideration:

You must configure **multicast-source ipv4** under L2VPN if snooping is enabled and mutlicast traffic source is located behind the AC port.

Configuration Example:

```
Router(config)# l2vpn
Router(config-l2vpn)# bridge group 1
Router(config-l2vpn-bg)#bridge-domain 1
Router(config-l2vpn-bg-bd)#multicast-source ipv4
Router(config-l2vpn-bg-bd)#efp-visibility
Router(config-l2vpn-bg-bd)#igmp snooping profile igmpsn
Router(config-l2vpn-bg-bd)#exit
Router(config-l2vpn-bg-bd)#interface TenGigE0/0/0/3.31
Router(config-l2vpn-bg-bd-ac)#exit
Router(config-l2vpn-bg-bd)#interface TenGigE0/0/0/3.32
Router(config-l2vpn-bg-bd-ac)#exit
Router(config-l2vpn-bg-bd)#routed interface BVI1
Router(config-l2vpn-bg-bd-bvi)#exit
```

Multicast Router Port

A Multicast router (Mrouter) port is a port that connects to a Multicast router. The device includes the Multicast router port(s) numbers when it forwards the Multicast streams and IGMP registration messages. This is required so that the Multicast routers can, in turn, forward the Multicast streams and propagate the registration messages to other subnets. The reports would be re-injected over mrouter ports.

Multicast Router and Host Ports

IGMP snooping classifies each port (for example, EFPs, PWs, physical ports, or EFP bundles) as one of the following:

- Multicast router ports (mrouter ports)—These are ports to which a multicast-enabled router is connected. Mrouter ports are usually dynamically discovered, but may also be statically configured. Multicast traffic is always forwarded to all mrouter ports, except when an mrouter port is the ingress port.
- Host ports—Any port that is not an mrouter port is a host port.

IGMP snooping classifies each port (for example, EFPs, physical ports, or EFP bundles) as a host ports, that is, any port that is not an mrouter port is a host port.

Multicast Traffic Handling within a Bridge Domain with IGMP Snooping Enabled

The following tables describe traffic handling behaviors by IGMP snooping mrouter and host ports.

By default, IGMP snooping supports IGMPv2 and IGMPv3. The version of the IGMP querier discovered in the bridge domain determines the operational version of the snooping processes. If you change the default, configuring IGMP snooping to support a minimum version of IGMPv3, IGMP snooping ignores any IGMPv2 queriers.

Table 4: Multicast Traffic Handling for an IGMPv2 Querier

Traffic Type	Received on MRouter Ports	Received on Host Ports
IP multicast source traffic	Forwards to all mrouter ports and to host ports that indicate interest.	Forwards to all mrouter ports and to host ports that indicate interest.
IGMP general queries	Forwards to all ports.	—
IGMP group-specific queries	Forwards to all other mrouter ports.	—
IGMPv2 joins	Examines (snoops) the reports. • If report suppression is enabled, forwards first join for a new group or first join following a general query for an existing group. • If report suppression is disabled, forwards on all mrouter ports.	Examines (snoops) the reports. • If report suppression is enabled, forwards first join for a new group or first join following a general query for an existing group. • If report suppression is disabled, forwards on all mrouter ports.
IGMPv3 reports	Ignores	Ignores
IGMPv2 leaves	Invokes last member query processing.	Invokes last member query processing.

Table 5: Multicast Traffic Handling for an IGMPv3 Querier

Traffic Type	Received on MRouter Ports	Received on Host Ports
IP multicast source traffic	Forwards to all mrouter ports and to host ports that indicate interest.	Forwards to all mrouter ports and to host ports that indicate interest.

Traffic Type	Received on MRouter Ports	Received on Host Ports
IGMP general queries	Forwards to all ports.	—
IGMP group-specific queries	If received on the querier port floods on all ports. Forwards to all other Mrouter ports.	—
IGMPv2 joins	Handles as IGMPv3 IS_EX{} reports.	Handles as IGMPv3 IS_EX{} reports.
IGMPv3 reports	• If proxy reporting is enabled—For state changes or source-list changes, generates a state change report on all mrouter ports. • If proxy reporting is disabled—Forwards on all mrouter ports.	• If proxy reporting is enabled—For state changes or source-list changes, generates a state change report on all mrouter ports. • If proxy reporting is disabled—Forwards on all mrouter ports.
IGMPv2 leaves	Handles as IGMPv3 IS_IN{} reports.	Handles as IGMPv3 IS_IN{} reports.

IGMP Snooping Configuration Profiles

To enable IGMP snooping on a bridge domain, you must attach a profile to the bridge domain. The minimum configuration is an empty profile if BVI is configured. An empty profile enables the default configuration options and settings for IGMP snooping, as listed in the *Default IGMP Snooping Configuration Settings*.

Note You must configure the **system-ip-address** and **internal-querier** when the BVI is not configured, and no other queriers are present in the same domain.

Configuration Example:

```
Router(config)#igmp snooping profile igmpsn
Router(config-igmp-snooping-profile)#system-ip-address 192.0.2.1
Router(config-igmp-snooping-profile)#internal-querier
```

You can attach IGMP snooping profiles to bridge domains or to ports under a bridge domain. The following guidelines explain the relationships between profiles attached to ports and bridge domains:

- Any IGMP Snooping profile attached to a bridge domain, even an empty profile, enables IGMP snooping. To disable IGMP snooping, detach the profile from the bridge domain.
- An empty profile configures IGMP snooping on the bridge domain and all ports under the bridge using default configuration settings.
- A bridge domain can have only one IGMP snooping profile attached to it (at the bridge domain level) at any time.
- Port profiles are not in effect if the bridge domain does not have a profile attached to it.
- IGMP snooping must be enabled on the bridge domain for any port-specific configurations to be in effect.

- If a profile attached to a bridge domain contains port-specific configuration options, the values apply to all of the ports under the bridge, including all mrouter and host ports, unless another port-specific profile is attached to a port.
- When a profile is attached to a port, IGMP snooping reconfigures that port, disregarding any port configurations that may exist in the bridge-level profile.

Creating Profiles

To create a profile, use the **igmp snooping profile** command in global configuration mode.

Attaching and Detaching Profiles

To attach a profile to a bridge domain, use the **igmp snooping profile** command in l2vpn bridge group bridge domain configuration mode. To attach a profile to a port, use the **igmp snooping profile** command in the interface configuration mode under the bridge domain. To detach a profile, use the **no** form of the command in the appropriate configuration mode.

When you detach a profile from a bridge domain or a port, the profile still exists and is available for use at a later time. Detaching a profile has the following results:

- If you detach a profile from a bridge domain, IGMP snooping is deactivated in the bridge domain.
- If you detach a profile from a port, IGMP snooping configuration values for the port are instantiated from the bridge domain profile.

Changing Profiles

You cannot make changes to an active profile. An active profile is one that is currently attached.

- If the active profile is configured under the bridge, you must detach it from the bridge, and reattach it.
- If the active profile is configured under a specific bridge port, you must detach it from the bridge port, and reattach it.

Another way to do this is to create a new profile incorporating the desired changes and attach it to the bridges or ports, replacing the existing profile. This deactivates IGMP snooping and then reactivates it with parameters from the new profile.

Default IGMP Snooping Configuration Settings

Table 6: IGMP Snooping Default Configuration Values

Scope	Feature	Default Value
Bridge Domain	IGMP snooping	Disabled on a bridge domain until an enabling IGMP snooping profile is attached to the bridge domain.
	internal querier	By default Internal Querier is disabled. To enable Internal Querier, add it to the IGMP snooping profile. Internal Querier is not recommended, when BVI and IGMP snooping is configured under a bridge.
	last-member-query-count	2
	last-member-query-interval	1000 (milliseconds)
	minimum-version	2 (supporting IGMPv2 and IGMPv3)
	querier query-interval	60 (seconds) **Note** This is a nonstandard default value.
	report-suppression	Enabled (enables report suppression for IGMPv2 and proxy-reporting for IGMPv3)
	querier robustness-variable	2
	router alert check	Enabled
	tcn query solicit	Disabled
	tcn flood	Enabled
	ttl-check	Enabled
	unsolicited-report-timer	1000 (milliseconds)
Port	immediate-leave	Disabled
	mrouter	No static mrouters configured; dynamic discovery occurs by default.
	static group	None configured

IGMP Snooping Configuration at the Bridge Domain Level

IGMP Minimum Version

The **minimum-version** command determines which IGMP versions are supported by IGMP snooping in the bridge domain:

- When minimum-version is 2, IGMP snooping intercepts IGMPv2 and IGMPv3 messages. This is the default value.
- When minimum-version is 3, IGMP snooping intercepts only IGMPv3 messages and drops all IGMPv2 messages.

IGMPv1 is not supported. The scope for this command is the bridge domain. The command is ignored in a profile attached to a port.

Group Membership Interval, Robustness Variable, and Query Interval

The group membership interval (GMI) controls when IGMP snooping expires stale group membership states. The **show igmp snooping group** command shows groups with an expiry time of 0 until that stale state is cleaned up following the next query interval.

The GMI is calculated as:

GMI = (robustness-variable * query-interval) + maximum-response-time

where:

- maximum-response-time (MRT) is the amount of time during which receivers are required to report their membership state.
- robustness-variable is an integer used to influence the calculated GMI.
- query-interval is the amount of time between general queries.

Values for the components in the GMI are obtained as follows:

- MRT is advertised in the general query, for both IGMPv2 and IGMPv3.
- If the querier is running IGMPv2, IGMP snooping uses the IGMP-snooping-configured values for the robustness-variable and query-interval. These parameter values must match the configured values for the querier. In most cases, if you are interacting with other Cisco routers, you should not need to explicitly configure these values—the default values for IGMP snooping should match the default values of the querier. If they do not, use the **querier robustness-variable** and **querier query-interval** commands to configure matching values.
- IGMPv3 general queries convey values for robustness-variable and query-interval (QRV and QQI, respectively). IGMP snooping uses the values from the query, making the IGMP snooping GMI exactly match that of the querier.

How to Configure IGMP Snooping

The first two tasks are required to configure basic IGMP snooping configuration.

Creating an IGMP Snooping Profile

Procedure

Step 1 **configure**

Step 2 **igmp snooping profile** *profile-name*

Example:

```
RP/0/RP0/CPU0:router(config)# igmp snooping profile default-bd-profile
```

Enters IGMP snooping profile configuration mode and creates a named profile.

The default profile enables IGMP snooping. You can commit the new profile without any additional configurations, or you can include additional configuration options to the profile. You can also return to the profile later to add configurations, as described in other tasks in this module.

Step 3 Optionally, add commands to override default configuration values.

If you are creating a bridge domain profile, consider the following:

- An empty profile is appropriate for attaching to a bridge domain. An empty profile enables IGMP snooping with default configuration values.
- You can optionally add more commands to the profile to override default configuration values.
- If you include port-specific configurations in a bridge domain profile, the configurations apply to all ports under the bridge, unless another profile is attached to a port.

If you are creating a port-specific profile, consider the following:

- While an empty profile could be attached to a port, it would have no effect on the port configuration.
- When you attach a profile to a port, IGMP snooping reconfigures that port, overriding any inheritance of configuration values from the bridge-domain profile. You must repeat the commands in the port profile if you want to retain those configurations.

You can detach a profile, change it, and reattach it to add commands to a profile at a later time.

Step 4 **commit**

Where to Go Next

Attach the profile to bridge domains or ports to complete immediate-leave configuration. See one of the following sections:

Attaching a Profile and Activating IGMP Snooping on a Bridge Domain

To activate IGMP snooping on a bridge domain, attach an IGMP snooping profile to the bridge domain, as described in the following steps.

Procedure

Step 1 **configure**

Step 2 **l2vpn**

Example:

```
RP/0/RP0/CPU0:router(config)# l2vpn
```

Enters Layer 2 VPN configuration mode.

Step 3 **bridge group** *bridge-group-name*

Example:

```
RP/0/RP0/CPU0:router(config-l2vpn)# bridge group GRP1
```

Enters Layer 2 VPN VPLS bridge group configuration mode for the named bridge group.

Step 4 **bridge-domain** *bridge-domain-name*

Example:

```
RP/0/RP0/CPU0:router(config-l2vpn-bg)# bridge-domain ISP1
```

Enters Layer 2 VPN VPLS bridge group bridge domain configuration mode for the named bridge domain.

Step 5 **igmp snooping profile** *profile-name*

Example:

```
RP/0/RP0/CPU0:router(config-l2vpn-bg-bd)# igmp snooping profile default-bd-profile
```

Attaches the named IGMP snooping profile to the bridge domain, enabling IGMP snooping on the bridge domain.

Step 6 **commit**

Step 7 **show igmp snooping bridge-domain detail**

Example:

```
RP/0/RP0/CPU0:router# show igmp snooping bridge-domain detail
```

(Optional) Verifies that IGMP snooping is enabled on a bridge domain and shows the IGMP snooping profile names attached to bridge domains and ports.

Step 8 **show l2vpn bridge-domain detail**

Example:

```
RP/0/RP0/CPU0:router# show l2vpn bridge-domain
```

(Optional) Verifies that IGMP snooping is implemented in the forwarding plane (Layer 2) on a bridge domain.

Detaching a Profile and Deactivating IGMP Snooping on a Bridge Domain

To deactivate IGMP snooping on a bridge domain, remove the profile from the bridge domain using the following steps.

Note A bridge domain can have only one profile attached to it at a time.

Procedure

Step 1 **configure**

Step 2 **l2vpn**

Example:

```
RP/0/RP0/CPU0:router(config)# l2vpn
```

Enters Layer 2 VPN configuration mode.

Step 3 **bridge group** *bridge-group-name*

Example:

```
RP/0/RP0/CPU0:router(config-l2vpn)# bridge group GRP1
```

Enters Layer 2 VPN VPLS bridge group configuration mode for the named bridge group.

Step 4 **bridge-domain** *bridge-domain-name*

Example:

```
RP/0/RP0/CPU0:router(config-l2vpn-bg)# bridge-domain ISP1
```

Enters Layer 2 VPN VPLS bridge group bridge domain configuration mode for the named bridge domain.

Step 5 **no igmp snooping disable**

Example:

```
RP/0/RP0/CPU0:router(config-l2vpn-bg-bd)# no igmp snooping disable
```

Detaches the IGMP snooping profile from the bridge domain, disabling IGMP snooping on that bridge domain.

Note
Only one profile can be attached to a bridge domain at a time. If a profile is attached, IGMP snooping is enabled. If a profile is not attached, IGMP snooping is disabled.

Step 6 **commit**

Step 7 **show igmp snooping bridge-domain detail**

Example:

```
RP/0/RP0/CPU0:router# show igmp snooping bridge-domain detail
```

(Optional) Verifies that IGMP snooping is disabled on a bridge domain.

Step 8 **show l2vpn bridge-domain detail**

Example:

```
RP/0/RP0/CPU0:router# show l2vpn bridge-domain
```

(Optional) Verifies that IGMP snooping is disabled in the forwarding plane (Layer 2) on a bridge domain.

Attaching and Detaching Profiles to Ports Under a Bridge

Before you begin

IGMP snooping must be enabled on the bridge domain for port-specific profiles to affect IGMP snooping behavior.

Procedure

Step 1 **configure**

Step 2 **l2vpn**

Example:

```
RP/0/RP0/CPU0:router(config)# l2vpn
```

Enters Layer 2 VPN configuration mode.

Step 3 **bridge group** *bridge-group-name*

Example:

```
RP/0/RP0/CPU0:router(config-l2vpn)# bridge group GRP1
```

Enters Layer 2 VPN bridge group configuration mode for the named bridge group.

Step 4 **bridge-domain** *bridge-domain-name*

Example:

```
RP/0/RP0/CPU0:router(config-l2vpn-bg)# bridge-domain ISP1
```

Enters Layer 2 VPN bridge group bridge domain configuration mode for the named bridge domain.

Step 5 **interface** *interface-type interface-number*

Example:

```
RP/0/RP0/CPU0:router(config-l2vpn-bg-bd)# interface gig 1/1/1/1
```

Enters Layer 2 VPN VPLS bridge group bridge domain interface configuration mode for the named interface or PW.

Step 6 Do one of the following:

- **igmp snooping profile** *profile-name*
- **no igmp snooping**

Example:

```
RP/0/RP0/CPU0:router(config-l2vpn-bg-bd-if)# igmp snooping profile mrouter-port-profile
```

Attaches the named IGMP snooping profile to the port.

Note
A profile on a port has no effect unless there is also a profile attached to the bridge.

The **no** form of the command detaches a profile from the port. Only one profile can be attached to a port.

Step 7 **commit**

Step 8 **show igmp snooping bridge-domain detail**

Example:

```
RP/0/RP0/CPU0:router# show igmp snooping bridge-domain detail
```

(Optional) Verifies that IGMP snooping is enabled on a bridge domain and shows the IGMP snooping profile names attached to bridge domains and ports.

Step 9 **show l2vpn bridge-domain detail**

Example:

```
RP/0/RP0/CPU0:router# show l2vpn bridge-domain
```

(Optional) Verifies that IGMP snooping is implemented in the forwarding plane (Layer 2) on a bridge domain.

Verifying Multicast Forwarding

Procedure

Step 1 **configure**

Step 2 **show l2vpn forwarding bridge-domain** [*bridge-group-name*:*bridge-domain-name*] **mroute ipv4** [**group** *group_IPaddress*] [**hardware** {**ingress** | **egress**}] [**detail**]**location** *node-id*

Example:

```
 RP/0/RP0/CPU0:router#show l2vpn forwarding bridge-domain bg1:bd1 mroute ipv4 group 234.192.4.1
hardware ingress detail location 0/1/cPU0
```

Displays multicast routes as they are converted into the forwarding plane forwarding tables. Use optional arguments to limit the display to specific bridge groups or bridge domains.

If these routes are not as expected, check the control plane configuration and correct the corresponding IGMP snooping profiles.

Step 3 **show l2vpn forwarding bridge-domain** [*bridge-group-name*:*bridge-domain-name*] **mroute ipv4 summary location** *node-id*

Example:

```
RP/0/RP0/CPU0:router# show l2vpn forwarding bridge-domain bg1:bd1 mroute ipv4 summary location 0/3/CPU0
```

Displays summary-level information about multicast routes as stored in the forwarding plane forwarding tables. Use optional arguments to limit the display to specific bridge domains.

Configuration Examples for IGMP Snooping

The following examples show how to enable IGMP snooping on Layer 2 VPLS bridge domains on Cisco 8000 Series Routers:

Configuring IGMP Snooping on Physical Interfaces Under a Bridge: Example

1. Create two profiles.

```
igmp snooping profile bridge_profile
!
igmp snooping profile port_profile
   mrouter
!
```

2. Configure two physical interfaces for L2 transport.

```
interface GigabitEthernet0/8/0/38
   negotiation auto
   l2transport
   no shut
   !
!
interface GigabitEthernet0/8/0/39
   negotiation auto
   l2transport
   no shut
```

```
   !
!
```

3. Add interfaces to the bridge domain. Attach bridge_profile to the bridge domain and port_profile to one of the Ethernet interfaces. The second Ethernet interface inherits IGMP snooping configuration attributes from the bridge domain profile.

```
l2vpn
  bridge group bg1
     bridge-domain bd1
     igmp snooping profile bridge_profile
     interface GigabitEthernet0/8/0/38
       igmp snooping profile port_profile
     interface GigabitEthernet0/8/0/39

!
  !
!
```

4. Verify the configured bridge ports.

```
show igmp snooping port
```

Configuring IGMP Snooping on VLAN Interfaces Under a Bridge: Example

1. Configure two profiles.

```
multicast-source ipv4
igmp snooping profile bridge_profile

igmp snooping profile port_profile
   mrouter
!
```

2. Configure VLAN interfaces for L2 transport.

```
interface GigabitEthernet0/8/0/8
   negotiation auto
   no shut
   !
!
interface GigabitEthernet0/8/0/8.1 l2transport
   encapsulation dot1q 1001
   rewrite ingress tag pop 1 symmetric
   !
!
interface GigabitEthernet0/8/0/8.2 l2transport
   encapsulation dot1q 1002
   rewrite ingress tag pop 1 symmetric
   !
!
```

3. Attach a profile and add interfaces to the bridge domain. Attach a profile to one of the interfaces. The other interface inherits IGMP snooping configuration attributes from the bridge domain profile.

```
l2vpn
  bridge group bg1
     bridge-domain bd1
     multicast-source ipv4
     igmp snooping profile bridge_profile
     interface GigabitEthernet0/8/0/8.1
       igmp snooping profile port_profile
     interface GigabitEthernet0/8/0/8.2

     !
  !
!
```

4. Verify the configured bridge ports.

```
show igmp snooping port
```

Configuring IGMP Snooping on Ethernet Bundles Under a Bridge: Example

1. This example assumes that the front-ends of the bundles are preconfigured. For example, a bundle configuration might consist of three switch interfaces, as follows:

```
    interface Port-channel1
    !
interface GigabitEthernet0/0/0/0
    !
interface GigabitEthernet0/0/0/1
!
    interface GigabitEthernet0/0/0/2
       channel-group 1 mode on
    !
    interface GigabitEthernet0/0/0/3
       channel-group 1 mode on
    !
```

2. Configure two IGMP snooping profiles.

```
multicast-source ipv4
       igmp snooping profile bridge_profile
       !
       multicast-source ipv4
       igmp snooping profile port_profile
          mrouter
       !
```

3. Configure interfaces as bundle member links.

```
       interface GigabitEthernet0/0/0/0
         bundle id 1 mode on
         negotiation auto
       !
       interface GigabitEthernet0/0/0/1
         bundle id 1 mode on
         negotiation auto
       !
```

```
interface GigabitEthernet0/0/0/2
  bundle id 2 mode on
  negotiation auto
!
interface GigabitEthernet0/0/0/3
  bundle id 2 mode on
  negotiation auto
!
```

4. Configure the bundle interfaces for L2 transport.

```
interface Bundle-Ether 1
          l2transport
          !
       !
       interface Bundle-Ether 2
          l2transport
          !
       !
```

5. Add the interfaces to the bridge domain and attach IGMP snooping profiles.

```
       l2vpn
         bridge group bg1
            bridge-domain bd1
            igmp snooping profile bridge_profile
            interface bundle-Ether 1
              igmp snooping profile port_profile
            interface bundle-Ether 2

            !
         !
      !
```

6. Verify the configured bridge ports.

```
show igmp snooping port
```

Configuring Multicast over Integrated Routing Bridging Active/Active Multihome

Configurations performed on peer 1:

1. Layer 2 Base Configuration

```
hostname peer1
!
interface Bundle-Ether2
!
interface Bundle-Ether2.2 l2transport
 encapsulation dot1q 2
 rewrite ingress tag pop 1 symmetric
!
interface TenGigE0/0/0/0
```

```
 bundle id 2 mode on
 no shut
!
```

2. IGMPv2 Snoop Configurations

```
hostname peer1
!
router igmp

  version 2
 !
!
l2vpn
 bridge group VLAN2
  bridge-domain VLAN2
   multicast-source ipv4
   igmp snooping profile 1
   interface Bundle-Ether2.2
   !

   evi 2
   !
  !
 !
multicast-source ipv4
igmp snooping profile 1
!
```

Configurations Performed on Peer 2:

1. Layer 2 Base Configuration

```
hostname peer2
!
interface Bundle-Ether2
!
interface Bundle-Ether2.2 l2transport
 encapsulation dot1q 2
 rewrite ingress tag pop 1 symmetric
!
interface TenGigE0/0/0/0
 bundle id 2 mode on
 no shut
!
```

2. IGMPv2 Snoop Configurations

```
hostname peer2
!
router igmp

  version 2
 !
!
l2vpn
 bridge group VLAN2
  bridge-domain VLAN2
   multicast-source ipv4
   igmp snooping profile 1
   interface Bundle-Ether2.2
   !

   evi 2
   !
  !
```

```
 !
multicast-source ipv4
igmp snooping profile 1
!
```

Verifying IGMP Snooping

In this example, the receiver sends an IGMPv2 join for the group 239.0.0.2. On Peer2, this group has a D Flag, that means the actual IGMP joined peer2, but not peer1. On Peer1, this group has a B flag, that means this group is learnt from BGP.

```
RP/0/RP0/CPU0:peer1#show igmp snooping group
Fri Aug 31 22:27:46.363 UTC

Key: GM=Group Filter Mode, PM=Port Filter Mode
Flags Key: S=Static, D=Dynamic, B=BGP Learnt, E=Explicit Tracking, R=Replicated

                           Bridge Domain VLAN10:VLAN10

Group           Ver GM Source          PM Port                       Exp   Flgs
-----           --- -- ------          -- ----                       ---   ----
239.0.0.2       V2  -  *               -  BE2.2                      never B


RP/0/RP0/CPU0:peer2#show igmp snooping group
Fri Aug 31 22:27:49.686 UTC

Key: GM=Group Filter Mode, PM=Port Filter Mode
Flags Key: S=Static, D=Dynamic, B=BGP Learnt, E=Explicit Tracking, R=Replicated

                           Bridge Domain VLAN10:VLAN10

Group           Ver GM Source          PM Port                       Exp   Flgs
-----           --- -- ------          -- ----                       ---   ----
239.0.0.2       V2  -  *               -  BE2.2                      74    D
```

Verifying Dual DR PIM Uplink

In this example, when the source 126.0.0.100 sends traffic to group 239.0.0.2, you see both Peer1 and Peer2 are sending PIM join upstream. The incoming interface for (*,G) and (S,G) should be the interface toward the RP and source respectively. For both Peer1 and Peer2, the outgoing interface should be the BVI interface facing the receiver.

```
RP/0/RP0/CPU0:peer1#show mrib route
:
:

(*,239.0.0.2) RPF nbr: 30.0.0.4 Flags: C RPF
  Up: 00:13:41
  Incoming Interface List
    HundredGigE0/0/0/1 Flags: A NS, Up: 00:13:41
  Outgoing Interface List
    BVI2 Flags: F NS LI, Up: 00:13:41

(126.0.0.100,239.0.0.2) RPF nbr: 30.0.0.4 Flags: RPF
  Up: 00:03:34
  Incoming Interface List
    HundredGigE0/0/0/1 Flags: A, Up: 00:03:34
```

```
  Outgoing Interface List
    BVI2 Flags: F NS, Up: 00:03:34
:
:
RP/0/RP0/CPU0:peer2#show mrib route
:
:

(*,239.0.0.2) RPF nbr: 50.0.0.4 Flags: C RPF
  Up: 00:13:33
  Incoming Interface List
    HundredGigE0/0/0/2 Flags: A NS, Up: 00:13:33
  Outgoing Interface List
    BVI2 Flags: F NS LI, Up: 00:13:33

(126.0.0.100,239.0.0.2) RPF nbr: 50.0.0.4 Flags: RPF
  Up: 00:03:24
  Incoming Interface List
    HundredGigE0/0/0/2 Flags: A, Up: 00:03:24
  Outgoing Interface List
    BVI2 Flags: F NS, Up: 00:03:24
:
:
```

MLD Snooping

Multicast Listener Discovery (MLD) snooping is a technique that uses the MLD protocol to optimize the delivery of multicast traffic.

Table 7: Feature History Table

Feature Name	Release Information	Feature Description
MLD snooping	Release 25.2.1	Introduced in this release on: Centralized Systems (8400 [ASIC: K100]) (select variants only*) *This feature is now supported on Cisco 8404 routers.

Feature Name	Release Information	Feature Description
MLD snooping	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100]), (8700 [ASIC: K100])(select variants only*). Multicast Listener Discovery (MLD) snooping is a technique that uses the MLD protocol to optimize the delivery of multicast traffic. When you enable MLD snooping on the router, it sends multicast data only to network segments with devices that have expressed interest in receiving it. By sending multicast data only to interested devices, the router minimizes unnecessary traffic and conserves bandwidth on the network. *This feature is now supported on: • 8212-48FH-M • 8712-MOD-M • 8711-32FH-M

Multicast Listener Discovery (MLD) snooping provides a way to constrain multicast traffic at Layer 2. By snooping the MLD membership reports sent by hosts in the bridge domain, the MLD snooping application can set up Layer 2 multicast forwarding tables to deliver traffic only to ports with at least one interested member, significantly reducing the volume of multicast traffic.

MLD snooping uses the information in MLD membership report messages to build corresponding information in the forwarding tables to restrict IPv6 multicast traffic at Layer 2. The forwarding table entries are in the form <Route, OIF List>, where:

- Route is a <*, G> route or <S, G> route.
- OIF List comprises all bridge ports that have sent MLD membership reports for the specified route plus all multicast router (mrouter) ports in the bridge domain.

For more information regarding MLD snooping, refer the *Multicast Configuration Guide for Cisco 8000 Series Routers*.

Prerequisites for MLD Snooping

- The network must be configured with a layer2 VPN.
- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Supported Features and Restrictions for MLD Snooping

- Receiver behind L2 ACs in the same L2 bridge domain is supported.
- Source behind L2 ACs in the same L2 bridge domain is supported.
- EVPN MLD sync is not supported.
- VPLS is not supported.
- The **router-alert-check disable** configuration command is not supported.
- EVPN configuration must have the **control-word-disable** configuration.
- PIM control packets (join and hello) processing is not supported when snooping is enabled, so a multicast router selection based on PIM packets won't occur.
- Explicit host tracking.
- Multicast Admission Control.
- Security filtering.
- Report rate limiting.
- Multicast router discovery.
- In an EVPN dual-home AA scenario:
  - If the multicast source and receiver are in the same bridge domain (BD), the receiver might receive permanent traffic duplication.
  - In an EVPN dual-home receiver AA scenario, transient traffic duplication is expected when the DH node role changes from DF to nDF and vice versa.
  - Source=ESI1=BE-X.A, Receiver=ESI1=BE-X.B under the same BD is not supported (where X.A and X.B represent two AC ports for the bundle interface BE).

Advantages of MLD Snooping

- In its basic form, it reduces bandwidth consumption by reducing multicast traffic that would otherwise flood an entire VPLS bridge domain.
- With the use of some optional configurations, it provides security between bridge domains by filtering the MLD reports received from hosts on one bridge port and preventing leakage towards the hosts on other bridge ports.

High Availability (HA) features for MLD

MLD supports the following HA features:

- Process restarts
- Stateful Switch-Over (SSO)

Bridge Domain Support for MLD

MLD snooping operates at the bridge domain level. When MLD snooping is enabled on a bridge domain, the snooping functionality applies to all ports under the bridge domain, including:

- Physical ports under the bridge domain.
- Ethernet flow points (EFPs)—An EFP can be a VLAN, VLAN range, list of VLANs, or an entire interface port.
- Ethernet bundles—Ethernet bundles include IEEE 802.3ad link bundles and Cisco EtherChannel bundles. From the perspective of the MLD snooping application, an Ethernet bundle is just another EFP. The forwarding application in the Cisco 8000 Series Routers randomly nominates a single port from the bundle to carry the multicast traffic.

Note The **efp-visibility** configuration is required when a bridge has attachment circuits as VLAN sub-interfaces from the same bundle-ether or physical interface.

Multicast Router and Host Ports

MLD snooping classifies each port as one of the following:

- Multicast router ports (mrouter ports)—These are ports to which a multicast-enabled router is connected. Mrouter ports are usually dynamically discovered, but may also be statically configured. Multicast traffic is always forwarded to all mrouter ports, except when an mrouter port is the ingress port.
- Host ports—Any port that is not an mrouter port is a host port.

Multicast Router Discovery for MLD

MLD snooping discovers mrouter ports dynamically. You can also explicitly configure a port as an emrouter port.

- Discovery- MLD snooping identifies upstream mrouter ports in the bridge domain by snooping mld query messages and Protocol Independent Multicast Version 2 (PIMv2) hello messages. Snooping PIMv2 hello messages identifies mld nonqueriers in the bridge domain.
- Static configuration—You can statically configure a port as an mrouter port with the **mrouter** command in a profile attached to the port. Static configuration can help in situations when incompatibilities with non-Cisco equipment prevent dynamic discovery.

Multicast Traffic Handling for MLD

The following tables describe the traffic handling behavior by MLD mrouters and host ports.

Table 8: Multicast Traffic Handling for a MLDv1 Querier

Traffic Type	Received on MRouter Ports	Received on Host Ports
IP multicast source traffic	Forwards to all mrouter ports and to host ports that indicate interest.	Forwards to all mrouter ports and to host ports that indicate interest.
MLD general queries	Forwards to all ports.	—
MLD group-specific queries	Forwards to all other mrouter ports.	Dropped
MLDv1 joins	Examines (snoops) the reports. • If report suppression is enabled, forwards first join for a new group or first join following a general query for an existing group. • If report suppression is disabled, forwards on all mrouter ports.	Examines (snoops) the reports. • If report suppression is enabled, forwards first join for a new group or first join following a general query for an existing group. • If report suppression is disabled, forwards on all mrouter ports.
MLDv2 reports	Ignores	Ignores
MLDv1 leaves	Invokes last member query processing.	Invokes last member query processing.

Table 9: Multicast Traffic Handling for a MLDv2 Querier

Traffic Type	Received on MRouter Ports	Received on Host Ports
IP multicast source traffic	Forwards to all mrouter ports and to host ports that indicate interest.	Forwards to all mrouter ports and to host ports that indicate interest.
MLD general queries	Forwards to all ports.	—
MLD group-specific queries	If received on the querier port floods on all ports.	—
MLDv1 joins	Handles as MLDv2 IS_EX{} reports.	Handles as MLDv2 IS_EX{} reports.
MLDv2 reports	• If proxy reporting is enabled—For state changes or source-list changes, generates a state change report on all mrouter ports. • If proxy reporting is disabled—Forwards on all mrouter ports.	• If proxy reporting is enabled—For state changes or source-list changes, generates a state change report on all mrouter ports. • If proxy reporting is disabled—Forwards on all mrouter ports.
MLDv1 leaves	Handles as MLDv2 IS_IN{} reports.	Handles as MLDv2 IS_IN{} reports.

Multicast Listener Discovery over BVI

Multicast IPv6 packets received from core, which has BVI as forwarding interface, is forwarded to access over snooped L2 AC or interface.

Note

- As per MLDv2 RFC recommendation the MLDv2 reports should carry the Hop-by-Hop options header for the reports to get punted up.
- MLDv2 is supported over BVI only when BVI is configured as a forwarding interface.

MLD and BVI Overview

Routers use the Internet Group Management Protocol (IGMP) (IPv4) and Multicast Listener Discovery (MLD) (IPv6) to learn whether members of a group are present on their directly attached subnets. Hosts join multicast groups by sending IGMP or MLD report messages.

MLDv1 and MLDv2 are supported on Cisco 8010 Series Routers (*select varaints). However, MLDv2 is enabled when you configure MLD by default.

MLDv2 shares feature parity with IGMPv3 with respect to all supported interface types with the exception of PPoE and subinterfaces. MLDv2 enables a node to report interest in listening to packets only from specific multicast source addresses.

A BVI interface is a routed interface representing a set of interfaces (bridged) in the same L2 broadcast domain. MLD join messages coming in or out of this broadcast domain passes through the BVI interface.

Multicast Traffic Over Layer 2 IPv6 Network

The Multicast Traffic over Layer 2 IPv6 Network (L2MC IPv6) is an optimized forwarding technique, and it helps in saving the bandwidth. By default, the bridge floods IPv6 multicast packets to all AC, whereas the L2MC IPv6 feature allows you to forward the IPv6 multicast packets only to the interested MLD-snooped AC.

When IPv6 multicast packets are received over Layer 2 AC and interfaces, the lookup gets done for Virtual Switch Interfaces (VSI), Groups (G), and Services (S) or for VSI and G. The VSI details show the VLAN or VXLAN segment to which the packet belongs, while the G and S identify the multicast groups and services to which the packet should be forwarded. Based on this lookup, the traffic is forwarded to the interested receivers connected to the Layer 2 AC.

The MLD control packets received over Layer 2 AC are snooped and punted to create the route entries. This route entries are needed to avail the following supports:

- Layer 2 Multicast IPv6 support.
- EVPN sync support for IPv4 routes.

Limitations and Restrictions

- This feature doesn’t support MLD sync.
- With L2MC IPv6 support, the existing L2MC IPv4 scale reduces proportionally.

Configuration Example

The L2MC IPv6 feature is not enabled by default. Following is a configuration example that shows how to enable the feature.

```
router(config)# l2vpn
 router(config-l2vpn)# bridge group 1
 router(config-l2vpn-bg)#bridge-domain 1
 router(config-l2vpn-bg-bd)#multicast-source ipv6
 router(config-l2vpn-bg-bd)#efp-visibility
 router(config-l2vpn-bg-bd)#mld snooping profile prof1
 router(config-l2vpn-bg-bd)#igmp snooping profile prof1
 router(config-l2vpn-bg-bd)#interface TenGigE0/0/0/0
 router(config-l2vpn-bg-bd-ac)#exit
 router(config-l2vpn-bg-bd)#interface TenGigE0/0/0/4.1
 router(config-l2vpn-bg-bd-ac)#exit
 router(config-l2vpn-bg-bd)#interface TenGigE0/0/0/4.2
 router(config-l2vpn-bg-bd-ac)#exit
 router(config-l2vpn-bg-bd)#routed interface BVI1
 router(config-l2vpn-bg-bd-bvi)#exit
 !
 !


 router(config-l2vpn-bg-bd)#mld snooping profile prof1
 router(config-l2vpn-bg-bd)#internal-querier
!
 router(config-l2vpn-bg-bd)#igmp snooping profile prof1
 router(config-l2vpn-bg-bd)#system-ip-address 1.2.3.4
 router(config-l2vpn-bg-bd)#internal-querier
```

Note With BVI configurations, there is no need to have internal queries address configured MLD snooping profile. It implies that you can make BVI as querier under BVI configuration.

Verification

The following command shows the information about group membership in the Layer 2 Forwarding tables.

```
router# show mld snooping group

Flags Key: S=Static, D=Dynamic, E=Explicit Tracking

                 Bridge Domain bg1:bd1

Group           Ver GM  Source           PM  Port                     Exp Flg

Ff12:1:1::1     V2  Exc  -               -   GigabitEthernet0/1/1/0   122  DE
Ff12:1:1::1     V2  Exc  2002:1::1       Inc GigabitEthernet0/1/1/1     5  DE
Ff12:1:1::1     V2  Exc  2002:1::1       Inc GigabitEthernet0/1/1/2 never   S
Ff12:1:1::1     V2  Exc  2002:1::1       Exc GigabitEthernet0/1/1/3     -  DE
Ff12:1:1::1     V2  Exc  2002:1::2       Inc GigabitEthernet0/1/1/0   202  DE
Ff12:1:1::1     V2  Exc  2002:1::2       Exc GigabitEthernet0/1/1/1     -  DE
Ff12:1:1::2     V2  Exc  2002:1::1       Inc GigabitEthernet0/1/1/0   145  DE
Ff12:1:1::2     V2  Exc  2002:1::1       Inc GigabitEthernet0/1/1/1     0  DE
Ff12:1:1::2     V2  Exc  2002:1::1       Exc GigabitEthernet0/1/1/2    11  DE


                 Bridge Domain bg1:bd4

Group           Ver GM  Source           PM  Port                     Exp Flg
```

```
Ff24:1:1::2     V1  Exc  -               -   GigabitEthernet0/1/1/0   122  DE
Ff28:1:1::1     V1  -    -               -   GigabitEthernet0/1/1/1    33  DE
Ff29:1:2::3     V1  Exc  -               -   GigabitEthernet0/1/2/0   122  DE
Ff22:1:2::3     V2  Exc  2000:1:1::2     Exc GigabitEthernet0/1/2/1     5  DE
```

The following command summarizes the number of bridge domains, mrouter ports, host ports, groups, and sources configured on the router.

```
router#show mld snooping summary
  Bridge Domains:                                 1
  MLD  Snooping Bridge Domains:                   1
  Ports:                                          3
  MLD  Snooping Ports:                            3
  Mrouters:                                       0
  STP Forwarding Ports:                           0
  ICCP Group Ports:                               0
  MLD  Groups:                                    0
    Member Ports:                                 0
  MLD  Source Groups:                             0
    Static/Include/Exclude:                   0/0/0
    Member Ports (Include/Exclude):             0/0
```

IPv6 Multicast Listener Discovery Snooping over BVI

Multicast Listener Discovery (MLD) snooping provides a way to constrain multicast traffic at L2. By snooping the MLD membership reports sent by hosts in the bridge domain, the MLD snooping application can set up L2 multicast forwarding tables. This table is later used to deliver traffic only to ports with at least one interested member, significantly reducing the volume of multicast traffic.

MLDv2 support over BVI enables implementing IPv6 multicast routing over a L2 segment of the network that is using an IPv6 VLAN. The multicast routes are bridged via BVI interface from L3 segment to L2 segment of the network.

MLDv2 snooping over BVI enables forwarding MLDv2 membership reports received over the L2 domain to MLD snooping instead of MLD.

Restrictions

- You cannot configure `ttl-check` and disable `router-alert-check` on the router for mld messages.
- Static mrouters are not supported for MLD snooping.
- Querier is supported for MLDV2, but it is not supported on MLDV1.

Configuring Internal Querier for MLD Snooping

This configuration enables a multicast router acting as a MLD querier to send out group-and-source-specific query:

```
router# config
RP0/0/RP0/CPU0:router(config)# mld snooping profile grp1
RP0/0/RP0/CPU0:router(config-mld-snooping-profile)# system-ip-address fe80::1 link-local
RP0/0/RP0/CPU0:router(config-mld-snooping-profile)# internal-querier
RP0/0/RP0/CPU0:router(config-mld-snooping-profile)# commit
```

Verification

Use the **show mld snooping profile detail** command to verify the MLD snooping configuration:

```
router# show mld snooping profile detail
Thu Nov 22 13:58:18.844 UTC
MLD  Snoop Profile grp1:
  System IP Address:                    fe80::1
  Bridge Domain References:             2
  Port References:                      12

MLD  Snoop Profile grp10:
  System IP Address:                    fe80::5610
  Bridge Domain References:             0
  Port References:                      0
```

Creating a MLD Snooping Profile

Configuration

```
/* Enter the global configuration mode */
RP/0/RP0/CPU0:router # configure
/* Enters MLD snooping profile configuration mode and creates a named profile. */
RP/0/RP0/CPU0:router(config)# mld snooping profile default-bd-profile
RP/0/RP0/CPU0:router # commit
```

The default profile enables MLD snooping. You can commit the new profile without any additional configurations, or you can include additional configuration options to the profile. You can also return to the profile later to add configurations, as described in other tasks in this module.

If you are creating a bridge domain profile, consider the following:

- An empty profile is appropriate for attaching to a bridge domain. An empty profile enables MLD snooping with default configuration values.
- You can optionally add more commands to the profile to override default configuration values.
- If you include port-specific configurations in a bridge domain profile, the configurations apply to all ports under the bridge, unless another profile is attached to a port.

If you are creating a port-specific profile, consider the following:

- While an empty profile could be attached to a port, it would have no effect on the port configuration.
- When you attach a profile to a port, MLD snooping reconfigures that port, overriding any inheritance of configuration values from the bridge-domain profile. You must repeat the commands in the port profile if you want to retain those configurations.

You can detach a profile, change it, and reattach it to add commands to a profile at a later time.

Running Configuration

```
RP/0/RP0/CPU0:router(config)# show running-config
configure
   mld snooping profile default-bd-profile
!
```

Verification

Verify that the MLD snooping profile is created:

```
RP/0/RP0/CPU0:router#show mld snooping profile

Profile                                  Bridge Domain       Port
-------                                  -------------       ----
default-bd-profile                                  0          0
grp1                                                1          2
grp10                                               1          2
```

Deactivating MLD Snooping on a Bridge Domain

To deactivate MLD snooping from a bridge domain, remove the profile from the bridge domain:

Note A bridge domain can have only one profile attached to it at a time.

Configuration

```
/* Enter the global configuration mode followed by the bridge group and the bridge domain
mode */
RP0/0/RP0/CPU0:router# configuration
RP0/0/RP0/CPU0:router(config)# l2vpn
RP0/0/RP0/CPU0:router(config-l2vpn)# bridge group GRP1
RP0/0/RP0/CPU0:router(config-l2vpn-bg)# bridge domain ISP1

/* Detache the MLD snooping profile from the bridge domain. This disables MLD snooping on
that bridge domain */
/* Note: Only one profile can be attached to a bridge domain at a time. If a profile is
attached, MLD snooping is enabled.
If a profile is not attached, MLD snooping is disabled. */
RP0/0/RP0/CPU0:router(config-l2vpn-bg-bd)# no mld snooping profile
RP0/0/RP0/CPU0:router(config-l2vpn-bg-bd)# commit
```

Running Configuration

```
RP0/0/RP0/CPU0:router# show running-config
configuration
l2vpn
 bridge-group GRP1
  bridge-domain ISP1
   no mld snooping profile
!
```

Configuring Static Mrouter Ports (MLD)

Prerequisite

MLD snooping must be enabled on the bridge domain for port-specific profiles to affect MLD snooping behavior.

Note Static mrouter port configuration is a port-level option and should be added to profiles intended for ports. It is not recommended to add mrouter port configuration to a profile intended for bridge domains.

Configuration

```
/* Enter the global configuration mode */
RP0/0/RP0/CPU0:router# configuration

/* Enter the MLD snooping profile configuration mode and create a new profile or accesses
an existing profile.*/
RP0/0/RP0/CPU0:router(config)# mld snooping profile mrouter-port-profile
RP0/0/RP0/CPU0:router(config-mld-snooping-profile)# mrouter
/* Configures a static mrouter on a port. */

RP0/0/RP0/CPU0:router(config-mld-snooping-profile)# commit
```

Running Configuration

```
RP0/0/RP0/CPU0:router# show running-config
configuration
  mld snooping profile mrouter-port-profile
    mrouter
!
```

Verification

The below show command output confirms that the mrouter configuration is enabled:

```
RP0/0/RP0/CPU0:router# show mld snooping profile mrouter-port-profile

MLD  Snoop Profile mrouter-port-profile:

  Static Mrouter:                       Enabled

  Bridge Domain References:             0
  Port References:                      0
```

Configuring Immediate-leave for MLD

To add the MLD snooping immediate-leave option to an MLD snooping profile:

Configuration

```
/* Enter the global configuration mode. */
RP0/0/RP0/CPU0:router# configuration

/* Enter MLD snooping profile configuration mode and create a new profile or accesses an
existing profile. */
RP0/0/RP0/CPU0:router(config)# mld snooping profile host-port-profile
/* Enable the immediate-leave option */
RP0/0/RP0/CPU0:router(config-mld-snooping-profile)# immediate-leave
RP0/0/RP0/CPU0:router(config-l2vpn-bg-bd)# commit
```

If you add the **immediate-leave** option:

- to a profile attached to a bridge domain, it applies to all ports under the bridge.
- to a profile attached to a port, it applies to the port.

Running Configuration

```
RP0/0/RP0/CPU0:router# show running-config
configuration
 mld snooping profile host-port-profile
 immediate-leave
!
```

Verification

Verify that the immediate leave config in the named profile is enabled:

```
RP0/0/RP0/CPU0:router# show mld snooping profile host-port-profile detail

MLD  Snoop Profile host-port-profile:

  Immediate Leave:                    Enabled
  Router Guard:                       Enabled

  Bridge Domain References:           0
  Port References:                    0
```

Configuring Internal Querier for MLD

Prerequisite

MLD snooping must be enabled on the bridge domain for this procedure to take effect.

Configuration

```
/* Enter the global configuration mode. */
RP0/0/RP0/CPU0:router# configuration

/* Enter MLD snooping profile configuration mode and create a new profile or accesses an
existing profile. */
RP0/0/RP0/CPU0:router(config)# mld snooping profile internal-querier-profile

/* Configure an IP address for internal querier use. The default system-ip-address value
(0.0.0.0) is not valid for the internal querier.
You must explicitly configure an IP address. Enter a valid link-local IPv6 address. */
RP0/0/RP0/CPU0:router(config-mld-snooping-profile)# system-ip-address fe80::98 link-local

/* Enable an internal querier with default values for all options.*/
RP0/0/RP0/CPU0:router(config-mld-snooping-profile)# internal-querier
RP0/0/RP0/CPU0:router(config-mld-snooping-profile)# commit
```

Running Configuration

```
RP0/0/RP0/CPU0:router# show running-config
configuration
 mld snooping profile internal-querier-profile
 system-ip-address fe80::98 link-local
```

```
 internal-querier
!
```

Note Internal Querier is not recommended, when BVI and MLD snooping is configured under a bridge.

Verification

Verify that the internal querier config is enabled:

```
RP0/0/RP0/CPU0:router# show mld snooping profile internal-querier-profile detail

MLD  Snoop Profile internal-querier-profile:

  System IP Address:                    fe80::98

  Internal Querier Support:             Enabled

  Bridge Domain References:             0
  Port References:                      0
```

Configuring Static Groups for MLD

To add one or more static groups or MLDv2 source groups to an MLD snooping profile, follow these steps:

Prerequisite

MLD snooping must be enabled on the bridge domain for port-specific profiles to affect MLD snooping behavior.

Configuration

```
/* Enter the global configuration mode. */
RP0/0/RP0/CPU0:router# configuration

/* Enter MLD snooping profile configuration mode and create a new profile or accesses an
existing profile. */
RP0/0/RP0/CPU0:router(config)# mld snooping profile host-port-profile

/* Configure a static group. */
/* Note: Repeat this step to add additional static groups. */
RP0/0/RP0/CPU0:router(config-mld-snooping-profile)# static group 239.1.1.1 source 198.168.1.1
RP0/0/RP0/CPU0:router(config-mld-snooping-profile)# commit
```

If you add the **static group** option:

- to a profile attached to a bridge domain, it applies to all ports under the bridge.
- to a profile attached to a port, it applies to the port.

Running Configuration

```
RP0/0/RP0/CPU0:router# show running-config
configuration
 mld snooping profile host-port-profile
```

```
 static group 239.1.1.1 source 198.168.1.1
!
```

Verification

```
RP0/0/RP0/CPU0:router# show mld snooping bridge-domain f1:100 detail

Bridge Domain        Profile                          Act  Ver  #Ports  #Mrtrs  #Grps
#SGs
-------------        -------                          ---  ---  ------  ------  -----
----
f1:100               grp1                              Y  v2       3       1   1000   1002


  Profile Configured Attributes:
    System IP Address:                   fe80::99
    Minimum Version:                     1
    Report Suppression:                  Enabled
    Unsolicited Report Interval:         1000 (milliseconds)
    TCN Query Solicit:                   Disabled
    TCN Membership Sync:                 Disabled
    TCN Flood:                           Enabled
    TCN Flood Query Count:               2
    Router Alert Check:                  Disabled
    TTL Check:                           Enabled
    nV Mcast Offload:                    Disabled
    Internal Querier Support:            Disabled
    Querier Query Interval:              125 (seconds)
    Querier LMQ Interval:                1000 (milliseconds)
    Querier LMQ Count:                   2
    Querier Robustness:                  2
    Startup Query Interval:              31 seconds
    Startup Query Count:                 2
    Startup Query Max Response Time:     10.0 seconds
    Mrouter Forwarding:                  Enabled
    P2MP Capability:                     Disabled
    Default IGMP Snooping profile:       Disabled
    IP Address:                          fe80::f278:16ff:fe63:4d81
    Port:                                BVI1000
    Version:                             v2
    Query Interval:                      125 seconds
    Robustness:                          2
    Max Resp Time:                       10.0 seconds
    Time since last G-Query:             97 seconds
  Mrouter Ports:                         1
    Dynamic:                             BVI1000
  STP Forwarding Ports:                  0
  ICCP Group Ports:                      0
  Groups:                                1000
    Member Ports:                        0
  V2 Source Groups:                      1002
    Static/Include/Exclude:              0/1002/0
    Member Ports (Include/Exclude):      1002/0
```

Configuring MLD Snooping

Configure

```
RP0/0/RP0/CPU0:router# configure
/* Create two profiles. */
RP0/0/RP0/CPU0:router(config)# mld snooping profile bridge_profile
RP0/0/RP0/CPU0:router(config-mld-snooping-profile)# mld snooping profile port_profile
RP0/0/RP0/CPU0:router(config-mld-snooping-profile)# mrouter
RP0/0/RP0/CPU0:router(config-mld-snooping-profile)# exit
RP0/0/RP0/CPU0:router(config)#

/* Configure two physical interfaces for L2 support.*/
RP0/0/RP0/CPU0:router(config)# interface GigabitEthernet0/8/0/38
RP0/0/RP0/CPU0:router(config-if)# negotiation auto
RP0/0/RP0/CPU0:router(config-if)# l2transport
RP0/0/RP0/CPU0:router(config-if)# no shut
RP0/0/RP0/CPU0:router(config-if)# exit
RP0/0/RP0/CPU0:router(config-mld-snooping-profile)# interface GigabitEthernet0/8/0/39
RP0/0/RP0/CPU0:router(config-if)# negotiation auto
RP0/0/RP0/CPU0:router(config-if)# l2transport
RP0/0/RP0/CPU0:router(config-if)# no shut
RP0/0/RP0/CPU0:router(config-if)# exit

/* Add interfaces to the bridge domain. Attach bridge_profile to the bridge domain and
port_profile to one of the Ethernet interfaces.
The second Ethernet interface inherits MLD snooping configuration attributes from the bridge
 domain profile.*/
RP0/0/RP0/CPU0:router(config)# l2vpn
RP0/0/RP0/CPU0:router(config-l2vpn)# bridge group bg1
RP0/0/RP0/CPU0:router(config-l2vpn-bg)# bridge-domain bd1
RP0/0/RP0/CPU0:router(config-l2vpn-bg-bd)# mld snooping profile bridge_profile
RP0/0/RP0/CPU0:router(config-l2vpn-bg-bd-mld-snooping)# interface GigabitEthernet0/8/0/38
RP0/0/RP0/CPU0:router(config-l2vpn-bg-bd-mld-snooping-if)# mld snooping profile port_profile
RP0/0/RP0/CPU0:router(config-l2vpn-bg-bd-mld-snooping-if)# interface GigabitEthernet0/8/0/39
RP0/0/RP0/CPU0:router(config-l2vpn-bg-bd-mld-snooping-if)# exit
RP0/0/RP0/CPU0:router(config-l2vpn-bg-bd-mld-snooping)# exit
RP0/0/RP0/CPU0:router(config-l2vpn-bg-bd)# commit
```

Running Configuration

```
RP0/0/RP0/CPU0:router# show running-config
configuration
 mld snooping profile bridge_profile
!
mld snooping profile port_profile
   mrouter
!

interface GigabitEthernet0/8/0/38
   negotiation auto
   l2transport
   no shut
   !
!
interface GigabitEthernet0/8/0/39
   negotiation auto
   l2transport
   no shut
```

```
   !
!

l2vpn
  bridge group bg1
     bridge-domain bd1
     mld snooping profile bridge_profile
     interface GigabitEthernet0/8/0/38
       mld snooping profile port_profile
     interface GigabitEthernet0/8/0/39
     !
  !
!
```

Verification

Verify the configured bridge ports.

```
RP0/0/RP0/CPU0:router# show mld snooping port


                        Bridge Domain f10:109

                                                        State
Port                                                  Oper  STP  Red   #Grps   #SGs
----                                                  ----  ---  ---   -----   ----
BVI1009                                                 Up    -    -       0      0
GigabitEthernet0/8/0/38                                 Up    -    -    1000   1000
GigabitEthernet0/8/0/39                                 Up    -    -    1000   1000
```

Configuring MLD Snooping on Ethernet Bundles

This example assumes that the front-ends of the bundles are preconfigured. For example, a bundle configuration might consist of three switch interfaces, as follows:

Configure

```
/*  Configure the front-ends of the bundles consisting of three switch interfaces.*/
RP0/0/RP0/CPU0:router# configure
RP0/0/RP0/CPU0:router(config)# interface bundle-ether 1
RP0/0/RP0/CPU0:router(config-if)# exit
RP0/0/RP0/CPU0:router(config)# interface GigabitEthernet0/0/0/0
RP0/0/RP0/CPU0:router(config-if)# exit
RP0/0/RP0/CPU0:router(config)# interface GigabitEthernet0/0/0/1
RP0/0/RP0/CPU0:router(config-if)# exit
RP0/0/RP0/CPU0:router(config)# interface GigabitEthernet0/0/0/2
RP0/0/RP0/CPU0:router(config-if)# channel-group 1 mode on
RP0/0/RP0/CPU0:router(config-if)# exit
RP0/0/RP0/CPU0:router(config)# interface GigabitEthernet0/0/0/3
RP0/0/RP0/CPU0:router(config-if)# channel-group 1 mode on
RP0/0/RP0/CPU0:router(config-if)# exit

/* Configure two MLD snooping profiles. */
RP0/0/RP0/CPU0:router(config)# mld snooping profile bridge_profile
RP0/0/RP0/CPU0:router(config-mld-snooping-profile)# exit     !
RP0/0/RP0/CPU0:router(config)# mld snooping profile port_profile
RP0/0/RP0/CPU0:router(config-mld-snooping-profile)# mrouter
RP0/0/RP0/CPU0:router(config-mld-snooping-profile)# exit
```

```
/* Configure interfaces as bundle member links. */

RP0/0/RP0/CPU0:router(config)# interface GigabitEthernet0/0/0/0
RP0/0/RP0/CPU0:router(config-if)# bundle id 1 mode on
RP0/0/RP0/CPU0:router(config-if)# negotiation auto
RP0/0/RP0/CPU0:router(config-if)# exit
RP0/0/RP0/CPU0:router(config)# interface GigabitEthernet0/0/0/1
RP0/0/RP0/CPU0:router(config-if)# bundle id 1 mode on
RP0/0/RP0/CPU0:router(config-if)# negotiation auto
RP0/0/RP0/CPU0:router(config-if)# exit
RP0/0/RP0/CPU0:router(config)# interface GigabitEthernet0/0/0/2
RP0/0/RP0/CPU0:router(config-if)# bundle id 2 mode on
RP0/0/RP0/CPU0:router(config-if)# negotiation auto
RP0/0/RP0/CPU0:router(config-if)# exit
RP0/0/RP0/CPU0:router(config)# interface GigabitEthernet0/0/0/3
RP0/0/RP0/CPU0:router(config-if)# bundle id 2 mode on
RP0/0/RP0/CPU0:router(config-if)# negotiation auto
RP0/0/RP0/CPU0:router(config-if)# exit

/* Configure the bundle interfaces for L2 transport. */
RP0/0/RP0/CPU0:router(config)# interface Bundle-Ether 1
RP0/0/RP0/CPU0:router(config-if)# l2transpor
RP0/0/RP0/CPU0:router(config-if)# exit
RP0/0/RP0/CPU0:router(config)# interface Bundle-Ether 2
RP0/0/RP0/CPU0:router(config-if)# l2transpor
RP0/0/RP0/CPU0:router(config-if)# exit

/* Add the interfaces to the bridge domain and attach MLD snooping profiles. */
RP0/0/RP0/CPU0:router(config)# l2vpn
RP0/0/RP0/CPU0:router(config-l2vpn)# bridge group bg1
RP0/0/RP0/CPU0:router(config-l2vpn-bg)# mld snooping profile bridge_profile
RP0/0/RP0/CPU0:router(config-l2vpn-bg-mld-snooping-profile)# interface bundle-Ether 1
RP0/0/RP0/CPU0:router(config-l2vpn-bg-mld-snooping-profile-if)# mld snooping profile
port_profile
RP0/0/RP0/CPU0:router(config-l2vpn-bg-mld-snooping-profile-if)# interface bundle-Ether 2
RP0/0/RP0/CPU0:router(config-l2vpn-bg-mld-snooping-profile-if)# commit
```

Running Configuration

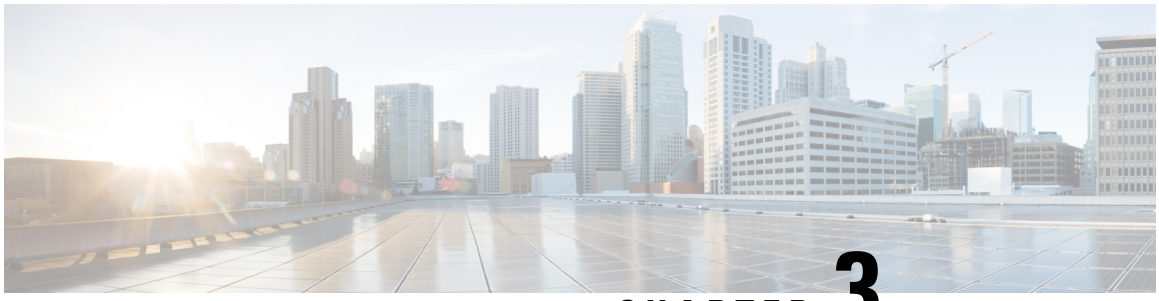
```
RP0/0/RP0/CPU0:router# show running-config
configuration
 interface Port-channel1
    !
interface GigabitEthernet0/0/0/0
    !
interface GigabitEthernet0/0/0/1
!
    interface GigabitEthernet0/0/0/2
       channel-group 1 mode on
    !
    interface GigabitEthernet0/0/0/3
       channel-group 1 mode on
    !
mld snooping profile bridge_profile
       !
       mld snooping profile port_profile
          mrouter
       !
interface GigabitEthernet0/0/0/0
         bundle id 1 mode on
         negotiation auto
       !
```

```
       interface GigabitEthernet0/0/0/1
         bundle id 1 mode on
         negotiation auto
       !
       interface GigabitEthernet0/0/0/2
         bundle id 2 mode on
         negotiation auto
       !
       interface GigabitEthernet0/0/0/3
         bundle id 2 mode on
         negotiation auto
       !
interface Bundle-Ether 1
          l2transport
          !
       !
       interface Bundle-Ether 2
          l2transport
          !
       !

l2vpn
         bridge group bg1
            bridge-domain bd1
            mld snooping profile bridge_profile
            interface bundle-Ether 1
              mld snooping profile port_profile
            interface bundle-Ether 2
            !
         !
      !
```

Verification

```
RP0/0/RP0/CPU0:router# show mld snooping port
Bridge Domain BG1:BD1
State
Port Oper STP Red #Grps #SGs
---- ---- --- --- ----- ----
HundredGigE0/0/0/3 Up - - 1 1
HundredGigE0/0/0/7 Up - - 1 1
HundredGigE0/19/0/11 Up - - 1 1
HundredGigE0/19/0/5 Up - - 1 1
RP/0/RP1/CPU0:Router#
```



CHAPTER 3

Implementing Layer-3 Multicast Routing

Want to deliver messages like corporate communications or newsletters to subscribed members using a minimum of network bandwidth?

With the traditional method like unicast, you can send messages from one source to one destination. Each host added to the network consumes bandwidth and it's a challenge to reduce the load on the traffic.

On the other hand, broadcast sends messages to all the hosts in the network and not to the selected members.

Enable Multicast routing to deliver data traffic efficiently from a single source to multiple users or selected members or even a group. It's scalable and yet reduces the load on the traffic.

Learn about Multicast

Many applications such as video conferencing, corporate communications, distance learning, and distribution of software, stock quotes, and news involve multiple participants. Multicast is naturally suitable for this communication paradigm.

Unlike unicast and broadcast, multicast allows a host to send a single data stream to a subset of hosts (group transmission) at about the same time. The IP hosts subscribed to a group are known as group members.

A multicast address is chosen from the multicast group. The sender uses that group address as the destination address of a datagram to reach all members of the group

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 IP unicast packets.

Membership in a multicast group is dynamic; hosts can join and leave at any time. There's 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.

To send messages, multicast routing uses the following components:

- The sender or the source address
- The receiver or the multicast address

  The receiver can be a group of members and are identified by a single multicast group address that falls under the IP Class D address range from 224.0.0.0 through 239.255.255.255. A multicast address is chosen for the receivers in a multicast group. Senders use that group address as the destination address of a datagram to reach all members of the group.

Note Any host, regardless of whether it's a member of a group or not, can send to a group. However, only the members of a group receive the message

- A protocol to identify the selected users to send a message.

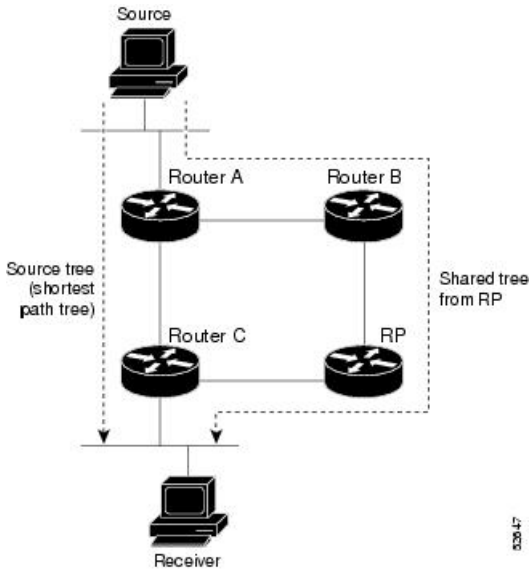
Cisco IOS XR Software supports the following protocols to implement multicast routing:

- IGMP (IPv4): Use IGMP to allow hosts (IPv4) to communicate with routers to express the interest to receive multicast traffic on specific groups. Use Multicast Listener Discovery (MLD v1/2) for IPv6.
- Protocol Independent Multicast in sparse mode (PIM-SM): Use PIM-SM between routers to track which multicast packets to forward to each other and to their directly connected LANs.
- Protocol Independent Multicast in Source-Specific Multicast (PIM-SSM): PIM-SSM is similar to PIM-SM. Hosts use PIM-SSM to report interest in receiving packets from specific source addresses.

  PIM-SSM is made possible by IGMPv3 and MLDv2. Hosts can now indicate interest in specific sources using IGMPv3 and MLDv2. SSM doesn't require a rendezvous point (RP) to operate.

This image shows IGMP and PIM-SM operating in a multicast environment.

Figure 1: Multicast Routing Protocols



- Key Protocols and Features Supported in the Cisco IOS XR Software Multicast Routing Implementation, on page 43
- Prerequisites for Implementing Multicast Routing, on page 44
- Restrictions for Implementing Multicast Routing, on page 44
- Configuring Multicast, on page 44
- Internet Group Management Protocol , on page 45
- Protocol Independent Multicast, on page 47
- Multicast-Only Fast Reroute, on page 62
- Multicast Source Discovery Protocol, on page 70

- Multicast Nonstop Forwarding, on page 70
- Multicast Configuration Submodes, on page 71
- Understanding Interface Configuration Inheritance, on page 72
- Understanding Interface Configuration Inheritance Disablement, on page 73
- Understanding Enabling and Disabling Interfaces, on page 73
- Controlling Source Information on MSDP Peer Routers, on page 74
- Multicast Routing Information Base, on page 75
- Multicast Forwarding Information Base, on page 75
- MSDP MD5 Password Authentication, on page 76
- Label Switch Multicast, on page 76
- Label switched multicast (LSM) multicast label distribution protocol (mLDP) based multicast VPN (mVPN) support, on page 78
- mLDP Loop-Free Alternative Fast Reroute, on page 85
- Rosen-GRE Multicast VPN for Profiles 0, 3, and 11, on page 97
- Multicast Route Statistics, on page 107
- MVPN Ingress Replication Over Dynamic TE-Tunnels, on page 111

Key Protocols and Features Supported in the Cisco IOS XR Software Multicast Routing Implementation

Table 10: Supported Features for IPv4 and IPv6

Feature	IPv4 Support	IPv6 Support
Auto-RP	Yes	No
BGP	Yes	Yes
BSR	Yes	Yes
Dynamic host registration	Yes (IGMP v2/3)	Yes (MLD v1/2)
Explicit tracking of hosts, groups, and channels	Yes (IGMP v3)	Yes
MSDP	Yes	No
Multicast NSF	Yes	Yes
OOR handling	Yes	Yes
PIM-SM	Yes	Yes
PIM-SSM	Yes	Yes
PIM-SSM Mapping	Yes	Yes

Prerequisites for Implementing Multicast Routing

- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
- You must be familiar with IPv4 and IPv6 multicast routing configuration tasks and concepts.
- Unicast routing must be operational.
- To enable multicast VPN, configure a VPN routing and forwarding (VRF) instance.

Restrictions for Implementing Multicast Routing

- The following features are not supported:
  - InterAS Option A
  - PIM Bidir
- IPv6 Multicast destination addresses are only allowed with a /96 mask. IPv6 Multicast destination address should vary only in the last 32 bits of the group address. If they vary outside this range, they might map to the same entry in the hardware.
- Restart of IPv4 or IPv6 multicast forwarding partner process will result in reloading the line card in modular systems or reloading the router in fixed/centralized systems.

Configuring Multicast

To configure multicast, perform the following configuration:

```
Router#configure
Router(config)# multicast-routing
Router(config-mcast)#address-family ipv4
Router(config-mcast-default-ipv4)#interface all enable
Router(config-mcast-default-ipv4)#exit
Router(config-mcast)#router igmp
Router(config-igmp)#version 3
Router(config-igmp)#commit
Tue Feb  4 04:43:37.679 UTC
Router(config-igmp)#exit
Router(config)#exit
```

Verification

```
Router#show pim ipv4 group-map
Tue Feb  4 04:48:29.003 UTC

IP PIM Group Mapping Table
(* indicates group mappings being used)
(+ indicates BSR group mappings active in MRIB)
```

```
Group Range         Proto Client   Groups RP address      Info

224.0.1.39/32*      DM    perm     0      0.0.0.0
224.0.1.40/32*      DM    perm     0      0.0.0.0
224.0.0.0/24*       NO    perm     0      0.0.0.0
232.0.0.0/8*        SSM   config   0      0.0.0.0
224.0.0.0/4*        SM    static   0      0.0.0.0         RPF: Null,0.0.0.0
```

To view the PIM topology table information for a specific group or all groups.

```
Router#show pim topology
IP PIM Multicast Topology Table
Entry state: (*/S,G)[RPT/SPT] Protocol Uptime Info
Entry flags: KAT - Keep Alive Timer, AA - Assume Alive, PA - Probe Alive
    RA - Really Alive, IA - Inherit Alive, LH - Last Hop
    DSS - Don't Signal Sources,  RR - Register Received
    SR - Sending Registers, SNR - Sending Null Registers
    E - MSDP External, EX - Extranet
    MFA - Mofrr Active, MFP - Mofrr Primary, MFB - Mofrr Backup
    DCC - Don't Check Connected, ME - MDT Encap, MD - MDT Decap
    MT - Crossed Data MDT threshold, MA - Data MDT Assigned
    SAJ - BGP Source Active Joined, SAR - BGP Source Active Received,
    SAS - BGP Source Active Sent, IM - Inband mLDP, X - VxLAN
Interface state: Name, Uptime, Fwd, Info
Interface flags: LI - Local Interest, LD - Local Dissinterest,
    II - Internal Interest, ID - Internal Dissinterest,
    LH - Last Hop, AS - Assert, AB - Admin Boundary, EX - Extranet,
    BGP - BGP C-Multicast Join, BP - BGP Source Active Prune,
    MVS - MVPN Safi Learned, MV6S - MVPN IPv6 Safi Learned

(*,224.0.1.40) DM Up: 00:56:47 RP: 0.0.0.0
JP: Null(never) RPF: Null,0.0.0.0 Flags: LH DSS
  Loopback0                   00:56:47  off LI II LH

(21.5.7.2,232.1.1.1)SPT SSM Up: 00:00:44
JP: Join(00:00:05) RPF: Null,0.0.0.0 Flags:
  FourHundredGigE0/0/0/11     00:00:44  fwd LI LH
```

Internet Group Management Protocol

Cisco IOS XR Software provides support for Internet Group Management Protocol (IGMP) over IPv4.

IGMP provides a means for hosts to indicate which multicast traffic they are interested in and for routers to control and limit the flow of multicast traffic throughout the network. Routers build state by means of IGMP and MLD messages; that is, router queries and host reports.

A set of queries and hosts that receive multicast data streams from the same source is called a *multicast group*. Hosts use IGMP and MLD messages to join and leave multicast groups.

Note IGMP messages use group addresses, which are Class D IP addresses. The high-order four bits of a Class D address are 1110. Host group addresses can be in the range 224.0.0.0 to 239.255.255.255. The address 224.0.0.0 is guaranteed not to be assigned to any group. The address 224.0.0.1 is assigned to all systems on a subnet. The address 224.0.0.2 is assigned to all routers on a subnet.

IGMP Versions

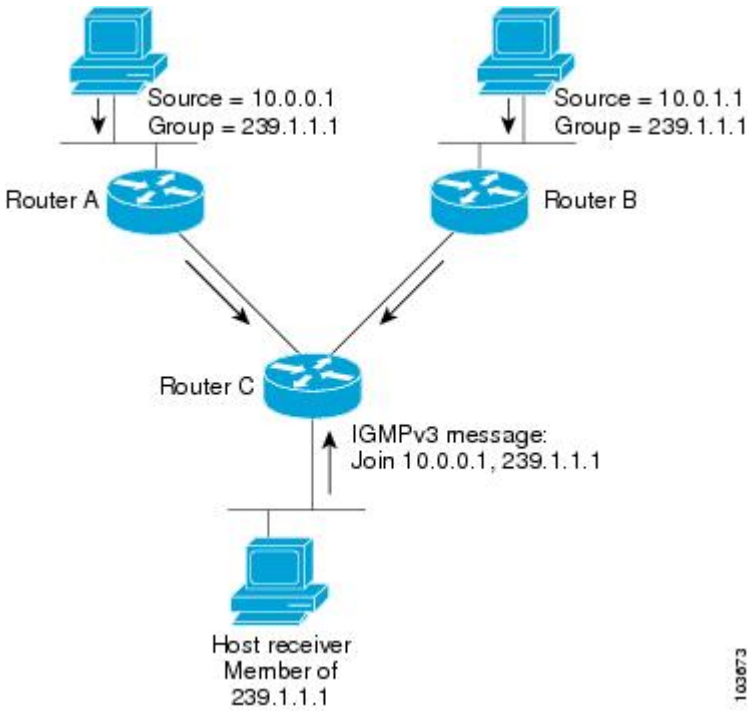
The following points describe IGMP versions 2, and 3:

- IGMP Version 2 extends IGMP allowing such features as the IGMP query timeout and the maximum query-response time. See RFC 2236.
- IGMP Version 3 permits joins and leaves for certain source and group pairs instead of requesting traffic from all sources in the multicast group.

IGMP Routing Example

The below image illustrates two sources, 10.0.0.1 and 10.0.1.1, that are multicasting to group 239.1.1.1. The receiver wants to receive traffic addressed to group 239.1.1.1 from source 10.0.0.1 but not from source 10.0.1.1. The host must send an IGMPv3 message containing a list of sources and groups (S, G) that it wants to join and a list of sources and groups (S, G) that it wants to leave. Router C can now use this information to prune traffic from Source 10.0.1.1 so that only Source 10.0.0.1 traffic is being delivered to Router C.

Figure 2: IGMPv3 Signaling



Note When configuring IGMP, ensure that all systems on the subnet support the same IGMP version. The router does not automatically detect Version 1 systems. Configure the router for Version 2 if your hosts do not support Version 3.

Configuring IGMP Per Interface States Limit

The IGMP Per Interface States Limit sets a limit on creating OLEs for the IGMP interface. When the set limit is reached, the group is not accounted against this interface but the group can exist in IGMP context for some other interface.

The following configuration sets a limit on the number of group memberships created on an interface as a result of receiving IGMP or MLD membership reports.

```
router igmp | mld [vrf <vrfname>]
            interface <ifname>
                        (no) maximum groups-per-interface <max> [threshold <threshold>]
[<acl>]
            !
!
```

where,

`<ifname>` is the interface name

`<max>` is the maximum limit on the groups

`<threshold>` is the threshold number of groups at which point a syslog warning message will be issued

`<acl>` provides an option for selective accounting. If provided, only groups or (S,G)s that are permitted by the ACL is accounted against the limit. Groups or (S, G)s that are denied by the ACL are not accounted against the limit. If not provided, all the groups are accounted against the limit.

The following messages are displayed when the threshold limit is reached for IGMP:

```
igmp[1160]: %ROUTING-IPV4_IGMP-4-OOR_THRESHOLD_REACHED : Threshold for Maximum number of
group per interface has been reached 3: Groups joining will soon be throttled.
Config a higher max or take steps to reduce states

igmp[1160]: %ROUTING-IPV4_IGMP-4-OOR_LIMIT_REACHED : Maximum number of group per interface
 has been reached 6: Groups joining is throttled.
Config a higher max or take steps to reduce states
```

Limitations

- If a user has configured a maximum of 20 groups and has reached the maximum number of groups, then no more groups can be created. If the user reduces the maximum number of groups to 10, the 20 joins will remain and a message of reaching the maximum is displayed. No more joins can be added until the number of groups has reached less than 10.
- If a user already has configured a maximum of 30 joins and add a max of 20, the configuration occurs displaying a message that the maximum has been reached. No state change occurs and also no more joins can occur until the threshold number of groups is brought down below the maximum number of groups.

Protocol Independent Multicast

Protocol Independent Multicast (PIM) is a routing protocol designed to send and receive multicast routing updates. Proper operation of multicast depends on knowing the unicast paths towards a source or an RP. PIM

relies on unicast routing protocols to derive this reverse-path forwarding (RPF) information. As the name PIM implies, it functions independently of the unicast protocols being used. PIM relies on the Routing Information Base (RIB) for RPF information.

If the multicast subsequent address family identifier (SAFI) is configured for Border Gateway Protocol (BGP), or if multicast intact is configured, a separate multicast unicast RIB is created and populated with the BGP multicast SAFI routes, the intact information, and any IGP information in the unicast RIB. Otherwise, PIM gets information directly from the unicast SAFI RIB. Both multicast unicast and unicast databases are outside of the scope of PIM.

The Cisco IOS XR implementation of PIM is based on RFC 4601 Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification. For more information, see RFC 4601 and the Protocol Independent Multicast (PIM): Motivation and Architecture Internet Engineering Task Force (IETF) Internet draft.

Note Cisco IOS XR Software supports PIM-SM, PIM-SSM, and PIM Version 2 only. PIM Version 1 hello messages that arrive from neighbors are rejected.

PIM-Sparse Mode

Typically, PIM in sparse mode (PIM-SM) operation is used in a multicast network when relatively few routers are involved in each multicast. Routers do not forward multicast packets for a group, unless there is an explicit request for traffic. Requests are accomplished using PIM join messages, which are sent hop by hop toward the root node of the tree. The root node of a tree in PIM-SM is the rendezvous point (RP) in the case of a shared tree or the first-hop router 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 sources that send multicast packets are registered with the RP by the first-hop router of the source.

As a PIM join travels up the tree, routers along the path set up the multicast forwarding state so that the requested multicast traffic is forwarded back down the tree. When multicast traffic is no longer needed, a router sends a PIM prune message 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 router updates its forwarding state appropriately. Ultimately, the forwarding state associated with a multicast group or source is removed. Additionally, if prunes are not explicitly sent, the PIM state will timeout and be removed in the absence of any further join messages.

PIM-SM is the best choice for multicast networks that have potential members at the end of WAN links.

PIM-Source Specific Multicast

Table 11: Feature History Table

Feature Name	Release Information	Feature Description
PIM-Source Specific Multicast	Release 24.4.1	Introduced in this release on: Fixed Systems (8700) (select variants only*) Enhance multicast efficiency by focusing on specific source transmissions to receivers, minimizing unwanted traffic. This feature allows routers to maintain multicast traffic solely from designated sources, improving bandwidth usage and reducing network congestion. By implementing tighter source filtering, it enhances security and control within multicast routing, ensuring that only desired data streams reach their intended receivers. This precise routing capability offers administrators improved management and reliability in multicast communication environments. *This functionality is now supported on Cisco 8712-MOD-M routers.

When PIM-SM is used with SSM, multi-cast routing is easier to manage. This is because RPs (rendezvous points) are not required and therefore, no shared trees (*,G) are built.

There is no specific IETF document defining PIM-SSM. However, RFC4607 defines the overall SSM behavior.

In the rest of this document, we use the term PIM-SSM to describe PIM behavior and configuration when SSM is used.

PIM in Source-Specific Multicast operation uses information found on source addresses for a multicast group provided by receivers and performs source filtering on traffic.

- By default, PIM-SSM operates in the 232.0.0.0/8 multicast group range for IPv4. To configure these values, use the **ssm range** command.
- If SSM is deployed in a network already configured for PIM-SM, only the last-hop routers must be upgraded with Cisco IOS XR Software that supports the SSM feature.
- No MSDP SA messages within the SSM range are accepted, generated, or forwarded.
- SSM can be disabled using the **ssm disable** command.
- The ssm allow-override command allows SSM ranges to be overridden by more specific ranges.

In many multicast deployments where the source is known, protocol-independent multicast-source-specific multicast (PIM-SSM) mapping is the obvious multicast routing protocol choice to use because of its simplicity. Typical multicast deployments that benefit from PIM-SSM consist of entertainment-type solutions like the ETTH space, or financial deployments that completely rely on static forwarding.

In SSM, delivery of data grams is based on (S,G) channels. Traffic for one (S,G) channel consists of datagrams with an IP unicast source address S and the multicast group address G as the IP destination address. Systems receive traffic by becoming members of the (S,G) channel. Signaling is not required, but receivers must subscribe or unsubscribe to (S,G) channels to receive or not receive traffic from specific sources. Channel subscription signaling uses IGMP to include mode membership reports, which are supported only in Version 3 of IGMP (IGMPv3).

To run SSM with IGMPv3, SSM must be supported on the multicast router, the host where the application is running, and the application itself. Cisco IOS XR Software allows SSM configuration for an arbitrary subset of the IP multicast address range 224.0.0.0 through 239.255.255.255.

When an SSM range is defined, existing IP multicast receiver applications do not receive any traffic when they try to use addresses in the SSM range, unless the application is modified to use explicit (S,G) channel subscription.

Benefits of PIM-SSM over PIM-SM

PIM-SSM is derived from PIM-SM. However, whereas PIM-SM allows for the data transmission of all sources sending to a particular group in response to PIM join messages, the SSM feature forwards traffic to receivers only from those sources that the receivers have explicitly joined. Because PIM joins and prunes are sent directly towards the source sending traffic, an RP and shared trees are unnecessary and are disallowed. SSM is used to optimize bandwidth utilization and deny unwanted Internet broad cast traffic. The source is provided by interested receivers through IGMPv3 membership reports.

PIM-SM and PIM-SSM

Protocol Independent Multicast (PIM) is a multicast routing protocol used to create multicast distribution trees, which are used to forward multicast data packets. PIM is an efficient IP routing protocol that is "independent" of a routing table, unlike other multicast protocols such as Multicast Open Shortest Path First (MOSPF) or Distance Vector Multicast Routing Protocol (DVMRP).

Cisco IOS XR Software supports Protocol Independent Multicast in sparse mode (PIM-SM) and Protocol Independent Multicast in Source-Specific Multicast (PIM-SSM), permitting these modes to operate on your router at the same time.

PIM-SM and PIM-SSM supports one-to-many applications by greatly simplifying the protocol mechanics for deployment ease.

- PIM in sparse mode operation is used in a multicast network when relatively few routers are involved in each multicast and these routers do not forward multicast packets for a group, unless there is an explicit request for the traffic.
- PIM in Source-Specific Multicast operation uses information found on source addresses for a multicast group provided by receivers and performs source filtering on traffic.
    - By default, PIM-SSM operates in the 232.0.0.0/8 multicast group range for IPv4 and ff3x::/32 (where x is any valid scope) in IPv6. To configure these values, use the **ssm range** command.
    - If SSM is deployed in a network already configured for PIM-SM, only the last-hop routers must be upgraded with Cisco IOS XR Software that supports the SSM feature.
    - No MSDP SA messages within the SSM range are accepted, generated, or forwarded.

Restrictions for PIM-SM and PIM-SSM

- Interoperability with SSM:

  PIM-SM operations within the SSM range of addresses change to PIM-SSM. In this mode, only PIM (S,G) join and prune messages are generated by the router, and no (S,G) RP shared tree or (*,G) shared tree messages are generated.

- **IGMP Version:**

To report multicast memberships to neighboring multicast routers, hosts use IGMP, and all routers on the subnet must be configured with the same version of IGMP.

A router running Cisco IOS XR Software does not automatically detect Version 1 systems. You must use the **version** command in router IGMP configuration submode to configure the IGMP version.

Configuring PIM-SSM for Use in a Legacy Multicast Deployment

Deploying PIM-SSM in legacy multicast-enabled networks can be problematic, because it requires changes to the multicast group management protocols used on the various devices attached to the network. Host, routers, and switches must all be upgraded in such cases.

To support legacy hosts and switches in a PIM-SSM deployment, this router offers a configurable mapping feature. Legacy group membership reports for groups in the SSM group range are mapped to a set of sources providing service for that set of (S,G) channels.

Restrictions for PIM-SSM Mapping

PIM-SSM mapping does not modify the SSM group range. Instead, the legacy devices must report group membership for desired groups in the SSM group range.

Configuration Example

To creconfigure PIM-SSM for use in a legacy multicast deployment, you must complete the following configurations:

1. Configuring a Set of Access Control Lists for Static SSM Mapping
2. Configuring a Set of Sources for SSM Mapping

Configuration

To configure a set of access control lists (ACLs) where each ACL describes a set of SSM groups to be mapped to one or more sources:

```
Router#configure
Tue Feb  4 05:15:56.544 UTC
Router(config)#ipv4 access-list mc3
Router(config-ipv4-acl)#permit 1 host 232.1.1.2 any
Router(config-ipv4-acl)#commit
Tue Feb  4 05:16:28.752 UTC
Router(config-ipv4-acl)#exit
Router(config)#exit
Router:ios#
```

To configure a set of sources mapped by SSM groups:

```
Router#configure
Router(config)#router igmp vrf vrf20
Router(config-igmp-vrf20)#ssm map static 232.1.1.1 mc2
Router(config-igmp-vrf20)#exit
Router(config-igmp)#commit
```

Configuring PIM Per Interface States Limit

The PIM Per Interface States Limit sets a limit on creating OLEs for the PIM interface. When the set limit is reached, the group is not accounted against this interface but the group can exist in PIM context for some other interface.

The following configuration sets a limit on the number of routes for which the given interface may be an outgoing interface as a result of receiving a PIM J/P message.

```
router pim | pim6 [vrf <vrfname>]
interface <ifname>
          maximum route-interfaces <max> [threshold <threshold>] [<acl>]
!
!
```

where,

`<ifname>` is the interface name

`<max>` is the maximum limit on the groups

`<threshold>` is the threshold number of groups at which point a syslog warning message will be issued

`<acl>` provides an option for selective accounting. If provided, only groups or (S,G)s that are permitted by the ACL is accounted against the limit. Groups or (S, G)s that are denied by the ACL are not accounted against the limit. If not provided, all the groups are accounted against the limit.

The following messages are displayed when the threshold limit is reached for PIM:

```
pim[1157]: %ROUTING-IPV4_PIM-4-CAC_STATE_THRESHOLD : The interface GigabitEthernet0_2_0_0
threshold number (4) allowed states has been reached.
State creation will soon be throttled. Configure a higher state limit value or take steps
to reduce the number of states.
```

```
pim[1157]: %ROUTING-IPV4_PIM-3-CAC_STATE_LIMIT : The interface GigabitEthernet0_2_0_0 maximum
 number (5) of allowed states has been reached.
State creation will not be allowed from here on. Configure a higher maximum value or take
steps to reduce the number of states
```

Limitations

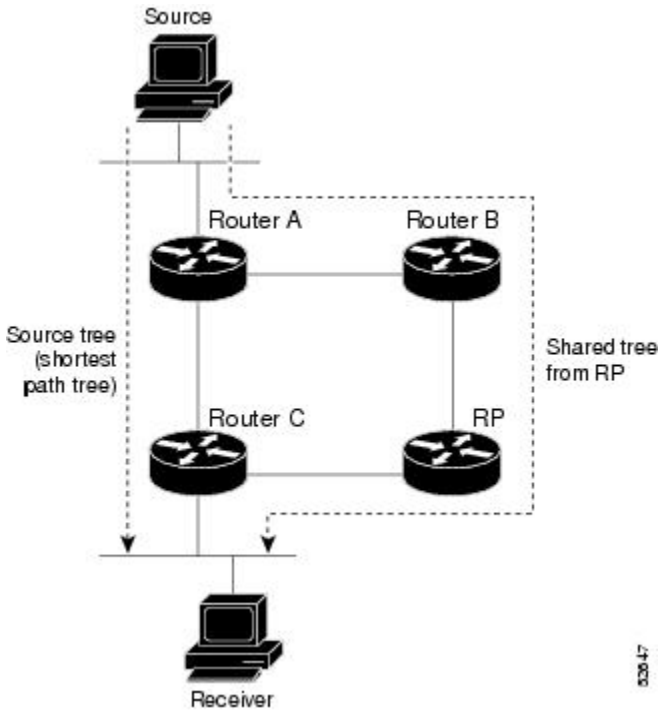
- If a user has configured a maximum of 20 groups and has reached the maximum number of groups, then no more groups/OLEs can be created. If the user now decreases the maximum number to 10, the 20 joins/OLE will remain and a message of reaching the max is displayed. No more joins/OLE can be added at this point until it has reached less than 10.

- If a user already has configured a maximum of 30 joins/OLEs and add a max of 20, the configuration occurs displaying a message that the max has been reached. No states will change but no more joins/OLEs can happen until the number is brought down below the maximum number of groups.
- Local interest joins are added, even if the limit has reached and is accounted for it.

PIM Shared Tree and Source Tree (Shortest Path Tree)

In PIM-SM, the rendezvous point (RP) is used to bridge sources sending data to a particular group with receivers sending joins for that group. In the initial setup of state, interested receivers receive data from senders to the group across a single data distribution tree rooted at the RP. This type of distribution tree is called a shared tree or rendezvous point tree (RPT) as illustrated in the below image. Data from senders is delivered to the RP for distribution to group members joined to the shared tree.

Figure 3: Shared Tree and Source Tree (Shortest Path Tree)



Unless the **spt-threshold infinity** command is configured, this initial state gives way as soon as traffic is received on the leaf routers (designated router closest to the host receivers). When the leaf router receives traffic from the RP on the RPT, the router initiates a switch to a data distribution tree rooted at the source sending traffic. This type of distribution tree is called a **shortest path tree** or **source tree**. By default, the Cisco IOS XR Software switches to a source tree when it receives the first data packet from a source.

The following process describes the move from shared tree to source tree in more detail:

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

8. RP deletes the link to Router C from outgoing interface of (S,G). RP triggers a prune message toward Source.

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

Tip The **spt-threshold infinity** command lets you configure the router so that it never switches to the shortest path tree (SPT).

Multicast-Intact

The multicast-intact feature provides the ability to run multicast routing (PIM) when Interior Gateway Protocol (IGP) shortcuts are configured and active on the router. Both Open Shortest Path First, version 2 (OSPFv2), and Intermediate System-to-Intermediate System (IS-IS) support the multicast-intact feature. Multiprotocol Label Switching Traffic Engineering (MPLS-TE) and IP multicast coexistence is supported in Cisco IOS XR Software by using the **mpls traffic-eng multicast-intact** IS-IS or OSPF router command. See the Routing Configuration Guide for Cisco 8000 Series Routers for information on configuring multicast intact using IS-IS and OSPF commands.

You can enable multicast-intact in the IGP when multicast routing protocols (PIM) are configured and IGP shortcuts are configured on the router. IGP shortcuts are MPLS tunnels that are exposed to IGP. The IGPs route the IP traffic over these tunnels to destinations that are downstream from the egress router of the tunnel (from an SPF perspective). PIM cannot use IGP shortcuts for propagating PIM joins because reverse path forwarding (RPF) cannot work across a unidirectional tunnel.

When you enable multicast-intact on an IGP, the IGP publishes a parallel or alternate set of equal-cost next-hops for use by PIM. These next-hops are called **mcast-intact next-hops**. The mcast-intact next-hops have the following attributes:

- They are guaranteed not to contain any IGP shortcuts.
- They are not used for unicast routing but are used only by PIM to look up an IPv4 next hop to a PIM source.
- They are not published to the Forwarding Information Base (FIB).
- When multicast-intact is enabled on an IGP, all IPv4 destinations that were learned through link-state advertisements are published with a set equal-cost mcast-intact next-hops to the RIB. This attribute applies even when the native next-hops have no IGP shortcuts.
- In IS-IS, the max-paths limit is applied by counting both the native and mcast-intact next-hops together. (In OSPFv2, the behavior is slightly different.)

Designated Routers

Cisco routers use PIM-SM to forward multicast traffic and follow an election process to select a designated router (DR) when there is more than one router on a LAN segment.

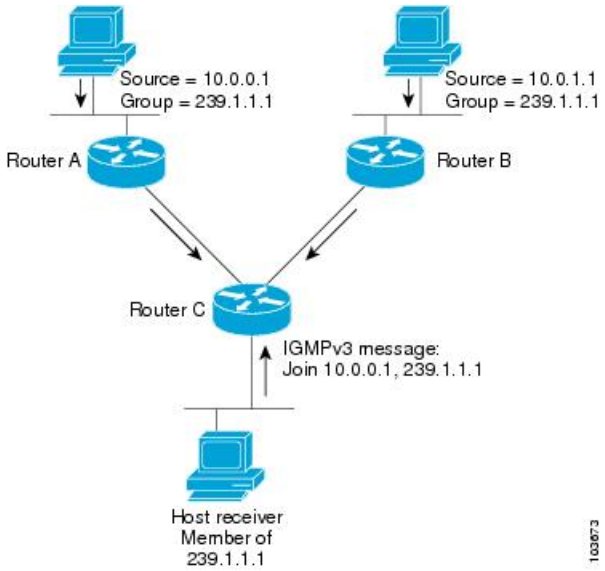
The designated router is responsible for sending PIM register and PIM join and prune messages toward the RP to inform it about host group membership.

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

Note DR election process is required only on multi access LANs. The last-hop router directly connected to the host is the DR.

The figure "Designated Router Election on a Multiaccess Segment", below illustrates what happens on a multi access segment. Router A (10.0.0.253) and Router B (10.0.0.251) are connected to a common multi access Ethernet segment with Host A (10.0.0.1) as an active receiver for Group A. As the Explicit Join model is used, only Router A, operating as the DR, sends joins to the RP to construct the shared tree for Group A. If Router B were also permitted to send (*,G) joins to the RP, parallel paths would be created and Host A would receive duplicate multicast traffic. When Host A begins to source multicast traffic to the group, the DR's responsibility is to send register messages to the RP. Again, if both routers were assigned the responsibility, the RP would receive duplicate multicast packets.

Figure 4: Designated Router Election on a Multiaccess Segment



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

Note Two PIM routers are neighbors if there is a direct connection between them. To display your PIM neighbors, use the show pim neighbor command in EXEC mode.

- They are not used for unicast routing but are used only by PIM to look up an IPv4 next hop to a PIM source.
- They are not published to the Forwarding Information Base (FIB).
- When multicast-intact is enabled on an IGP, all IPv4 destinations that were learned through link-state advertisements are published with a set equal-cost mcast-intact next-hops to the RIB. This attribute applies even when the native next-hops have no IGP shortcuts.
- In IS-IS, the max-paths limit is applied by counting both the native and mcast-intact next-hops together. (In OSPFv2, the behavior is slightly different.)

Configuration Example

Configures the router to use DR priority 4 for TenGigE interface 0/0/0/1, but other interfaces will inherit DR priority 2:

```
Router#configure
Router(config)#router pim
Router(config-pim-default)#address-family ipv4
Router(config-pim-default-ipv4)#dr-priority 2
Router(config-pim-default-ipv4)#interface TenGigE0/0/0/1
Router(config-pim-ipv4-if)#dr-priority 4
Router(config-ipv4-acl)#commit
```

Running Configuration

```
Router#show run router pim
router pim
 address-family ipv4
  dr-priority 2
  spt-threshold infinity
  interface TenGigE 0/0/0/1
   dr-priority 4
   hello-interval 45
```

Verification

Verify if the parameters are set according to the configured values:

```
Router#show pim interface
PIM interfaces in VRF default
Address           Interface           PIM  Nbr   Hello  DR    DR Count Intvl  Prior
100.1.1.1         TenGigE0/0/0/1       on   1     45     4     this system
26.1.1.1          TenGigE0/0/0/26      on   1     30     2     this system
```

Rendezvous Points

When PIM is configured in sparse mode, you must choose one or more routers to operate as a rendezvous point (RP). A rendezvous point is a single common root placed at a chosen point of a shared distribution tree,

as illustrated in PIM Shared Tree and Source Tree (Shortest Path Tree), on page 53. A rendezvous point can be either configured statically in each box or learned through a dynamic mechanism.

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

- Encapsulated in register packets and unicast directly to the rendezvous point by the first-hop router operating as the DR.
- Multicast forwarded by the RPF forwarding algorithm, described in the Reverse-Path Forwarding, on page 60, if the rendezvous point has itself joined the source tree.

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

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

You must manually configure a PIM router to function as a rendezvous point.

Configuration Example

The following example shows how to configure a static RP and allow backward compatibility:

```
RP/0/RP0/CPU0:ios#configure
Thu Jan 30 08:30:02.187 UTC
RP/0/RP0/CPU0:ios(config)#router pim
RP/0/RP0/CPU0:ios(config-pim)#old-register-checksum
RP/0/RP0/CPU0:ios(config-pim)#exit
RP/0/RP0/CPU0:ios(config)#ipv4 access-list rp-access
RP/0/RP0/CPU0:ios(config-ipv4-acl)#permit 239.1.1.0 0.0.255.255
RP/0/RP0/CPU0:ios(config-ipv4-acl)#commit
Thu Jan 30 08:31:22.679 UTC
RP/0/RP0/CPU0:ios(config-ipv4-acl)#
```

Auto-RP

Automatic route processing (Auto-RP) is a feature that automates the distribution of group-to-RP mappings in a PIM network. This feature has these benefits:

- It is easy to use multiple RPs within a network to serve different group ranges.
- It allows load splitting among different RPs.
- It facilitates the arrangement of RPs according to the location of group participants.
- It avoids inconsistent, manual RP configurations that might cause connectivity problems.

Multiple RPs can be used to serve different group ranges or to serve as hot backups for each other. To ensure that Auto-RP functions, configure routers as candidate RPs so that they can announce their interest in operating as an RP for certain group ranges. Additionally, a router must be designated as an RP-mapping agent that receives the RP-announcement messages from the candidate RPs, and arbitrates conflicts. The RP-mapping agent sends the consistent group-to-RP mappings to all remaining routers. Thus, all routers automatically determine which RP to use for the groups they support.

Tip By default, if a given group address is covered by group-to-RP mappings from both static RP configuration, and is discovered using Auto-RP or PIM BSR, the Auto-RP or PIM BSR range is preferred. To override the default, and use only the RP mapping, use the **rp-address override** keyword.

Note Auto-RP is not supported on VRF interfaces. Auto-RP Lite allows you to configure auto-RP on the CE router. It allows the PE router that has the VRF interface to relay auto-RP discovery, and announce messages across the core and eventually to the remote CE. Auto-RP is supported in only the IPv4 address family.

Configuring Example

```
Router#configure
Router(config)# router pim
Router(config-pim-ipv4)# auto-rp candidate-rp GigabitEthernet0/1/0/1 scope 31 group-list 2
 bidir
Router(config-pim-ipv4)# auto-rp mapping-agent GigabitEthernet0/1/0/1 scope 20
Router(config-pim-ipv4)# exit
Router(config)# ipv4 access-list 2
Router(config-ipv4-acl)# permit 239.1.1.1 0.0.0.0
Router(config-ipv4-acl)#commit
```

This example shows that Auto-RP messages are prevented from being sent out of the GigabitEthernet interface 0/3/0/0. It also shows that access list 111 is used by the Auto-RP candidate and access list 222 is used by the boundary command to contain traffic on GigabitEthernet interface 0/3/0/0.

```
ipv4 access-list 111
 10 permit 224.1.0.0 0.0.255.255
 20 permit 224.2.0.0 0.0.255.255
!
!Access list 111 is used by the Auto-RP candidate.
!
ipv4 access-list 222
 10 deny any host 224.0.1.39
 20 deny any host 224.0.1.40
!
!Access list 222 is used by the boundary command to contain traffic (on
GigabitEthernet0/3/0/0) that is sent to groups 224.0.1.39 and 224.0.1.40.
!
router pim
 auto-rp mapping-agent loopback 2 scope 32 interval 30
 auto-rp candidate-rp loopback 2 scope 15 group-list 111 interval 30
multicast-routing
 interface hundredGigE 0/0/0/25
 boundary 222
!
```

PIM Bootstrap Router

The PIM bootstrap router (BSR) provides a fault-tolerant, automated RP discovery and distribution mechanism.

To avoid a single point of failure, you can configure several candidate BSRs in a PIM domain. A BSR is elected among the candidate BSRs automatically.

Candidates use bootstrap messages to discover which BSR has the highest priority. The candidate with the highest priority sends an announcement to all PIM routers in the PIM domain that it is the BSR.

Routers that are configured as candidate RPs unicast to the BSR the group range for which they are responsible. The BSR includes this information in its bootstrap messages and disseminates it to all PIM routers in the domain. Based on this information, all routers are able to map multicast groups to specific RPs. As long as a router is receiving the bootstrap message, it has a current RP map.

Configuration Example

Configures the router as a candidate BSR with a hash mask length of 30:

```
Router# configure
Router:(config)# router pim
Router:(config-pim)# bsr candidate-bsr 10.0.0.1 hash-mask-len 30
Router:(config-ipv4-acl)#commit
```

Configures the router to advertise itself as a candidate rendezvous point to the BSR in its PIM domain. Access list number 4 specifies the prefix associated with the candidate rendezvous point address 10.2.1.1 . This rendezvous point is responsible for the groups with the prefix 239.

```
RP/0/RP0/CPU0:ios#configure
Thu Jan 30 08:03:47.952 UTC
RP/0/RP0/CPU0:ios(config)#router pim
RP/0/RP0/CPU0:ios(config-pim)#bsr candidate-bsr 10.0.0.1 hash-mask-len 30
RP/0/RP0/CPU0:ios(config-pim)#bsr candidate-rp 172.3.2.1 group-list 4 bidir
RP/0/RP0/CPU0:ios(config-pim)#interface fourHundredGigE 0/0/0/1
RP/0/RP0/CPU0:ios(config-pim-ipv4-if)# bsr-border
RP/0/RP0/CPU0:ios(config-pim-ipv4-if)#exit
RP/0/RP0/CPU0:ios(config-pim-default-ipv4)#exit
RP/0/RP0/CPU0:ios(config-pim)#exit
RP/0/RP0/CPU0:ios(config)#ipv4 access-list 4
RP/0/RP0/CPU0:ios(config-ipv4-acl)#permit 239.1.1.1 0.255.255.255
RP/0/RP0/CPU0:ios(config-ipv4-acl)#commit
Thu Jan 30 08:05:36.780 UTC
RP/0/RP0/CPU0:ios(config-ipv4-acl)#exit
RP/0/RP0/CPU0:ios(config)#exit
```

Running Configuration

```
RP/0/RP0/CPU0:ios#show running-config router pim
Thu Jan 30 08:08:06.568 UTC
router pim
 address-family ipv4
  interface FourHundredGigE0/0/0/1
   bsr-border
  !
  bsr candidate-bsr 10.0.0.1 hash-mask-len 30 priority 1
  bsr candidate-rp 172.3.2.1 group-list 4 priority 192 interval 60 bidir
 !
!
```

Verification

Displays PIM candidate RP information for the BSR.

```
RP/0/RP0/CPU0:ios#show pim bsr candidate-rp
Thu Jan 30 08:08:32.851 UTC
PIM BSR Candidate RP Info

Cand-RP          mode  scope priority uptime    group-list
172.3.2.1        BD    16    192      00:00:00  4
```

Displays PIM candidate election information for the BSR.

```
RP/0/RP0/CPU0:ios#show pim bsr election
Thu Jan 30 08:08:58.846 UTC
PIM BSR Election State

Cand/Elect-State        Uptime   BS-Timer BSR                C-BSR

Inactive/Accept-Any     00:00:00 00:00:00 0.0.0.0 [0, 0]     10.0.0.1 [1, 30]
```

Displays PIM RP cache information for the BSR.

```
RP/0/RP0/CPU0:ios#show pim bsr rp-cache
Thu Jan 30 08:09:44.901 UTC
PIM BSR Candidate RP Cache
```

Displays group-to-PIM mode mapping.

```
RP/0/RP0/CPU0:ios#show pim ipv4 group-map
Thu Jan 30 08:10:14.793 UTC
No ranges found.
```

Reverse-Path Forwarding

Reverse-path forwarding (RPF) is an algorithm used for forwarding multicast datagrams. It functions as follows:

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

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

- If a PIM router has an (S,G) entry present in the multicast routing table (a source-tree state), the router performs the RPF check against the IP address of the source for the multicast packet.
- If a PIM router has no explicit source-tree state, this is considered a shared-tree state. The router performs the RPF check on the address of the RP, which is known when members join the group.

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

Multicast Non-Stop Routing

Multicast Non-Stop Routing (NSR) enables the router to synchronize the multicast routing tables on both the active and standby RSPs so that during an HA scenario like an RSP failover there is no loss of multicast data. Multicast NSR is enabled through the multicast processes being hot standby. Multicast NSR supports both Zero Packet Loss (ZPL) and Zero Topology Loss (ZTL). With Multicast NSR, there is less CPU churn and no multicast session flaps during a failover event.

Multicast NSR is enabled by default, however, if any unsupported features like BNG or Snooping are configured, Multicast performs Non-Stop Forwarding (NSF) functionality during failover events. When Multicast NSR is enabled, multicast routing state is synchronized between the active and standby RSPs. Once the synchronization occurs, each of the multicast processes signal the NSR readiness to the system. For the multicast processes to support NSR, the processes must be hot standby compliant. That is, the processes on active and standby RSPs both have to be in synchronization at all times. The active RSP receives packets from the network and makes local decisions while the standby receives packet from the network and synchronizes it with the active RSPs for all the local decisions. Once the state is determined, a check is performed to verify if the states are synchronized. If the states are synchronized, a signal in the form NSR_READY is conveyed to the NSR system.

With NSR, in the case of a failover event, routing changes are updated to the forwarding plane immediately. With NSF, there is an NSF hold time delay before routing changes can be updated.

Non-Supported Features

The following features are unsupported on NG NSR:

- IGMP and MLD Snooping
- BNG

Configuration Example

```
RP/0/RP0/CPU0:ios#configure
Fri Feb  7 08:53:51.603 UTC
RP/0/RP0/CPU0:ios(config)#router pim address-family ipv4
RP/0/RP0/CPU0:ios(config-pim-default-ipv4)#nsf lifetime 30
RP/0/RP0/CPU0:ios(config-pim-default-ipv4)#exit
RP/0/RP0/CPU0:ios(config-pim)#router igmp
RP/0/RP0/CPU0:ios(config-igmp)#nsf lifetime 30
RP/0/RP0/CPU0:ios(config-igmp)#commit
Fri Feb  7 08:54:45.747 UTC
RP/0/RP0/CPU0:ios(config-igmp)#exit
RP/0/RP0/CPU0:ios(config)#exit
RP/0/RP0/CPU0:ios#show igmp nsf
Fri Feb  7 08:55:02.046 UTC
IGMP Non-Stop Forwarding Status:
Multicast routing state: Normal
   NSF Lifetime:       00:00:30

RP/0/RP0/CPU0:ios#show mfib nsf
Fri Feb  7 08:55:12.462 UTC
IP MFWD Non-Stop Forwarding Status:
 NSF Lifetime:          00:15:00

On node 0/RP0/CPU0 :
Multicast routing state: Normal

RP/0/RP0/CPU0:ios#show mrib nsf
Fri Feb  7 08:55:24.228 UTC
IP MRIB Non-Stop Forwarding Status:
Multicast routing state: Normal
   NSF Lifetime:       00:01:30
RP/0/RP0/CPU0:ios#show pim nsf
Fri Feb  7 08:55:33.499 UTC
IP PIM Non-Stop Forwarding Status:
Multicast routing state: Normal
   NSF Lifetime:       00:00:30
RP/0/RP0/CPU0:ios#
```

Verification

Verify the state of NSF operation in IGMP.

```
RP/0/RP0/CPU0:ios#show igmp nsf
Fri Feb  7 08:55:02.046 UTC
IGMP Non-Stop Forwarding Status:
Multicast routing state: Normal
   NSF Lifetime:       00:00:30
```

Verify the state of NSF operation for the MFIB line cards.

```
RP/0/RP0/CPU0:ios#show mfib nsf
Fri Feb  7 08:55:12.462 UTC
IP MFWD Non-Stop Forwarding Status:
 NSF Lifetime:          00:15:00

On node 0/RP0/CPU0 :
Multicast routing state: Normal
```

Verify the state of NSF operation in the MRIB.

```
RP/0/RP0/CPU0:ios#show mrib nsf
Fri Feb  7 08:55:24.228 UTC
IP MRIB Non-Stop Forwarding Status:
Multicast routing state: Normal
   NSF Lifetime:       00:01:30
```

Verify the state of NSF operation for PIM.

```
RP/0/RP0/CPU0:ios#show pim nsf
Fri Feb  7 08:55:33.499 UTC
IP PIM Non-Stop Forwarding Status:
Multicast routing state: Normal
   NSF Lifetime:       00:00:30
RP/0/RP0/CPU0:ios#
```

Failure Scenarios in NSR

If a switchover occurs before all multicast processes issue an NSR_READY signal, the proceedings revert back to the existing NSF behavior. Also, on receiving the GO_ACTIVE signal from the multicast processes, the following events occur in processes that have not signaled NSR_READY:

1. IGMP starts the NSF timer for one minute.
2. PIM starts the NSF timer for two minutes.
3. MSDP resets all peer sessions that are not synchronized.

Multicast-Only Fast Reroute

Multicast-Only Fast Reroute (MoFRR) allows fast reroute for multicast traffic on a multicast router. MoFRR minimizes packet loss in a network when node or link failures occur (at the topology merge point). It works by making simple enhancements to multicast routing protocols.

MoFRR involves transmitting a multicast join message from a receiver towards a source on a primary path and transmitting a secondary multicast join message from the receiver towards the source on a backup path.

Data packets are received from the primary and secondary paths. The redundant packets are discarded at topology merge points with the help of Reverse Path Forwarding (RPF) checks. When a failure is detected on the primary path, the repair occurs locally by changing the interface on which packets are accepted to the secondary interface, thus improving the convergence times in the event of a node or link failure on the primary path.

RIB-Based MoFRR

Table 12: Feature History Table

Feature Name	Release Information	
RIB-Based MoFRR	Release 24.4.1	Introduced in this release on: Fixed Systems (8700) (select variants only*) RIB-Based MoFRR enhances network reliability by allowing multicast traffic to be rerouted through alternate paths in case of a primary path failure. This feature significantly reduces traffic disruption and improves fault tolerance by quickly switching to pre-calculated backup routes. It leverages the Routing Information Base (RIB) to maintain efficient multicast forwarding, ensuring uninterrupted service delivery. By integrating with existing routing protocols, RIB-Based MoFRR provides a seamless and resilient multicast routing solution, optimizing network performance and maintaining high availability without requiring extensive configuration changes. *This functionality is now supported on Cisco 8712-MOD-M routers.

RIB-based MoFRR enables PIM to perform a fast convergence of specified routes or flows upon detecting a failure on any of the multiple equal-cost paths between the router and the source.

Configuring RIB-Based MoFRR

When a failure is detected on one of multiple equal-cost paths between the router and the source, perform a fast convergence (MoFRR) of specified routes or flows using the **mofrr rib** command.

Configuration example

```
Router(config)# router pim
Router(pim)# mofrr rib route-list
```

Running Configuration

```
Router#show running-config router pim
router pim
 address-family ipv4
  mofrr
   rib route-list
  !
 !
!
```

Note The *route-list* keyword is a previously defined IPv4 access-list which has the source and group information to match the specific multicast route that needs MoFRR enabled. Please refer to *Modular QoS Configuration Guide* on how to configure an access-list.

Verify RIB-Based MoFRR is Enabled

Verify that you have successfully configured RIB-Based MoFRR using the following CLI command. The command output shows the MoFRR-RIB flag, the primary and secondary RPF interfaces are enabled for MoFRR.

Sample Configuration:

```
Router# show pim topology src-ip-address/ grp-address detail
```

Verification Example:

```
Router# show pim topology 232.1.1.1 detail

IP PIM Multicast Topology Table

(100.1.1.1,232.1.1.1)SPT SSM Up: 00:00:35
JP: Join(00:00:14) RPF: FourHundredGigE0/0/0/2,100.1.1.1* MoFRR-RIB, Flags:
Up: MT clr (00:00:00) MDT: JoinSend N, Cache N/N/N, Misc (0x0,0/0)
Cache: Add 00:00:00, Rem 00:00:00. MT Cnt: Set 0, Unset 0. Joins sent 0
MDT-ifh 0x0/0x0, MT Slot none/ none
RPF-redirect BW usage: 0, Flags: 0x0, ObjID: 0x0
c-multicast-routing: PIM* BGPJP: 1w0d
RPF Table: IPv4-Unicast-default
RPF Secondary: FourHundredGigE0/0/0/1,100.1.1.1
  FourHundredGigE0/0/0/0      00:00:35  fwd LI LH
```

Note To ensure that MoFRR yields better convergence, prioritize the multicast source routes using IGP protocol for RPF check. Thus ensuring the routes are always taken first for SPF calculation in case of path changes.

```
Router(config)# router isis isp
Router(config-isis)#)#address-family ipv4 unicast
Router(config-isis-af)#spf prefix-priority critical ISIS-CRIT
```

```
Router#show running-config ipv4 prefix-list ISIS-CRIT
Wed May 27 01:26:58.653 PDT
ipv4 prefix-list ISIS-CRIT
10 permit 192.0.2.1/32 ge 32
11 permit 192.0.2.252/32 ge 32
```

Protection-based MoFRR

Table 13: Feature History Table

Feature Name	Release Information	Feature Description
Protection-based MoFRR	Release 24.4.1	Introduced in this release on: Fixed Systems (8200, 8700); Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This functionality is now supported on Cisco 8712-MOD-M routers.
Protection-based MoFRR	Release 24.2.11	We have made fault detection and convergence faster for multicast routes, ensuring multicast data, such as IPTV feeds, is delivered with minimum interruptions. This is made possible because we enable the use of a Protection Global Identifier (GID) for Multicast-Only Fast Reroute (MoFRR), which allows the router to quickly identify and switch to a backup or secondary path when a failure is detected on the primary path. This feature introduces the following changes: **CLI**: • The **protect** keyword is introduced in the **mofrr** command. **YANG Data Model:** • New XPaths for `Cisco-IOS-XR-ipv4-pim-cfg.yang` (see GitHub, YANG Data Models Navigator)

Protection GID for Faster Multicast Route Convergence

The Protection-based MoFRR feature ensures quicker route convergence than RIB-Based MoFRR, which is particularly beneficial for network service providers offering residential triple-play services that simultaneously contain voice, data, and video applications. Using enhanced Multicast Forwarding Information Base (MFIB)

programming, this feature optimizes fault detection and route convergence by creating a unique GID entry for each multicast source and its associated primary and secondary Reverse Path Forwarding (RPF) interfaces.

A single Protection GID can be associated with multiple multicast routes (S,G), but it's unique to each multicast source address (S) and the corresponding primary-secondary RPF interfaces. In the event of Equal-Cost Multi-Path (ECMP) failures, updating the Protection GID alone suffices to switch traffic to the backup or secondary RPF interface. As a result, Protection-Based MoFRR facilitates more rapid convergence than its RIB-based counterpart.

The table, "Sample Protection GID Database per Multicast Source" shows an example of a Protection GID, such as 0x1, with a specific multicast source (S1), its primary RPF interface (Int1) and secondary interface (Int2), demonstrating the rapid identification and switching mechanism to a backup route in the event of a primary path failure. For example, switching from Int2 to Int1 for the multicast route (S1, G6) with Protection GID, 0x4.

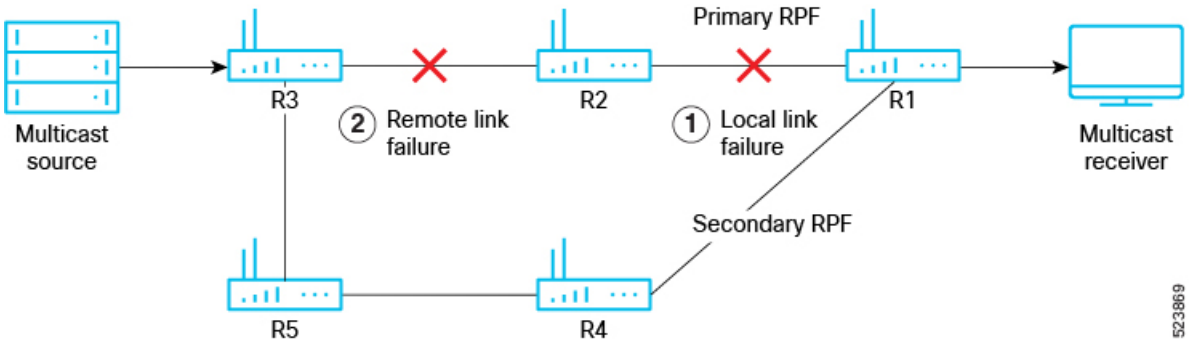
The values and combinations used in the table are for representation purpose only.

Table 14: Sample Protection GID Database per Multicast Source

Multicast Route (S, G)	Protection GID	Multicast Source	Primary RPF Interface	Secondary RPF Interface
(S1, G1)	0x1	S1	Int1	Int2
(S1, G2)	0x1	S1	Int1	Int2
(S2, G11)	0x2	S2	Int1	Int2
(S2, G12)	0x2	S2	Int1	Int2
(S1, G5)	0x3	S1	Int2	Int3
(S1, G6)	0x4	S1	Int2	Int1

Protection-based MoFRR for Local and Remote Link Failures

Figure 5: Local and Remote Link Failures Topology



As shown in the figure, "Local and Remote Link Failures Topology", with protection based MoFRR, different convergence times are possible depending on the type of upstream path failure:

- Local link failure — occurs on a link directly connected to the router that is performing the fast reroute action. For instance, this could be the failure of an interface on the router (R1) or the failure of a link (R1<—>R2) where one end is directly connected to the router.
- Remote link failure — occurs on a part of the network that is not directly connected to the router performing the fast reroute action. The failure is not immediately detectable by the router's direct interfaces, as it happens further upstream or downstream (R3<—>R2<—>R1)in the network.

To enhance fault detection and reduce convergence time for multicast routes, it is possible to configure the feature to give priority to local link failures rather than remote link failures. For more details, see Configure Protection-Based MoFRR for Local Link Failure, on page 68.

Prerequisites for Protection-Based MoFRR

The tasks in this module assume that IP multicasting has been enabled and that PIM interfaces have been configured.

Limitations and Usage Guidelines for Protection-Based MoFRR

Supported Protocols

The Protection-Based MoFRR feature supports the following protocols:

- Interior Gateway Protocol (IGP)
- Intermediate System to Intermediate System (ISIS)
- Protocol Independent Multicast (PIM) for IPv4 native multicast
- Multiprotocol Label Switching (MPLS) and Label Distribution Protocol (LDP)
- Segment Routing (SR)
- Segment Routing Traffic Engineering (SR TE) Autoroute announce policies

Multicast and Unicast Support

For multicast and unicast data traffic, the following limitations and guidelines apply to the Protection-Based MoFRR feature:

- Supports only native multicast.
- Supports only IPv4 multicast, not IPv6 multicast.
- Supports only IP multicast, not labelled multicast, such as Multicast Label Distribution Protocol (MLDP).
- Does not support non-congruent topologies for unicast and multicast, for example, multicast-intact in IGP.

Local and Remote Link Failure Support

Configuring the **local-fault-only** option does not optimize the router for remote-fault MoFRR convergences.

Configure and Verify Protection-Based MoFRR

This section provides details on how to configure the Protection-Based MoFRR feature in various scenarios and to verify if the feature is enabled.

Before you Begin

The tasks in this module assume that IP multicasting has been enabled and that PIM interfaces have been configured.

Configure Protection-Based MoFRR

To perform faster route convergence during any link or node failures, configure the Protection-Based MoFRR feature using the following CLI command:

```
Router#configure
Router(config)#router pim
Router(pim)#mofrr protect route-list
```

Running Configuration

```
router pim
 address-family ipv4
  mofrr
   protect route-list
  !
 !
!
```

Configure Protection-Based MoFRR for Local Link Failure

Remote link failures require longer convergence times compared to local link failures. To prioritize local link failures and improve route convergence time, configure the local-fault-only option using the following CLI command.

```
Router#configure
Router(config)#router pim
Router(pim)#mofrr protect route-list local-fault-only
```

Running Configuration

```
router pim
 address-family ipv4
  mofrr
   protect route-list local-fault-only
  !
 !
!
```

Verify Protection-Based MoFRR is Enabled

Use the examples and sample CLI commands to verify you have successfully configured Protection-Based MoFRR.

- Verify the multicast source and group addresses are configured for MoFRR.

  Sample Configuration:

  Router# **show pim topology** *src-ip-address/ grp-address* **detail**

Verification Example:

```
Router# show pim topology 224.1.1.1 detail

(192.0.2.4,224.1.1.1)SPT SM Up: 00:00:50
JP: Join(00:00:43) RPF: FourHundredGigE0/0/0/5,192.0.2.2 MoFRR, Flags:
Up: MT clr (00:00:00) MDT: JoinSend N, Cache N/N/N, Misc (0x0,0/0)
Cache: Add 00:00:00, Rem 00:00:00. MT Cnt: Set 0, Unset 0. Joins sent 0
MDT-ifh 0x0/0x0, MT Slot none/ none
RPF-redirect BW usage: 0, Flags: 0x0, ObjID: 0x0
c-multicast-routing: PIM BGPJP: 01:18:47
RPF Table: IPv4-Unicast-default
RPF Secondary: FourHundredGigE0/0/0/3,192.0.2.3
  FourHundredGigE0/0/0/9      00:00:50  fwd Join(00:02:39) L
```

- Verify the primary and secondary RPF interfaces are configured and enabled for MoFRR.

  Sample Configuration:

  ```
  Router# show mrib route src-ip-address/ grp-address
  ```

  Verification Example:

  ```
  Router# show mrib route 224.1.1.1

  (192.0.2.4,224.1.1.1) RPF nbr: 192.0.2.2 Flags: RPF MoFE MoFS
    Up: 00:06:27
    MOFRR State: Inactive Sequence No 1
    Incoming Interface List
      FourHundredGigE0/0/0/3 Flags: A2, Up: 00:05:43
      FourHundredGigE0/0/0/5 Flags: A, Up: 00:06:27
    Outgoing Interface List
      FourHundredGigE0/0/0/9 Flags: F NS LI, Up: 00:06:27
  ```

- Verify the protection GID is enabled for MoFRR.

  Sample Configuration:

  ```
  Router# show mfib route src-ip-address/ grp-address
  ```

  Verification Example:

  ```
  Router# show mfib route 224.1.1.1

  (192.0.2.4,224.1.1.1),  Flags:  MoFE MoFS
    Up: 00:02:01
    Last Used: never
    SW Forwarding Counts: 0/0/0
    SW Replication Counts: 0/0/0
    SW Failure Counts: 0/0/0/0/0
    GID ENTRY: 0x30a6662d18
    FourHundredGigE0/0/0/3 Flags:  A2, Up:00:01:16
    FourHundredGigE0/0/0/5 Flags:  A, Up:00:02:01
    FourHundredGigE0/0/0/9 Flags:  NS EG, Up:00:02:01
  ```

- Verify the protection GID is enabled in MIFB for MoFRR.

  Sample Configuration:

  ```
  Router# show mfib route src-ip-address/ grp-address
  ```

  Verification Example:

```
Router# show mfib mofrr-protection-gid
```

```
                    MoFRR Protection GID Entry Database
                   =====================================

      GID-ENTRY    Table-ID   Primary-IFH  Secondary-IFH   FRR Active  Source     Retry
-------------------------------------------------------------------------------------------------
0X30A6662D18 0xe0000000 0XF0001B8   0XF0001A8       FALSE      192.0.2.4   FALSE
```

Multicast Source Discovery Protocol

Multicast Source Discovery Protocol (MSDP) is a mechanism to connect multiple PIM sparse-mode domains. MSDP allows multicast sources for a group to be known to all rendezvous points (RPs) in different domains. Each PIM-SM domain uses its own RPs and need not depend on RPs in other domains.

An RP in a PIM-SM domain has MSDP peering relationships with MSDP-enabled routers in other domains. Each peering relationship occurs over a TCP connection, which is maintained by the underlying routing system.

MSDP speakers exchange messages called Source Active (SA) messages. When an RP learns about a local active source, typically through a PIM register message, the MSDP process encapsulates the register in an SA message and forwards the information to its peers. The message contains the source and group information for the multicast flow, as well as any encapsulated data. If a neighboring RP has local joiners for the multicast group, the RP installs the S, G route, forwards the encapsulated data contained in the SA message, and sends PIM joins back towards the source. This process describes how a multicast path can be built between domains.

Note Although you should configure BGP or Multiprotocol BGP for optimal MSDP interdomain operation, this is not considered necessary in the Cisco IOS XR Software implementation. For information about how BGP or Multiprotocol BGP may be used with MSDP, see the MSDP RPF rules listed in the Multicast Source Discovery Protocol (MSDP), Internet Engineering Task Force (IETF) Internet draft.

Restriction

Loop-Free Alternative Fast Reroute is not supported.

MSDP Configuration Submode

When you issue the **router msdp** command, the CLI prompt changes to “config-msdp,” indicating that you have entered router MSDP configuration submode.

Multicast Nonstop Forwarding

The Cisco IOS XR Software nonstop forwarding (NSF) feature for multicast enhances high availability (HA) of multicast packet forwarding. NSF prevents hardware or software failures on the control plane from disrupting the forwarding of existing packet flows through the router.

The contents of the Multicast Forwarding Information Base (MFIB) are frozen during a control plane failure. Subsequently, PIM attempts to recover normal protocol processing and state before the neighboring routers time out the PIM hello neighbor adjacency for the problematic router. This behavior prevents the NSF-capable router from being transferred to neighbors that will otherwise detect the failure through the timed-out adjacency. Routes in MFIB are marked as stale after entering NSF, and traffic continues to be forwarded (based on those

routes) until NSF completion. On completion, MRIB notifies MFIB and MFIB performs a mark-and-sweep to synchronize MFIB with the current MRIB route information.

Multicast Configuration Submodes

Cisco IOS XR Software moves control plane CLI configurations to protocol-specific submodes to provide mechanisms for enabling, disabling, and configuring multicast features on a large number of interfaces.

Cisco IOS XR Software allows you to issue most commands available under submodes as one single command string from the global or XR config mode.

For example, the **ssm** command could be executed from the PIM configuration submode like this:

```
RP/0/RSP0/CPU0:router(config)# router pim
RP/0/RSP0/CPU0:router(config-pim)# address-family ipv4
RP/0/RSP0/CPU0:router(config-pim-default-ipv4)# ssm range
```

Alternatively, you could issue the same command from the global or XR config mode like this:

```
RP/0/RSP0/CPU0:router(config)# router pim ssm range
```

The following multicast protocol-specific submodes are available through these configuration submodes:

Multicast-Routing Configuration Submode

Basic multicast services start automatically without any explicit configuration required. The following multicast services are started automatically:

- MFWD
- MRIB
- PIM
- IGMP

Other multicast services require explicit configuration before they start. For example, to start the MSDP process, you must enter the **router msdp** command and explicitly configure it.

When you issue the **multicast-routing ipv4 or multicast-routing ipv6** command, all default multicast components (PIM, IGMP, MLD, MFWD, and MRIB) are automatically started, and the CLI prompt changes to “config-mcast-ipv4” or “config-mcast-ipv6”, indicating that you have entered multicast-routing configuration submode.

PIM Configuration Submode

When you issue the **router pim** command, the CLI prompt changes to “config-pim-ipv4,” indicating that you have entered the default pim address-family configuration submode.

To enter pim address-family configuration submode for IPv6, type the **address-family ipv6** keyword together with the **router pim** command before pressing Enter.

IGMP Configuration Submode

When you issue the **router igmp** command, the CLI prompt changes to "config-igmp," indicating that you have entered IGMP configuration submode.

MLD Configuration Submode

When you issue the **router mld** command, the CLI prompt changes to "config-mld," indicating that you have entered MLD configuration submode.

MSDP Configuration Submode

When you issue the **router msdp** command, the CLI prompt changes to "config-msdp," indicating that you have entered router MSDP configuration submode.

Understanding Interface Configuration Inheritance

Cisco IOS XR Software allows you to configure commands for a large number of interfaces by applying command configuration within a multicast routing submode that could be inherited by all interfaces. To override the inheritance mechanism, you can enter interface configuration submode and explicitly enter a different command parameter.

For example, in the following configuration you could quickly specify (under router PIM configuration mode) that all existing and new PIM interfaces on your router will use the hello interval parameter of 420 seconds. However, Packet-over-SONET/SDH (POS) interface 0/1/0/1 overrides the global interface configuration and uses the hello interval time of 210 seconds.

```
RP/0/RP0/CPU0:router(config)# router pim
RP/0/RP0/CPU0:router(config-pim-default-ipv4)# hello-interval 420
RP/0/RP0/CPU0:router(config-pim-default-ipv4)# interface pos 0/1/0/1
RP/0/RP0/CPU0:router(config-pim-ipv4-if)# hello-interval 210
```

The following is a listing of commands (specified under the appropriate router submode) that use the inheritance mechanism:

```
router pim
  dr-priority
  hello-interval
  join-prune-interval

multicast-routing
  version
  query-interval
  query-max-response-time
  explicit-tracking
router mld
  interface all disable
  version
  query-interval
  query-max-response-time
  explicit-tracking
```

```
router msdp
  connect-source
  sa-filter
  filter-sa-request list
  remote-as
  ttl-threshold
```

Understanding Interface Configuration Inheritance Disablement

As stated elsewhere, Cisco IOS XR Software allows you to configure multiple interfaces by applying configurations within a multicast routing submode that can be inherited by all interfaces.

To override the inheritance feature on specific interfaces or on all interfaces, you can enter the address-family IPv4 or IPv6 submode of multicast routing configuration mode, and enter the **interface-inheritance disable** command together with the **interface** *type interface-path-id* or **interface all** command. This causes PIM or IGMP protocols to disallow multicast routing and to allow only multicast forwarding on those interfaces specified. However, routing can still be explicitly enabled on specified individual interfaces.

The following configuration disables multicast routing interface inheritance under PIM and IGMP generally, although forwarding enablement continues. The example shows interface enablement under IGMP of GigabitEthernet 0/6/0/3:

```
RP/0/RP0/CPU0:router# multicast-routing address-family ipv4
RP/0/RP0/CPU0:router(config-mcast-default-ipv4)# interface all enable
RP/0/RP0/CPU0:router(config-mcast-default-ipv4)# interface-inheritance disable

!

!
RP/0/RP0/CPU0:router(config)# router igmp
RP/0/RP0/CPU0:router(config-igmp)# vrf default
RP/0/RP0/CPU0:router(config-igmp)# interface GigabitEthernet0/6/0/0
RP/0/RP0/CPU0:router(config-igmp-name-if)# router enable
```

For related information, see Understanding Enabling and Disabling Interfaces, on page 73.

Understanding Enabling and Disabling Interfaces

When the Cisco IOS XR Software multicast routing feature is configured on your router, by default, no interfaces are enabled.

To enable multicast routing and protocols on a single interface or multiple interfaces, you must explicitly enable interfaces using the **interface** command in multicast routing configuration mode.

To set up multicast routing on all interfaces, enter the **interface all** command in multicast routing configuration mode. For any interface to be fully enabled for multicast routing, it must be enabled specifically (or be default) in multicast routing configuration mode, and it must not be disabled in the PIM and IGMP/MLD configuration modes.

For example, in the following configuration, all interfaces are explicitly configured from multicast routing configuration submode:

```
RP/0/RP0/CPU0:router(config)# multicast-routing
RP/0/RP0/CPU0:router(config-mcast)# interface all enable
```

To disable an interface that was globally configured from the multicast routing configuration submode, enter interface configuration submode, as illustrated in the following example:

```
RP/0/RP0/CPU0:router(config-mcast)# interface GigabitEthernet0pos 0/1/0/0
RP/0/RP0/CPU0:router(config-mcast-default-ipv4-if)# disable
```

Controlling Source Information on MSDP Peer Routers

Your MSDP peer router can be customized to control source information that is originated, forwarded, received, cached, and encapsulated.

When originating Source-Active (SA) messages, you can control to whom you will originate source information, based on the source that is requesting information.

When forwarding SA messages you can do the following:

- Filter all source/group pairs
- Specify an extended access list to pass only certain source/group pairs
- Filter based on match criteria in a route map

When receiving SA messages you can do the following:

- Filter all incoming SA messages from an MSDP peer
- Specify an extended access list to pass certain source/group pairs
- Filter based on match criteria in a route map

In addition, you can use time to live (TTL) to control what data is encapsulated in the first SA message for every source. For example, you could limit internal traffic to a TTL of eight hops. If you want other groups to go to external locations, you send those packets with a TTL greater than eight hops.

By default, MSDP automatically sends SA messages to peers when a new member joins a group and wants to receive multicast traffic. You are no longer required to configure an SA request to a specified MSDP peer.

Configuration Example

```
Router#configure
Router(config)# router msdp
Router(config-msdp)# sa-filter out router.cisco.com list 100
Router(config-msdp)# cache-sa-state 100
Router(config-msdp)# ttl-threshold 8
Router(config-msdp)# exit
Router(config)# ipv4 access-list 100 20 permit 239.1.1.1 0.0.0.0
Router(config)# commit
```

Multicast Routing Information Base

The Multicast Routing Information Base (MRIB) is a protocol-independent multicast routing table that describes a logical network in which one or more multicast routing protocols are running. The tables contain generic multicast routes installed by individual multicast routing protocols. There is an MRIB for every logical network (VPN) in which the router is configured. MRIBs do not redistribute routes among multicast routing protocols; they select the preferred multicast route from comparable ones, and they notify their clients of changes in selected attributes of any multicast route.

Multicast Forwarding Information Base

Table 15: Feature History Table

Feature Name	Release Information	Feature Description
MFIB scale enhancement	Release 7.3.15	This feature allows you to increase the route-scale for IPv4 SSM from 64K to 120K using the hw-module multicast route-scale command.

Multicast Forwarding Information Base (MFIB) is a protocol-independent multicast forwarding system that contains unique multicast forwarding entries for each source or group pair known in a given network. There is a separate MFIB for every logical network (VPN) in which the router is configured. Each MFIB entry resolves a given source or group pair to an incoming interface (IIF) for reverse-path forwarding (RPF) checking and an outgoing interface list (olist) for multicast forwarding.

Restrictions

- The **hw-module multicast route-scale** command is supported only on modular systems and not on fixed/centralized systems.

Enable 120K Route-Scale for IPv4 SSM

Use the **hw-module multicast route-scale** command to enable the 120K route-scale for IPv4 SSM. Note that IPv6 supports only 64K route-scale.

```
Router# configure
Router(config)# hw-module multicast route-scale
```

See **hw-module multicast route-scale** command under the *Multicast Routing Forwarding Commands* chapter in *Multicast Command Reference for Cisco 8000 Series Routers*.

Note For the new route-scale to take effect, you must reload all the nodes on your router using the **reload** command.

```
Router# reload location all
```

MSDP MD5 Password Authentication

MSDP MD5 password authentication is an enhancement to support Message Digest 5 (MD5) signature protection on a TCP connection between two Multicast Source Discovery Protocol (MSDP) peers. This feature provides added security by protecting MSDP against the threat of spoofed TCP segments being introduced into the TCP connection stream.

MSDP MD5 password authentication verifies each segment sent on the TCP connection between MSDP peers. The **password clear** command is used to enable MD5 authentication for TCP connections between two MSDP peers. When MD5 authentication is enabled between two MSDP peers, each segment sent on the TCP connection between the peers is verified.

Note MSDP MD5 authentication must be configured with the same password on both MSDP peers to enable the connection between them. The 'password encrypted' command is used only for applying the stored running configuration. Once you configure the MSDP MD5 authentication, you can restore the configuration using this command.

MSDP MD5 password authentication uses an industry-standard MD5 algorithm for improved reliability and security.

Configuration Example

```
Router#configure
Router(config)#router msdp
Router(config-msdp)#peer 10.0.5.4
Router(config-msdp-peer)#password encrypted a34bi5m
Router(config-msdp-peer)#commit
```

Label Switch Multicast

Label Switch Multicast (LSM) is MPLS technology extensions to support multicast using label encapsulation. Next-generation MVPN is based on Multicast Label Distribution Protocol (mLDP), which can be used to build P2MP and MP2MP LSPs through a MPLS network. These LSPs can be used for transporting both IPv4 and IPv6 multicast packets, either in the global table or VPN context.

Benefits of LSM MLDP based MVPN

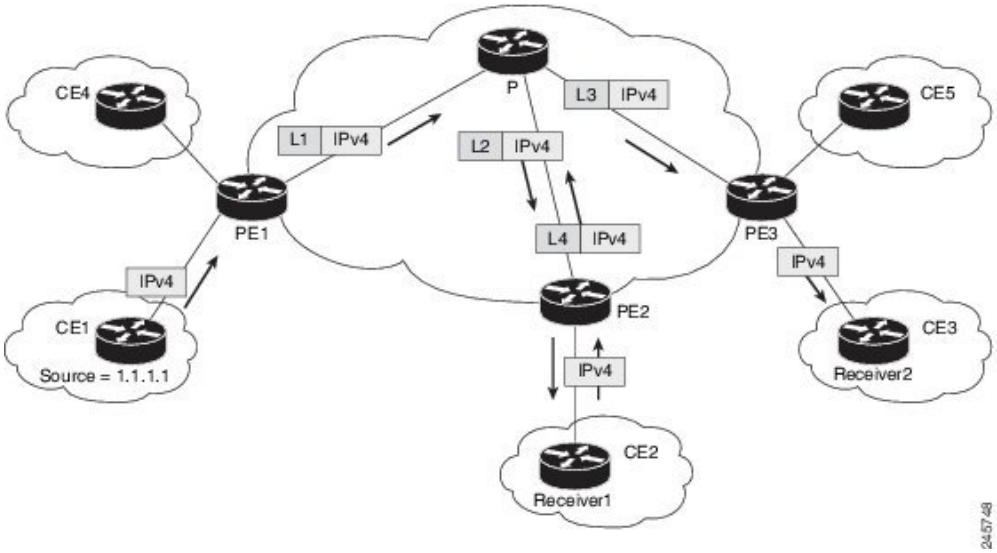
LSM provides these benefits when compared to GRE core tunnels that are currently used to transport customer traffic in the core:

- It leverages the MPLS infrastructure for transporting IP multicast packets, providing a common data plane for unicast and multicast.
- It eliminates the complexity associated PIM.

Configuring MLDP MVPN

The MLDP MVPN configuration enables IPv4 and IPV6 multicast packet delivery using MPLS. This configuration uses MPLS labels to construct default and data Multicast Distribution Trees (MDTs). The MPLS replication is used as a forwarding mechanism in the core and edge network. For MLDP MVPN configuration to work, ensure that the global MPLS MLDP configuration is enabled. To configure MVPN extranet support, configure the source multicast VPN Routing and Forwarding (mVRF) on the receiver Provider Edge (PE) router or configure the receiver mVRF on the source PE. MLDP MVPN is supported for both intranet and extranet.

Figure 6: MLDP based MPLS Network on Core Routers



Packet Flow in mLDP-based Multicast VPN

For each packet coming in, MPLS creates multiple out-labels. Packets from the source network are replicated along the path to the receiver network. The CE1 router sends out the native IP multicast traffic. The Provider Edge1 (PE1) router imposes a label on the incoming multicast packet and replicates the labeled packet towards the MPLS core network. When the packet reaches the core router (P), the packet is replicated with the appropriate labels for the MP2MP default MDT or the P2MP data MDT and transported to all the egress PEs. Once the packet reaches the egress PE (edge routers), the label is removed and the IP multicast packet is replicated onto the VRF interface. Basically, the packets are encapsulated at headend and decapsulated at tailend on the PE routers.

Multicast Label Distribution Protocol (MLDP) as Core Router

Multicast Label Distribution Protocol (MLDP) provides extensions to the Label Distribution Protocol (LDP) for the setup of point-to-multipoint (P2MP) and multipoint-to-multipoint (MP2MP) Label Switched Paths (LSPs) in Multiprotocol Label Switching (MPLS) networks.

MLDP eliminates the use of native multicast PIM to transport multicast packets across the core. In MLDP multicast traffic is label switched across the core. This saves a lot of control plane processing effort.

Configuration

For more information about MLDP configuration, see the *Enabling MLDP* section in the *Implementing MPLS Label Distribution Protocol* chapter of the *MPLS Configuration Guide for Cisco 8000 Routers.*

Point-to-Multipoint Traffic Engineering Label-Switched Multicast

IP multicast was traditionally used for IPTV broadcasting and content delivery services. Point-to-Multipoint (P2MP) Traffic-Engineering is fast replacing the IP multicast technique because of the various advantages of MPLS-TE, such as:

- Fast re-routing (FRR) and restoration in case of link/ node failure
- Bandwidth guarantee

Configuration

For more information about Point-to-Multipoint Traffic Engineering Label-Switched Multicast configuration, see the *Point-to-Multipoint Traffic-Engineering* section in the *Implementing MPLS Traffic Engineering* chapter of the *MPLS Configuration Guide for Cisco 8000 Routers.*

Label switched multicast (LSM) multicast label distribution protocol (mLDP) based multicast VPN (mVPN) support

Table 16: Feature History Table

Feature Name	Release Information	Feature Description
LSM mLDP based MVPN bud or tail node enhancements on edge routers	Release 24.4.1	Introduced in this release on: Fixed Systems(8200, 8700); Centralized Systems (8600); Modular Systems (8800 [LC ASIC: Q200, P100]). This feature extends the support for the rendezvous point (RP) placement on the LSM mLDP based mVPN bud or tail node on edge routers. With this feature, the BUD node is now supported on these profiles: 0, 1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 14, 15, 17, 19, 23, 25, 27, 28, and 29. Previously, the BUD node was supported only on Profiles 21 and 22.

Feature Name	Release Information	Feature Description
LSM mLDP based MVPN support on edge routers	Release 7.8.1	Label Switch Multicast (LSM) is MPLS technology extensions to support multicast using label encapsulation. Next-generation MVPN is based on Multipoint Label Distribution Protocol (mLDP), which can be used to build P2MP and MP2MP LSPs through a MPLS network. These LSPs can be used for transporting both IPv4 and IPv6 multicast packets, either in the global table or VPN context. From this release, mLDP is supported on edge routers on profiles 1, 2, 4, 5, 6, 7, 9, 12, 13, 14, 15, 17, 19, 21, 23, 25, 27, 28, and 29.

Label Switch Multicast (LSM) is a MPLS technology extension to support multicast using label encapsulation. Next-generation MVPN is based on Multicast Label Distribution Protocol (mLDP), which can be used to build P2MP and MP2MP LSPs through a MPLS network. These LSPs can be used for transporting both IPv4 and IPv6 multicast packets, either in the global table or VPN context.

When router is positioned as the core router running mLDP, it supports the profiles 1, 2, 4, 5, 6,7, 9, 12, 13, 14, 15, 17, 19, 21, 23, 25, 27, 28 and 29.

Starting from Cisco IOS XR Release 24.4.1, the mLDP-based mVPN supports the RP placement on the BUD/TAIL node on edge routers. In addition to profiles 21 and 22, the BUD node is now supported on profiles 0, 1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 14, 15, 17, 19, 23, 25, 27, 28, and 29.

Benefits of LSM MLDP based MVPN

LSM provides these benefits when compared to GRE core tunnels that are currently used to transport customer traffic in the core:

- It leverages the MPLS infrastructure for transporting IP multicast packets, providing a common data plane for unicast and multicast.
- It eliminates the complexity associated PIM.
- It applies the benefits of MPLS to IP multicast such as Fast ReRoute (FRR). For more information on FRR, see mLDP Loop-Free Alternative Fast Reroute, on page 85

Configuring MLDP MVPN

The MLDP MVPN configuration enables IPv4 multicast packet delivery using MPLS. This configuration uses MPLS labels to construct default and data Multicast Distribution Trees (MDTs). The MPLS replication is used as a forwarding mechanism in the core and edge network. For MLDP MVPN configuration to work,

ensure that the global MPLS MLDP configuration is enabled. To configure MVPN extranet support, configure the source multicast VPN Routing and Forwarding (mVRF) on the receiver Provider Edge (PE) router or configure the receiver mVRF on the source PE. MLDP MVPN is supported for both intranet and extranet.

Packet Flow in mLDP-based Multicast VPN

For each packet coming in, MPLS creates multiple out-labels. Packets from the source network are replicated along the path to the receiver network. The CE1 router sends out the native IP multicast traffic. The Provider Edge1 (PE1) router imposes a label on the incoming multicast packet and replicates the labeled packet towards the MPLS core network. When the packet reaches the core router (P), the packet is replicated with the appropriate labels for the MP2MP default MDT or the P2MP data MDT and transported to all the egress PEs. Once the packet reaches the egress PE (edge routers), the label is removed and the IP multicast packet is replicated onto the VRF interface. Basically, the packets are encapsulated at headend and decapsulated at tailend on the PE routers.

Realizing a mLDP-based Multicast VPN

There are different ways a Label Switched Path (LSP) built by mLDP can be used depending on the requirement and nature of application such as:

- P2MP LSPs for global table transit Multicast using in-band signaling.
- P2MP/MP2MP LSPs for MVPN based on MI-PMSI or Multidirectional Inclusive Provider Multicast Service Instance (Rosen Draft).
- P2MP/MP2MP LSPs for MVPN based on MS-PMSI or Multidirectional Selective Provider Multicast Service Instance (Partitioned E-LAN).

The router performs the following important functions for the implementation of MLDP:

1. Encapsulating VRF multicast IP packet with a Label and replicating to core interfaces (imposition node).
2. Replicating multicast label packets to different interfaces with different labels (Mid node).
3. Decapsulate and replicate label packets into VRF interfaces (Disposition node).

MLDP inband signaling

MLDP Inband signaling allows the core to create (S,G) or (*,G) state without using out-of-band signaling such as BGP or PIM. It is supported in VRF (and in the global context). Both IPv4 and IPv6 multicast groups are supported.

In MLDP Inband signaling, one can configure an ACL range of multicast (S,G). This (S,G) can be transported in MLDP LSP. Each multicast channel (S,G), is 1 to 1 mapped to each tree in the inband tree. The (S,G) join, through IGMP/MLD/PIM, will be registered in MRIB, which is the client of MLDP.

MLDP In-band signalling supports transiting PIM (S,G) or (*,G) trees across a MPLS core without the need for an out-of-band protocol. In-band signaling is only supported for shared-tree-only forwarding (also known as sparse-mode threshold infinity). PIM Sparse-mode behavior is not supported (switching from (*,G) to (S,G).

Multicast Traffic Flow over Multicast Distribution Tree for MVPN (Profile 22) on Edge Routers

Table 17: Feature History Table

Feature Name	Release Information	Feature Description
Multicast Traffic Flow over Multicast Distribution Tree for MVPN (Profile 22) on Edge Routers	Release 24.4.1	Introduced in this release on: Fixed Systems(8700)(select variants only*) *This functionality is now supported on Cisco 8712-MOD-M routers.
Multicast Traffic Flow over Multicast Distribution Tree for MVPN (Profile 22) on Edge Routers	Release 24.1.1	By delivering multicast traffic to specific PE routers that have interested receivers, this feature reduces the amount of replication and bandwidth required for multicast traffic. Plus, Profile 22 in MVPN over edge routers provides enhanced scalability by supporting a large number of MVPNs and multicast groups. It also supports mLDP and P2MP-TE core tree protocols, and enables using the S-PMSI (Selective Provider Multicast Service Interface) to transport traffic over a Multicast Distribution Tree (MDT).

Profile 22, also known as Default MDT-P2MP-TE with BGP C-multicast Routing in MVPN over edge routers provides enhanced scalability by providing support for a large number of MVPNs and multicast groups in your network. It also enables using the S-PMSI (Selective Provider Multicast Service Interface) to transport traffic over a Multicast Distribution Tree (MDT). This profile improves the efficiency by delivering the multicast traffic to specific PE routers that have interested receivers. It not only improves operational performance by reducing the amount of replication and bandwidth required for multicast traffic, but also reduces operational costs as it consolidates multicast and unicast VPNs on the same device.

These are the characteristics of this profile:

- Dynamic P2MP-TE tunnels with BGP C-multicast Routing
- All Upstream Multicast Hop (UMH) options supported.
- Default and Data MDT supported.
- Customer traffic can be SM or SSM .
- RIB-Tail-end-Extranet, RPL-Tail-end-Extranet supported.

- Customer-RP-discovery (Embedded-RP, AutoRP & BSR) is supported.
- Fast Reroute (FRR) is supported.
- Inter-AS Option A supported. Options B and C not supported.
- All PEs for each VRF must have a unique BGP Route Distinguisher (RD) value.

Limitations and User Guidelines

The following limitations and user guidelines are applicable for this feature:

- The P2MP Auto-TE tunnels are used for this profile.
- While using PIM SM and SSM, a physical interface must be multicast enabled in the default VRF.
- The data MDTs are optional. The **ipv4 unnumbered mpls traffic-eng Loopback0** command is a global command. You cannot have the **core-tree-protocol rsvp-te** command configured under the Multicast-Routing VRF one section in the configuration.

Configuration Example for Multicast Traffic Flow over Multicast Distribution Tree for MVPN (Profile 22)

Configure VRF Entry

```
Router#config
Router(config)#vrf one
Router(config-one)#address-family ipv4 unicast
Router(config-one-af)#import route-target
Router(config-one-af)#1:1
Router(config-one-af)#exit
Router(config-one-af)#export route-target
Router(config-one-af)#1:1
Router(config-one-af)#exit
Router(config-one)#exit
Router(config)#commit
```

Assign a Route Policy in PIM to Select a Reverse-Path Forwarding Topology

```
Router#config
Router(config)#router pim
Router(config-pim)#vrf one
Router(config-pim-one)#address-family ipv4
Router(config-pim-one-af)#rpf topology route-policy rpf-vrf-one
Router(config-pim-one-af)#mdt c-multicast-routing bgp
Router(config-pim-one-af)#interface GigabitEthernet0/0/0/1.100
Router(config-pim-one-af-if)#enable
```

Configure route policy to set the MDT type to P2MP-TE default

```
Router#config
Router(config)#route-policy rpf-vrf-one
Router(config-rpl)#set core-tree p2mp-te-default
Router(config-rpl)#end-policy
```

Enable Default MDT-P2MP-TE with BGP C-signalling multicast routing

```
Router#config
Router(config)#multicast-routing
```

```
Router(config-mcast)#vrf one
Router(config-mcast-one)#address-family ipv4
Router(config-mcast-one-af)#mdt source Loopback0
Router(config-mcast-one-af)#mdt default p2mp-te
Router(config-mcast-one-af)#rate-per-route
Router(config-mcast-one-af)#interface all enable
Router(config-mcast-one-af)#mdt data p2mp-te 100
Router(config-mcast-one-af)#bgp auto-discovery p2mp-te
Router(config-mcast-one-af)#accounting per-prefix
Router(config-mcast-one-af)#ipv4 unnumbered mpls traffic-eng Loopback0
Router(config-mcast-one-af)#mpls traffic-eng
Router(config-mcast-one-af)#interface GigabitEthernet0/0/0/0
Router(config-mcast-one-af-if)#exit
Router(config-mcast-one-af)#interface GigabitEthernet0/0/0/2
Router(config-mcast-one-af)#auto-tunnel p2mp
Router(config-mcast-one-af)#tunnel-id min 1000 max 2000
```

Configure Fast Reroute (FRR)

To configur FRR on the head node (R1) configure the following:

```
Router#config
Router(config)#mpls traffic-eng
Router(config)#interface HundredGigE0/0/0/2 -- > (Link 2 or Backup link)
Router(config-if)#exit
Router(config)#interface HundredGigE0/0/0/2 -- > (Link 1 or Protected link)
Router(config-if)#auto-tunnel backup
Router(config-if-auto-backup)#nhop only
Router(config-if-auto-backup)#exit
Router(config-if)#exit
Router(config-if)#auto-tunnel backup
Router(config-if-auto-backup)#tunnel-id min 3000 max 4000
Router(config-if-auto-backup)#exit
Router(config-if)#attribute-set p2mp-te FRR
Router(config-if-attribute-set)#fast-reroute
Router(config-if-attribute-set)#exit
Router(config-if)#reoptimize events link-up
Router(config-if)#exit
Router(config)#multicast routing
Router(config-mcast)#vrf one
Router(config-mcast-one)#address-family ipv4
Router(config-mcast-one-af)#mdt default p2mp-te attribute-set FRR
Router(config-mcast-one-af)#exit
Router(config-mcast-one)#exit
Router(config-mcast)#exit
Router(config)#commit
```

To configure FRR on the Mid node Router 2 (R2), you must configure Router 1 (R1) as well:

To configure R1:

```
Router#config
Router(config)#mpls traffic-eng
Router(config)#auto-tunnel backup
Router(config-auto-backup)#tunnel-id min 3000 max 4000
Router(config-auto-backup)#exit
Router(config)#attribute-set p2mp-te FRR
Router(config-attribute-set)#fast-reroute
Router(config-attribute-set)#exit
Router(config)#reoptimize events link-up
Router(config)#multicast routing
Router(config-mcast)#vrf one
Router(config-mcast-one)#address-family ipv4
```

```
Router(config-mcast-one-af)#mdt default p2mp-te attribute-set FRR
Router(config-mcast-one-af)#exit
Router(config-mcast-one)#exit
Router(config-mcast)#exit
Router(config)#commit
```

To configure R2:

```
Router#config
Router(config)#mpls traffic-eng
Router(config)#interface HundredGigE0/0/0/27 -- > (Link 2 or Backup link)
Router(config-if)#exit
Router(config)#interface HundredGigE0/0/0/31 -- > (Link 1 or Protected link)
Router(config-if)#auto-tunnel backup
Router(config-if-auto-backup)#nhop only
Router(config-if-auto-backup)#exit
Router(config-if)#exit
Router(config-if)#auto-tunnel backup
Router(config-if-auto-backup)#tunnel-id min 3000 max 4000
Router(config-if-auto-backup)#exit
Router(config-if)#reoptimize events link-up
```

Verification

Verify the configuration of profile 22 using the **show mrib vrf p22_20 route detail** command.

```
Router# show mrib vrf p22_20 route detail
IP Multicast Routing Information Base

Entry flags: L - Domain-Local Source, E - External Source to the Domain,
    C - Directly-Connected Check, S - Signal, IA - Inherit Accep
    IF - Inherit From, D - Drop, ME - MDT Encap, EID - Encap ID,
    MD - MDT Decap, MT - MDT Threshold Crossed, MH - MDT interface handle
    CD - Conditional Decap, MPLS - MPLS Decap, EX - Extranet
    MoFE - MoFRR Enabled, MoFS - MoFRR State, MoFP - MoFRR Primary
    MoFB - MoFRR Backup, RPFID - RPF ID Set, X - VXLAN
Interface flags: F - Forward, A - Accept, IC - Internal Copy
    NS - Negate Signal, DP - Don't Preserve, SP - Signal Present,
   II - Internal Interest, ID - Internal Disinterest, LI - Local Interest,
    LD - Local Disinterest, DI - Decapsulation Interface
   EI - Encapsulation Interface, MI - MDT Interface, LVIF - MPLS Encap,
   EX - Extranet, A2 - Secondary Accept, MT - MDT Threshold Crossed,
   MA - Data MDT Assigned, LMI - mLDP MDT Interface, TMI - P2MP-TE MDT Interface
 IRMI - IR MDT Interface, TRMI - TREE SID MDT Interface, MH - Multihome Interface

(192.0.2.0,203.0.1.1) Ver: 0xb009 RPF nbr: 80.0.0.11 Flags: RPF EID,
 PD: Slotmask: 0x0
     MGID: 592
  Up: 00:07:14
  RPF-ID: 0, Encap-ID: 262145
  Incoming Interface List
    TenGigE 0/0/0/2 Flags: A, Up: 00:07:14
  Outgoing Interface List
    Tmdtp22/ssm/v4/vrf2 Flags: F TMI, Up: 00:07:14, Head LSM-ID: 0x4000360
```

Restrictions for mLDP on Edge Routers

The restrictions applicable for mLDP on edge routers are as follows:

- NETCONF/YANG on MVPN for Profile 6 and Profile 7 is not supported.
- MLDP ping traceroute is not supported.

- BVI is not supported.
- Netflow for MPLS-encapsulated multicast packets is not supported.
- MLDP Fast-Reroute (FRR) is not supported on the Tree-SID Profiles 27, 28 and 29.
- Use the **immediate-switch** keyword only for data MDT switchover. Switchover from the default MDT to the data MDT is not supported based on the threshold.
- While using PIM SM and SSM, a physical interface must be multicast enabled in the default VRF.
- RSVP-TE profiles are only supported in Cisco IOS XR Software Release 24.1.1.

mLDP Loop-Free Alternative Fast Reroute

Table 18: Feature History Table

Feature Name	Release Information	Feature Description
mLDP Loop-Free Alternative Fast Reroute	Release 7.3.15	When this feature is enabled, mLDP relies on the Loop-Free Alternative algorithm to calculate the primary and backup, which is also referred as fast re-route path. During the event of a link failure, the router uses this precomputed backup path to send the multicast traffic. The fast switchover helps to reduce multicast traffic loss and the switchover time is less than 50 milliseconds.

Background

Generally, in a network, a network topology change, caused by a failure in a network, results in a loss of connectivity until the control plane convergence is complete. There can be various levels of loss of connectivity depending on the performance of the control plane, fast convergence tuning, and leveraged technologies of the control plane on each node in the network.

The amount of loss of connectivity impacts some loss-sensitive applications, which have severe fault tolerance (typically of the order of hundreds of milliseconds and up to a few seconds). In order to ensure that the loss of connectivity conforms to such applications, a technology implementation for data plane convergence is essential. **Fast Reroute (FRR)** is one of such technologies that is primarily applicable to the network core.

With the FRR solution, at each node, the backup path is pre-computed, and the traffic is routed through this backup path. As a result, the reaction to failure is local; immediate propagation of the failure and subsequent processing on to other nodes is not required. With FRR, if the failure is detected quickly, a loss of connectivity as low as 10s of milliseconds is achieved.

Loop-Free Alternative Fast Reroute

IP Loop Free Alternative FRR is a mechanism that enables a router to rapidly switch traffic to a pre-computed or a pre-programmed **loop-free alternative (LFA)** path, which is Data Plane Convergence, following either an adjacent link and node failure, or an adjacent link or node failure in both IP and LDP networks. The LFA path is used to switch traffic till the router installs the new primary next-hops based upon the changed network topology, which is Control Plane Convergence.

The goal of LFA FRR is to reduce the loss of connectivity to tens of milliseconds by using a pre-computed alternative next-hop, in the case where the selected primary next-hop fails.

There are two approaches to computing LFA paths:

- **Link-based (per-link):** In link-based LFA paths, all prefixes reachable through the primary (protected) link share the same backup information. This means that the whole set of prefixes sharing the same primary also shares the repair and FRR ability.
- **Prefix-based (per-prefix):** Prefix-based LFAs allow computing backup information for each prefix. This means that the repair and backup information computed for a given prefix using prefix-based LFA may be different from the one computed by link-based LFA.

  Protection against a node failure by rerouting traffic around the failed node (Node-protection support) is available with per-prefix LFA FRR on ISIS currently. It uses a tie-breaker mechanism in the code to select node-protecting backup paths.

The per-prefix LFA approach is preferred to the per-link LFA approach for the following reasons:

- Better node failure resistance.
- Better coverage: Each prefix is analyzed independently.
- Better capacity planning: Each flow is backed up on its own optimized shortest path.

mLDP LFA FRR

The point-to-point physical or bundle interface FRR mechanism is supported on mLDP. FRR with LFA backup is supported on mLDP. When there is a link failure, mLDP automatically sets up and chooses the backup path.

With this implementation, you must configure the physical or bundle interface for unicast traffic, so that the mLDP can act as an mLDP FRR.

LFA FRR support on mLDP is a per-prefix backup mechanism. As part of computing the LFA backup for a remote IP, the LFA backup paths for the loopback address of the downstream intermediate nodes are also computed. MLDP uses this small subset of information, by using the loopback address of the peer to compute the LFA backup path.

Note Both IPv4 and IPv6 traffic is supported on the mLDP LFA FRR solution.

MLDP LFA FRR - Features

- Supports both IPv4 and IPv6 multicast traffic carried by MLDP label.
- Supports all MLDP profiles and behaves both as MLDP core router and MLDP edge router.

- Supports both ISIS and OSPF routing protocols

Advantages of LFA FRR

The following are the advantages of the LFA FRR solution:

- The backup path for the traffic flow is pre-computed, so that it help in faster convergence.
- Reaction to failure is local, an immediate propagation and processing of failure on to other nodes is not required.
- If the failure is detected in time, the loss of connectivity of up to 50 milliseconds can be achieved.
- The mechanism is locally significant and does not impact the Interior Gateway Protocol (IGP) communication channel.
- LFA next-hop can protect against:
  - a single link failure
  - failure of one of more links within a shared risk link group (SRLG)
  - any combination of the above
- Supports switchover time of less than 50 milliseconds.
- Supports switchover time to be independent of the number of multicast routes that has to be switched over.

Limitations of LFA FRR

The following are some of the known limitations of the LFA FRR solution:

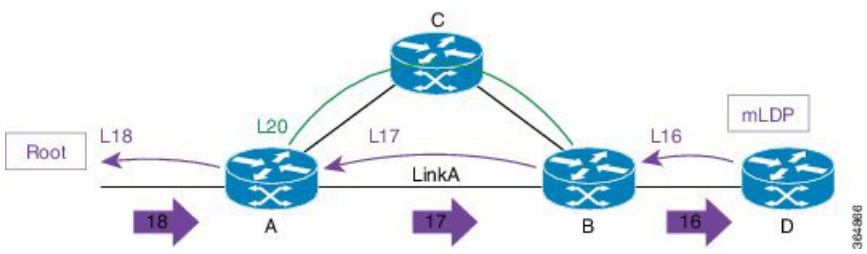
- When a failure that is more extensive than that which the alternate path was intended to protect occurs, there is the possibility of temporarily looping traffic (micro looping) until Control Plane Convergence.

MLDP LFA FRR - Workflow

To enable FRR for mLDP over physical or bundle interfaces, LDP session-protection feature has to be configured. The sequence of events that occur in an mLDP LFA FRR scenario is explained with the following example:

1. Step 1: MLDP LFA FRR - Initial Setup

Figure 7: MLDP LFA FRR - Setup



a. In this set up, Router A is the source provider edge router, and the next Hop is Router B. The primary path is Router A -> Router B - > Router D, and the backup path is from Router A -> Router C -> Router B -> Router D. The backup path is pre-computed by IGP through LFA prefix-based selection.

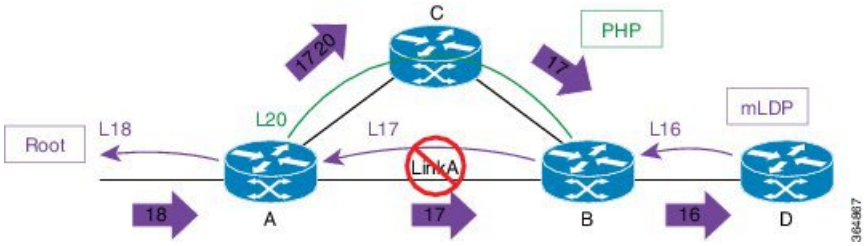
b. Backup paths are configured for Link A or auto-tunnels are enabled.

c. MLDP LSP is built from D, B, and A towards the root.

d. Router A installs a downstream forwarding replication over link A to Router B. This entry has both the primary interface (Link A) and the backup paths programmed.

2. Step 2: Link Failure

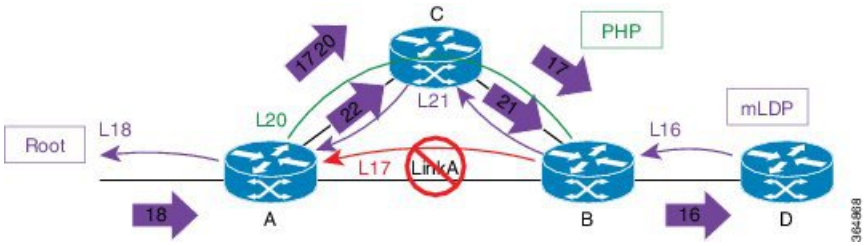
Figure 8: Link Failure



- When a failure occurs on Link A:
  a. Traffic over link A is rerouted over the backup path with same MLDP Label 17 (inner label) , plus a unicast label 20 (outer label) towards mid Router C.
  b. Router C performs penultimate hop popping (PHP) and removes the outer label 20.
  c. Router B receives the mLDP packets with label 17 and forwards to Router D.

3. Step 3: Re-optimization

Figure 9: Re-optimization



During re-optimization:

a. mLDP is notified that the root is reachable through Router C, and mLDP converges. With this, a new mLDP path is built to router A through Router C.

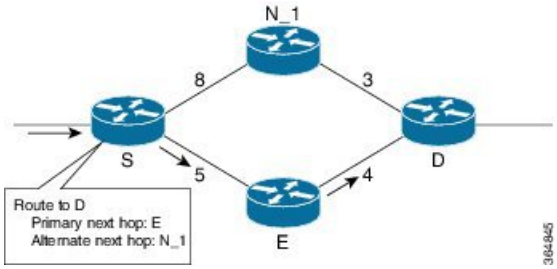
b. Router A forwards packets natively with old label 17 and also new label 22.

c. Router B drops traffic carried from new label 22 and forwards traffic with label 17.

MLDP LFA FRR - Behavior

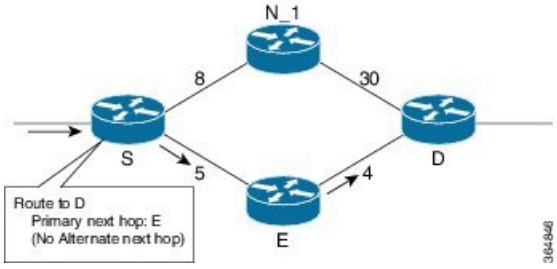
In the following scenarios, S is source router, D is the destination router, E is primary next hop, and N_1 is the alternative next hop.

Figure 10: LFA FRR Behavior - LFA Available



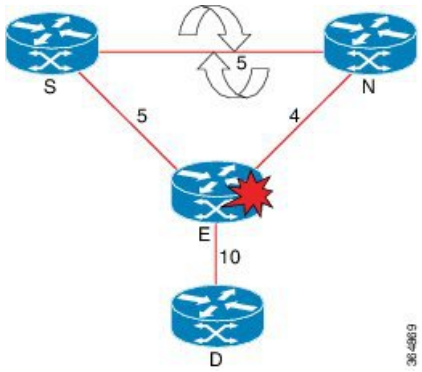
With LFA FRR, the source router S calculates an alternative next hop N_1 to forward traffic towards the destination router D through N_1, and installs N_1 as a the alternative next hop. On detecting the link failure between routers S and E, router S stops forwarding traffic destined for router D towards E through the failed link; instead it forwards the traffic to a pre-computed alternate next hop N_1, until a new SPF is run and the results are installed.

Figure 11: LFA FRR Behavior - LFA Not Available



In the above scenario, if the link cost between the next hop N_1 and the destination router D is increased to 30, then the next hop N_1 would no longer be a loop-free alternative. (The cost of the path, from the next hop N_1 to the destination D through the source S, would be 17, while the cost from the next hop N_1 directly to destination D would be 30). Thus, the existence of a LFA next hop is dependent on the topology and the nature of the failure, for which the alternative is calculated.

Figure 12: Link Protecting LFA



In the above illustration, if router E fails, then both router S and router N detects a failure and switch to their alternates, causing a forwarding loop between both routers S and N. Thus, the link protecting LFA causes loop on node failure; however, this can be avoided by using a down-stream path, which can limit the coverage of alternates. Router S will be able to use router N as a downstream alternate, however, router N can't use S. Therefore, N would have no alternate and would discard the traffic, thus avoiding the micro looping.

Configuring MLDP Loop-Free Alternative Fast Reroute

The following section describes the configurations to enable LFA FRR:

Note mLDP FRR relies on the IGP protocol, you can either configure OSPF or ISIS.

Configuring Router OSPF LFA FRR

mLDP FRR relies on the IGP protocol, you can configure either OSPF or ISIS.

The OSPF LFA FRR uses a precomputed alternate next hop to reduce failure reaction time when the primary next hop fails. It lets you configure a per-prefix LFA path that redirects traffic to a next hop other than the primary neighbor.

Configuration Example

```
Router# configure
Wed Apr  7 08:54:52.769 UTC
Router(config)# router ospf 0
Router(config-ospf)# area 0
Router(config-ospf-ar)# interface Bundle-Ether10

/*Enabling Per Prefix LFA*/
Router(config-ospf-ar-if)# fast-reroute per-prefix

/*To add interfaces to LFA Candidate List:*/
Router(config-ospf-ar-if)# fast-reroute per-prefix lfa-candidate interface Bundle-Ether10

/*To exclude interface from backup*/
Router(config-ospf-ar-if)# fast-reroute per-prefix exclude interface Bundle-Ether10

/*To restrict the backup interface to the LFA candidate list:*/
Router(config-ospf-ar-if)# fast-reroute per-prefix use-candidate-only enable
Router(config-ospf-ar-if)# commit
```

Running Configuration

```
router ospf 0
 area 0
  interface Bundle-Ether10
   fast-reroute per-prefix
   fast-reroute per-prefix exclude interface Bundle-Ether10
   fast-reroute per-prefix lfa-candidate interface Bundle-Ether10
   fast-reroute per-prefix use-candidate-only enable
  !
 !
!
```

Configuring Router ISIS LFA FRR

IS-IS computes LFA next-hop routes for the forwarding plane to use in case of primary path failures. LFA is computed per prefix.

Configuration Example

```
Router# configure
Router(config)# router isis MCAST
Router(config-isis)# net 49.0001.0000.0000.0001.00
Router(config-isis-af)# interface HundredGigE0/0/24
Router(config-isis-if-af)# address-family ipv4 unicast
/*configure per-prefix Link based LFA*/
Router(config-isis-if-af)# fast-reroute per-prefix
```

Running Configuration

```
!
router isis MCAST
 net 49.0001.0000.0000.0001.00
 interface HundredGigE0/0/0/24
  address-family ipv4 unicast
   fast-reroute per-prefix  <---- configure per-prefix Link based LFA
  !
 !
!
```

Configuring Bidirectional Forwarding Detection

When a local interface is down, it can take a long delay for the remote peer to detect the link disconnection. To quickly detect if the remote interface is down, the physical port and bundle interfaces must have Bidirectional Forwarding Detection (BFD) to ensure faster failure detection.

```
Router#configure
Router(config)#router ospf 0
Router(config-ospf)#nsr
Router(config-ospf)#router-id 21.21.21.21
Router(config-ospf)#nsf cisco
Router(config-ospf)#address-family ipv4 unicast
Router(config-ospf)#area 0
Router(config-ospf-ar)#bfd minimum-interval 3
Router(config-ospf-ar)#bfd fast-detect
Router(config-ospf-ar)#bfd multiplier 2
Router(config-ospf-ar)#fast-reroute per-prefix
Router(config-ospf-ar)#mpls traffic-eng
Router(config-ospf-ar)#interface Bundle-Ether100.1
Router(config-ospf-ar-if)#bfd fast-detect
Router(config-ospf-ar-if)#fast-reroute per-prefix

Router(config-ospf-ar)#interface Bundle-Ether100.2
Router(config-ospf-ar-if)#bfd fast-detect
Router(config-ospf-ar-if)#fast-reroute per-prefix
Router(config-ospf-ar-if)#commit
```

Running Configuration

```
router ospf 0
 nsr
```

```
router-id 21.21.21.21
nsf cisco
address-family ipv4 unicast
area 0
 bfd minimum-interval 3
 bfd fast-detect   <---- configure bfd fast-detect
 bfd multiplier 2
 fast-reroute per-prefix
 mpls traffic-eng
 interface Bundle-Ether100.1
  bfd fast-detect    <---- configure bfd fast-detect under the "protected" inerface
  fast-reroute per-prefix
 !
 interface Bundle-Ether100.2
  bfd fast-detect   <---- configure bfd fast-detect under the "protected" inerface
  fast-reroute per-prefix
 !
```

For bundle main interface, configure BFD under the bundle interface:

Note The **bundle minimum-active links** is required if LACP is not configured on the bundle members.

```
interface Bundle-Ether101
bfd address-family ipv4 timers start 60
bfd address-family ipv4 timers nbr-unconfig 3600
bfd address-family ipv4 multiplier 2
bfd address-family ipv4 destination 44.2.0.4
bfd address-family ipv4 fast-detect
bfd address-family ipv4 minimum-interval 3
ipv4 address 44.2.0.1 255.255.255.0
ipv6 address 44:2::1/64
bundle minimum-active links 3
!
```

For LACP, configure **mode active** under bundle member:

```
interface HundredGigE0/0/0/22
bundle id 101 mode active
!
interface HundredGigE0/0/0/28
bundle id 101 mode active
!
interface HundredGigE0/0/0/29
bundle id 101 mode active
!
```

For physical interface and subinterface and bundle subinterface, configure BFD under IGP, for example ISIS:

```
router isis ring
interface HundredGigE0/0/0/22
  bfd minimum-interval 10
  bfd multiplier 2
  bfd fast-detect ipv4
  point-to-point
  address-family ipv4 unicast
   fast-reroute per-prefix
```

```
   metric 1000
   weight 1000
  !
!
```

Configuring MPLS LFA FRR

Configuration Example

Configure session protection to support MLDP LFA FRR:

```
Router# configure
Router(config)# mpls ldp
Router(config-ldp)# nsr
Router(config-ldp)# graceful-restart
Router(config-ldp)# router-id 20.20.20.20
Router(config-ldp)# session protection
Router(config-ldp)# address-family ipv4
Router(config-ldp-af)#commit
```

Show Running Configuration

```
mpls ldp
  nsr
  graceful-restart
 !
 nsr
 graceful-restart
 router-id 20.20.20.20
 session protection
 address-family ipv4
 !
!
```

Make Before Break Configuration for LFA FRR

Make Before Break (MBB) is an inherent nature of MLDP. In MBB configuration, configure **forwarding recursive** to enable LFA FRR feature. If forwarding recursive is not configured, MLDP uses non-recursive method to select MLDP core facing interface towards next hop. The detailed configuration steps and an example follows.

```
Router(config)# mpls ldp
Router(config-ldp)# log
Router(config-ldp-log)# neighbor
Router(config-ldp-log)# nsr
Router(config-ldp-log)# graceful-restart
Router(config-ldp-log)# mldp
Router(config-ldp-mldp)# address-family ipv4
Router(config-ldp-mldp-af)# forwarding recursive
Router(config-ldp-mldp-af)# make-before-break delay 60
Router(config-ldp-mldp-af)# commit
```

Configuring Make Before Break Delay and Delete

By default, MBB is set to 10 seconds. You can configure different MBB timing to determine when the merge node starts to accept the new label.

In this configuration example, the MBB (delay) period is set of 90 seconds. The merge node starts accepting new label 90 seconds after detecting the link disconnection towards the head node. The delete delay is set to 60 seconds; that is, when MBB expires, the time period after which the merge node sends old label delete request to head node is 60 seconds. The default value is zero. The range of delete delay is from 30 to 60, for scale LSPs.

```
Router# configure
Router(config)# mpls ldp
Router(config-ldp)# mldp
Router(config-ldp-mldp)# address-family ipv4
Router(config-ldp-mldp-af)# make-before-break delay 90
Router(config-ldp-mldp-af)# make-before-break delay 90 60
Router(config-ldp-mldp-af)# commit
```

Verification of MLDP Configuration

Use the following show commands to verify the mLDP LFA FRR configuration:

The following example shows how to verify mLDP Neighbor:

```
Router# show mrib regdb
Tue Mar 23 17:45:27.762 UTC
  NH  addr          : 45.45.45.45    <--- Next Hop Peer's Loopback Address
  Destination vrf   : default
  Regdb Entry Type  : Label
  IP Ole count      : 0x0
  Label Ole count   : 0x2
  MLC Ole count     : 0x0
  ECD registered    : YES
  ECD stale         : NO
  ECD Information   : 55a76e23e000
  ECD Length        : 50
  Number of notifs  : 1
```

The following example shows how to verify the mLDP traffic. The zero in the following example indicates that there's no mLDP packet forwarding out of that outgoing interface.

```
Router# show mpls forwarding p2mp
Local Outgoing Prefix Outgoing Next Hop Bytes
Label Label or ID Interface Switched
------ ----------- ------------------ ------------ --------------- ------------
40127 40018 mLDP/IR: 0x003e9 Hu0/3/0/0 61.154.2.50 113972905604
40045 mLDP/IR: 0x003e9 Hu0/3/0/26 82.154.5.2 113339018314
40128 40019 mLDP/IR: 0x003ea Hu0/1/0/17/1 97.1.1.5 113972768600
40046 mLDP/IR: 0x003ea Hu0/3/0/32 82.154.4.2 0
40129 40020 mLDP/IR: 0x003eb Hu0/1/0/17/1 97.1.1.5 113972234482
40047 mLDP/IR: 0x003eb Hu0/3/0/26 82.154.5.2 0
40130 40021 mLDP/IR: 0x003ec Hu0/1/0/5 10.10.10.17 113972828144
40048 mLDP/IR: 0x003ec Hu0/3/0/26 82.154.5.2 0
40131 40022 mLDP/IR: 0x003ed Hu0/1/0/17/1 97.1.1.5 113973181832
40049 mLDP/IR: 0x003ed Hu0/3/0/32 82.154.4.2 0
40132 40023 mLDP/IR: 0x003ee Hu0/1/0/17/1 97.1.1.5 113972294384
40050 mLDP/IR: 0x003ee Hu0/3/0/26 82.154.5.2 0
40133 40024 mLDP/IR: 0x003ef Hu0/3/0/0 61.154.2.50 113972687482
```

The following example shows how to view the list of mLDP in the router:

```
Router# show mrib mpls forwarding detail
```

```
LSP information (mLDP) :
  LSM-ID: 0x00014, Role: Mid
    Incoming Label       : 24028
    Transported Protocol : <unknown>
    Explicit Null        : None
    IP lookup            : disabled
    Platform information : MCGID: 56633, Tunnel RIF: -1, RIF VRF: -1  <--- The local label
 24028 has MCGID: 56633,

          used for programming label's FAP ID bitmask
    Outsegment Info #1 [M/Swap, Recursive]:
      OutLabel: 24027, NH: 45.45.45.45, ID: 0x14, Sel IF: Bundle-Ether101(V)  <---Primary
path BE101 (HundredGigE0/2/0/17)
        UL IF: HundredGigE0/2/0/17, Node-ID: 0x9
      Backup Tunnel: Un:0x0  Backup State: Ready, NH: 0.0.0.0, MP Label: 0
        Backup Sel IF: Bundle-Ether102(V), UL IF: HundredGigE0/0/0/13, Node-ID: 0x1
<-----Backup path BE102 (HundredGigE0/0/0/13).

LSP information (mLDP) :
  LSM-ID: 0x00015, Role: Mid
    Incoming Label       : 24029
    Transported Protocol : <unknown>
    Explicit Null        : None
    IP lookup            : disabled
    Platform information : MCGID: 56634, Tunnel RIF: -1, RIF VRF: -1

    Outsegment Info #1 [M/Swap, Recursive]:
      OutLabel: 24028, NH: 45.45.45.45, ID: 0x15, Sel IF: Bundle-Ether101(V)
        UL IF: HundredGigE0/2/0/18, Node-ID: 0x8
      Backup Tunnel: Un:0x0  Backup State: Ready, NH: 0.0.0.0, MP Label: 0
        Backup Sel IF: Bundle-Ether102(V), UL IF: HundredGigE0/0/0/27, Node-ID: 0x2
```

The following example shows how to view the details of a specific mLDP.

```
Router# show mpls forwarding labels 24028 detail
Tue Mar 23 17:47:28.962 UTC
Local  Outgoing    Prefix            Outgoing     Next Hop        Bytes
Label  Label       or ID             Interface                    Switched
------ ----------- ----------------- ------------ --------------- ------------
24028              mLDP/IR: 0x00014 (0x00014)
       Updated Mar 23 17:28:29.946
       mLDP/IR LSM-ID: 0x00014, MDT: 0x0
       Flags:IP Lookup:not-set, Expnullv4:not-set, Expnullv6:not-set
             Payload Type v4:not-set, Payload Type v6:not-set, l2vpn:not-set
             Head:not-set, Tail:not-set, Bud:not-set, Peek:not-set, inclusive:not-set
             Ingress Drop:not-set, Egress Drop:not-set
             RPF-ID:0, Encap-ID:0
             Disp-Tun:[ifh:0x0, label:-]
             Platform Data [28]:
              { 0 0 221 57 0 0 0 4
                0 0 0 2 0 0 0 0
                144 207 44 47 255 255 255 255
                0 0 0 0 }
       mpls paths: 1, local mpls paths: 1, protected mpls paths: 1

       24027       mLDP/IR: 0x00014 (0x00014)   \
                                    BE101        44.2.0.4        1130065367760
         Updated: Mar 23 17:28:29.952
         My Nodeid:0x2000
         Interface Nodeids:
           [ 0x9 - - - - - - - - -  ]
         Interface Handles:
           [ 0x1000218 - - - - - - - - -  ]
         Backup Interface Nodeids:
```

```
    [ 0x1 - - - - - - - - - ]
  Backup Interface Handles:
    [ 0x240 - - - - - - - - - ]
  Packets Switched: 1121096595
```

The following example shows how to see the mLDP neighbors:

```
Router# show mpls mldp neighbors
Sat May  9 06:37:06.877 UTC
mLDP neighbor database
 MLDP peer ID      : 20.20.20.20:0, uptime 01:38:38 Up,
  Capabilities     : GR, Typed Wildcard FEC, P2MP, MP2MP, MBB
  Target Adj       : No
  Upstream count   : 1
  Branch count     : 2
  LDP GR           : Enabled
                   : Instance: 1
  Label map timer  : never
  Policy filter in : None
  Path count       : 2
  Path(s)          : 21.20.100.1       Bundle-Ether100 LDP
                   : 21.20.20.2        Bundle-Ether20.1 LDP
  Adj list         : 21.20.20.2        Bundle-Ether20.1
                   : 21.20.100.1       Bundle-Ether100
  Peer addr list   : 172.18.51.116
                   : 20.20.20.20
                   : 20.22.5.1
                   : 22.20.23.2
                   : 21.20.20.2
                   : 21.20.100.1

 MLDP peer ID      : 22.22.22.22:0, uptime 01:38:38 Up,
  Capabilities     : GR, Typed Wildcard FEC, P2MP, MP2MP
  Target Adj       : No
  Upstream count   : 0
  Branch count     : 2
  LDP GR           : Enabled
                   : Instance: 1
  Label map timer  : never
  Policy filter in : None
  Path count       : 2
  Path(s)          : 22.21.23.1        TenGigE0/1/0/3 LDP
                   : 22.21.20.1        TenGigE0/2/1/0 LDP
  Adj list         : 22.21.20.1        TenGigE0/2/1/0
                   : 22.21.23.1        TenGigE0/1/0/3
  Peer addr list   : 172.18.51.118
                   : 22.22.22.22
                   : 20.22.5.2
                   : 22.2.9.1
                   : 22.20.23.1
                   : 22.21.20.1
                   : 22.21.23.1#
```

Rosen-GRE Multicast VPN for Profiles 0, 3, and 11

Table 19: Feature History Table

Feature Name	Release Information	Feature Description
Rosen-GRE Multicast VPN for Profiles 0, 3, and 11	Release 24.4.1	Introduced in this release on: Fixed Systems(8200 , 8700);Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This feature is now supported on: • 8212-32FH-M • 8711-32FH-M • 8712-MOD-M • 88-LC1-12TH24FH-E
Rosen-GRE Multicast VPN for Profiles 0, 3, and 11	Release 24.2.11	Rosen-GRE is a widely used MVPN model and uses GRE tunnels to securely transmit multicast traffic between the PE routers. It also enables ease of deployment by using the Protocol-Independent Multicast (PIM) protocol between edge routers (PE) and hosts (CE), and between PE routers that are running in VRF mode. You can now configure MVPN using Rosen-GRE for profiles 0, 3, and 11.

In Rosen GRE, the MCAST customer packets (c-packets) are encapsulated into the provider MCAST packets (p-packets), so that the PIM protocol is enabled in the provider core, and mrib/mfib is used for forwarding p-packets in the core. MVPN uses GRE with unique multicast distribution tree (MDT) forwarding to enable scalability of native IP Multicast in the core network. Rosen-GRE Multicast VPN uses Generic Routing Encapsulation (GRE) as an overlay protocol. All multicast packets are encapsulated inside GRE. In Cisco IOS XR Release 24.2.1, Profiles 0, 3, and 11 are supported.

Note In native multicast and draft-rosen mVPN, load-balancing between bundle members is performed based on outer IP/User Datagram Protocol (UDP) header fields. However, in all MPLS based mVPN design solution, a single tunnel is pinned down to a bundle member.

Characteristics of mLDP Profiles

Profile 0: Rosen-GRE with PIM C-Mcast Signaling

These are the characteristics of profile 0:

- PIM is used as the multicast routing protocol between the edge routers (PE) and hosts (CE), and between the PE routers in the VRF mode.
- Default and Data MDTs are supported.
- The PE routers directly connect using a Default Multicast Distribution Tree (MDT) formed between the PE routers.
- The PE routers connect to each other as PIM neighbors across the Default MDT.

Profile 3:Rosen-GRE with BGP-AD

These are the characteristics of profile 3:

- PIM-trees are used in the core. The data encapsulation method used is GRE.
- SM or SSM is used in the core.
- Default and Data MDTs are supported.
- The multicast traffic can be SM or SSM.
- MoFRR in the core is supported.
- Extranet, Hub and Spoke, CsC, Customer-RP-discovery (Embedded-RP, Auto-RP, and BSR) in the core are supported.
- Inter-AS Options A and C are supported. VRF-Route-Import EC is announced in VPN-IP routes.

Profile 11 : Rosen-PIM/GRE with BGP C-multicast Routing

These are the characteristics of profile 11:

- PIM trees in the core use GRE for data encapsulation and BGP C-multicast routing.
- Static configuration of (S,G) is required on the head-end PE.
- All UMH options are supported for PIM-SSM core-tree and PIM-SM core-tree with no SPT-infinity.
- For PIM-SM core-tree with SPT-infinity, only SFS (highest PE or hash-of-BGP-paths) is supported. The hash-of-installed-paths method is not supported.
- Default and Data MDTs are supported.
- Customer traffic can be SM or SSM.
- Inter-AS Options A and C are supported. Option B is not supported.
- All PEs must have a unique BGP Route Distinguisher (RD) value.

Benefits of Using Rosen-GRE Multicast VPN

- Rosen-GRE (profiles 0, 3, and 11) is a widely used model that is fairly easy to deploy. It uses native multicast in the core and does not require any additional configuration on customer routers or in the core.

Guidelines and Limitations for Configuring Rosen-GRE Multicast VPN

- IPv6 is not supported for the core.
- BVI is not supported.
- When using PIM-SM or PIM-SSM, a physical interface must be multicast-enabled in the default VRF.
- If an IPv4 unicast GRE tunnel is configured in your network, the Maximum Transmission Unit (MTU) size of the configured unicast GRE tunnel impacts the MTU of the Profile-0 MDT multicast. Ensure that the Profile-0 MDT multicast packet size does not exceed the MTU value of the IPv4 unicast GRE tunnel. If the multicast packet size exceeds the MTU value of the tunnel, the packet is dropped.
- Use the `immediate-switch` keyword only for data MDT switchover. Switchover from the default MDT to the data MDT based on the threshold is not supported.

Configure MVPN using Rosen-GRE

To configure Rosen-GRE, use the **route-policy** command.

To configure Profile 0 on the PE devices, perform the following steps:

```
/*Configure route-policy*/
Router# configure
Router(config)# route-policy rosen-gre
Router(config-rpl)#  set core-tree pim-default
Router(config-rpl)# end-policy

/*Enable Multicast Distribution Tree (MDT) within a VPN*/
Router(config)# multicast-routing
Router(config-mcast)# vrf vpn101
Router(config-mcast-vpn101)# address-family ipv4
Router(config-mcast-vpn101-ipv4)#  mdt source Loopback0
Router(config-mcast-vpn101-ipv4)#   mdt default ipv4 233.252.0.1
Router(config-mcast-vpn101-ipv4)#   mdt data 233.252.0.2/24 immediate-switch
Router(config-mcast-vpn101-ipv4)#   interface all enable

/*Set up Protocol Independent Multicast (PIM) within a Virtual Routing and Forwarding (VRF)
 instance*/
Router(config)# router pim
Router(config-pim)# address-family ipv4
Router(config-pim-default-ipv4)# vrf vpn101
Router(config-pim-vpn101)#  address-family ipv4
Router(config-pim-vpn101-ipv4)#   rpf topology route-policy rosen-gre
Router(config-pim-vpn101-ipv4)#   exit
Router(config-pim-vpn101-ipv4)# commit
```

For more information about configuring mVPN profiles (Profiles 3 and 11), see Configure mVPN Profiles within Cisco IOS XR.

Running Configuration

/*Head Configuration*/:

```
hostname PE1
logging console disable
vrf vpn101
```

```
 address-family ipv4 unicast
  import route-target
   1:1
  !
  export route-target
   1:1
  !
 !
 address-family ipv6 unicast
  import route-target
   1:1
  !
  export route-target
   1:1
  !
 !
!

interface Bundle-Ether1
 load-interval 30
 l2transport
 !
!
interface Loopback0
 ipv4 address 192.0.2.1 198.51.100.1
 ipv6 address 2001:DB8::/64
!
interface Loopback1
 vrf vpn101
 ipv4 address 198.51.100.2 198.51.100.1
 ipv6 address 2001:DB8::1:1/64
!
interface MgmtEth0/RP0/CPU0/0
 ipv4 address 192.0.2.53 255.255.0.0
!
interface TenGigE0/0/0/4
 ipv4 address 198.51.100.1 255.255.255.0
 ipv6 address 2001:DB8::2:1/124
 load-interval 30
!
interface TenGigE0/0/0/18
 vrf vpn101
 ipv4 address 198.51.100.2 255.255.0.0
 ipv6 address 2001:DB8::1/124
 load-interval 30
!
!
route-policy PASS
  pass
end-policy
!
route-policy rosen-gre
  set core-tree pim-default
end-policy
!

router static
 address-family ipv4 unicast
  192.0.2.0/24 192.0.5.1
 !
!

router bgp 100
 bgp router-id 192.0.2.1
```

```
 address-family ipv4 unicast
  redistribute connected
  allocate-label all
 !
 address-family vpnv4 unicast
 !
 address-family ipv6 unicast
  redistribute connected
  allocate-label all
 !
 address-family vpnv6 unicast
 !
 address-family ipv4 mdt
 !
 address-family ipv4 mvpn
 !
 address-family ipv6 mvpn
 !
 neighbor 192.0.2.21
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
  !
  address-family vpnv4 unicast
  !
  address-family ipv6 unicast
  !
  address-family vpnv6 unicast
  !
  address-family ipv4 mdt
  !
  address-family ipv4 mvpn
  !
  address-family ipv6 mvpn
  !
 !
 vrf vpn101
  rd 1:1
  address-family ipv4 unicast
   route-target download
   redistribute connected
  !
  address-family ipv6 unicast
   route-target download
   redistribute connected
  !
 !
!

multicast-routing
 address-family ipv4
  interface all enable
 !
 address-family ipv6
  interface all enable
 !
 vrf vpn101
  address-family ipv4
   mdt source Loopback0
   rate-per-route
   interface all enable
   mdt default ipv4 233.252.2.1
   mdt data 233.252.0.0/24 immediate-switch
  !
```

```
  address-family ipv6
   mdt source Loopback0
   rate-per-route
   interface all enable
   mdt default ipv4 233.252.1.1
   mdt data 233.252.0.2/8 immediate-switch
  !
 !
!
lldp
!
router pim
 vrf vpn101
  address-family ipv4
   rpf topology route-policy rosen-gre
   hello-interval 1
   rp-address 192.0.3.1
  !
  address-family ipv6
   rpf topology route-policy rosen-gre
   hello-interval 1
  !
 !
!
end
```

/*Tail Configuration*/:

```
hostname PE2
logging console disable
vrf vpn101
 address-family ipv4 unicast
  import route-target
   1:1
  !
  export route-target
   1:1
  !
 !
 address-family ipv6 unicast
  import route-target
   1:1
  !
  export route-target
   1:1
  !
 !
!

interface Loopback0
 ipv4 address 192.0.2.21 198.51.100.1
 ipv6 address 2001:DB8::21:21/124
!
interface Loopback1
 vrf vpn101
 ipv4 address 192.0.3.1 198.51.100.1
 ipv6 address 2001:DB8::1:1:1/124
!
interface MgmtEth0/RP0/CPU0/0
 ipv4 address 198.51.100.57 255.255.0.0
 shutdown
!
interface TenGigE0/0/0/4
```

```
 ipv4 address 192.0.2.2 255.255.255.0
 ipv6 address 2001:DB8::10:1:1/124
 load-interval 30
!
interface TenGigE0/0/0/6
 vrf vpn101
 ipv4 address 203.0.113.1 255.255.0.0
 ipv6 address 2001:DB8::3:1/124
 load-interval 30
!
!
route-policy PASS
  pass
end-policy
!
route-policy rosen-gre
  set core-tree pim-default
end-policy
!

router static
 address-family ipv4 unicast
  192.0.2.0/24 192.0.5.1
 !
!

router bgp 100
 bgp router-id 192.0.2.21
 bgp graceful-restart
 address-family ipv4 unicast
  redistribute connected
 !
 address-family vpnv4 unicast
 !
 address-family ipv6 unicast
  redistribute connected
 !
 address-family vpnv6 unicast
 !
 address-family ipv4 mdt
 !
 address-family ipv4 mvpn
 !
 address-family ipv6 mvpn
 !
 neighbor 192.0.2.1
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
  !
  address-family vpnv4 unicast
  !
  address-family ipv6 unicast
  !
  address-family vpnv6 unicast
  !
  address-family ipv4 mdt
  !
  address-family ipv4 mvpn
  !
  address-family ipv6 mvpn
  !
 !
 vrf vpn101
```

```
  rd 1:1
  address-family ipv4 unicast
   route-target download
   redistribute connected
  !
  address-family ipv6 unicast
   route-target download
   redistribute connected
  !
 !
!
multicast-routing
 address-family ipv4
  interface all enable
 !
 address-family ipv6
  interface all enable
 !
 vrf vpn101
  address-family ipv4
   mdt source Loopback0
   rate-per-route
   interface all enable
   mdt default ipv4 233.252.2.1
   mdt data 233.252.0.0/24 immediate-switch
  !
  address-family ipv6
   mdt source Loopback0
   rate-per-route
   interface all enable
   mdt default ipv4 233.252.1.1
   mdt data 233.252.0.2/8 immediate-switch
  !
 !
!
lldp
!
router pim
 vrf vpn101
  address-family ipv4
   rpf topology route-policy rosen-gre
   hello-interval 1
   rp-address 192.0.3.1
  !
  address-family ipv6
   rpf topology route-policy rosen-gre
   hello-interval 1
  !
 !
!
router igmp
 vrf vpn101
  interface  interface TenGigE0/0/0/6
   static-group 233.252.4.1 192.0.4.1
  !
 !
!
end
```

Verification

```
Router# show mrib vrf vpn101 route detail
Wed Aug  9 10:20:40.486 UTC
```

```
IP Multicast Routing Information Base
Entry flags: L - Domain-Local Source, E - External Source to the Domain,
    C - Directly-Connected Check, S - Signal, IA - Inherit Accept,
    IF - Inherit From, D - Drop, ME - MDT Encap, EID - Encap ID,
    MD - MDT Decap, MT - MDT Threshold Crossed, MH - MDT interface handle
    CD - Conditional Decap, MPLS - MPLS Decap, EX - Extranet
    MoFE - MoFRR Enabled, MoFS - MoFRR State, MoFP - MoFRR Primary
    MoFB - MoFRR Backup, RPFID - RPF ID Set, X - VXLAN
Interface flags: F - Forward, A - Accept, IC - Internal Copy,
    NS - Negate Signal, DP - Don't Preserve, SP - Signal Present,
    II - Internal Interest, ID - Internal Disinterest, LI - Local Interest,
    LD - Local Disinterest, DI - Decapsulation Interface
    EI - Encapsulation Interface, MI - MDT Interface, LVIF - MPLS Encap,
    EX - Extranet, A2 - Secondary Accept, MT - MDT Threshold Crossed,
    MA - Data MDT Assigned, LMI - mLDP MDT Interface, TMI - P2MP-TE MDT Interface
    IRMI - IR MDT Interface, TRMI - TREE SID MDT Interface, MH - Multihome Interface

(*,233.252.0.4/4) Ver: 0x9b0d RPF nbr: 192.0.3.1 Flags: L C RPF P, MRID: 13, MCGID: 138,
FLAGS: 0x1, Stats(T): 0/0/0
  Up: 00:03:33
  Outgoing Interface List
    Decapstunnel1 Flags: NS DI, Up: 00:03:28

(*,233.252.0.4/24) Ver: 0xec6c Flags: D P, MRID: 7, MCGID: 38, FLAGS: 0x0, Stats(F)
  Up: 00:03:33

(*,233.252.0.3) Ver: 0xe7dc Flags: S P, MRID: 5, MCGID: 36, FLAGS: 0x0, Stats(F)
  Up: 00:03:33

(*,233.252.0.1) Ver: 0xf5fb Flags: S P, MRID: 6, MCGID: 37, FLAGS: 0x0, Stats(F)
  Up: 00:03:33
  Outgoing Interface List
    TenGigE0/0/0/6 Flags: II LI, Up: 00:03:33

(*,233.252.0.2/8) Ver: 0x96d1 Flags: D P, MRID: 8, MCGID: 39, FLAGS: 0x0, Stats(F)
  Up: 00:03:33

(192.0.2.2,233.252.4.1) Ver: 0x2c3f RPF nbr: 192.0.2.1 Flags: RPF RPFID, MRID: 14, MCGID:
139, FLAGS: 0x1, Stats(T): 0/0/0
  Up: 00:00:22
  RPF-ID: 1, Encap-ID: 0
  Incoming Interface List
    mdtvpn101 Flags: A MI, Up: 00:00:22
  Outgoing Interface List
    TenGigE0/0/0/6 Flags: F NS LI, Up: 00:00:22



Router# show mrib route detail
Wed Aug  9 15:00:03.092 UTC

IP Multicast Routing Information Base
Entry flags: L - Domain-Local Source, E - External Source to the Domain,
    C - Directly-Connected Check, S - Signal, IA - Inherit Accept,
    IF - Inherit From, D - Drop, ME - MDT Encap, EID - Encap ID,
    MD - MDT Decap, MT - MDT Threshold Crossed, MH - MDT interface handle
    CD - Conditional Decap, MPLS - MPLS Decap, EX - Extranet
    MoFE - MoFRR Enabled, MoFS - MoFRR State, MoFP - MoFRR Primary
    MoFB - MoFRR Backup, RPFID - RPF ID Set, X - VXLAN
Interface flags: F - Forward, A - Accept, IC - Internal Copy,
    NS - Negate Signal, DP - Don't Preserve, SP - Signal Present,
    II - Internal Interest, ID - Internal Disinterest, LI - Local Interest,
    LD - Local Disinterest, DI - Decapsulation Interface
    EI - Encapsulation Interface, MI - MDT Interface, LVIF - MPLS Encap,
```

```
    EX - Extranet, A2 - Secondary Accept, MT - MDT Threshold Crossed,
    MA - Data MDT Assigned, LMI - mLDP MDT Interface, TMI - P2MP-TE MDT Interface
    IRMI - IR MDT Interface, TRMI - TREE SID MDT Interface, MH - Multihome Interface

(*,233.252.0.4/24) Ver: 0x7f6e Flags: D P, MRID: 3, MCGID: 34, FLAGS: 0x0, Stats(F)
  Up: 00:06:54

(*,233.252.0.3) Ver: 0x7af2 Flags: S P, MRID: 1, MCGID: 32, FLAGS: 0x0, Stats(F)
  Up: 00:06:54

(*,233.252.0.1) Ver: 0x9f9b Flags: S P, MRID: 2, MCGID: 33, FLAGS: 0x0, Stats(F)
  Up: 00:06:54
  Outgoing Interface List
    TenGigE0/0/0/4 Flags: II LI, Up: 00:06:54

(*,233.252.0.2/8) Ver: 0x517d Flags: D P, MRID: 4, MCGID: 35, FLAGS: 0x0, Stats(F)
  Up: 00:06:54

(192.0.2.1,233.252.2.1) Ver: 0xc19e RPF nbr: 192.0.2.1 Flags: RPF ME MH, MRID: 9, MCGID:
134, FLAGS: 0x1, Stats(T): 0/0/0
  MVPN TID: 0xe0000002
  MVPN Remote TID: 0x0
  MVPN Payload: IPv4
  MDT IFH: 0x2000806c
  Up: 00:06:49
  RPF-ID: 1, Encap-ID: 0
  Incoming Interface List
    Loopback0 Flags: F A, Up: 00:06:49
  Outgoing Interface List
    Loopback0 Flags: F A, Up: 00:06:49
    TenGigE0/0/0/4 Flags: F NS, Up: 00:04:13

(192.0.2.21,233.252.2.1) Ver: 0xa354 RPF nbr: 192.0.2.2 Flags: RPF MD MH CD, MRID: 10,
MCGID: 135, FLAGS: 0x1, Stats(T): 0/0/1
  MVPN TID: 0xe0000002
  MVPN Remote TID: 0x0
  MVPN Payload: IPv4
  MDT IFH: 0x2000806c
  Up: 00:04:13
  RPF-ID: 1, Encap-ID: 0
  Incoming Interface List
    TenGigE0/0/0/4 Flags: A, Up: 00:04:13
  Outgoing Interface List
    Loopback0 Flags: F NS, Up: 00:04:13

(192.0.2.1,233.252.1.1) Ver: 0xbab2 RPF nbr: 192.0.2.1 Flags: RPF ME MH, MRID: 11, MCGID:
136, FLAGS: 0x1, Stats(T): 0/0/2
  MVPN TID: 0x0
  MVPN Remote TID: 0xe0800002
  MVPN Payload: IPv6
  MDT IFH: 0x2000806c
  Up: 00:06:49
  RPF-ID: 1, Encap-ID: 0
  Incoming Interface List
    Loopback0 Flags: F A, Up: 00:06:49
  Outgoing Interface List
    Loopback0 Flags: F A, Up: 00:06:49

(192.0.2.1,233.252.0.1) Ver: 0x86b RPF nbr: 192.0.2.1 Flags: RPF ME MH, MRID: 12, MCGID:
137, FLAGS: 0x1, Stats(T): 0/0/0
  MVPN TID: 0xe0000002
  MVPN Remote TID: 0x0
  MVPN Payload: IPv4
  MDT IFH: 0x2000806c
```

```
Up: 00:01:17
RPF-ID: 1, Encap-ID: 0
Incoming Interface List
  Loopback0 Flags: F A, Up: 00:01:17
Outgoing Interface List
  Loopback0 Flags: F A, Up: 00:01:17
  TenGigE0/0/0/4 Flags: F NS, Up: 00:01:17
```

Multicast Route Statistics

Table 20: Feature History Table

Feature Name	Release Information	Feature Description
Multicast Route Statistics enhancement on P100, Q100 and Q200 ASIC-based systems	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [LC ASIC: P100], 8700 [LC ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) Starting with this release, the route statistics counters are programmed on the ingress line card, making traffic measurement more efficient. The `stats-ole` counter-based implementation is activated when at least one of the Cisco Silicon One Q100 or Q200 ASIC-based systems, along with the Cisco Silicon One P100 ASIC-based systems, is operational on a router. When `stats-ole` counter-based implementation is active, the show mrib route detail command output shows `stats-ole` location; otherwise, it shows `INVALID`. *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM • 88-LC1-36EH

Feature Name	Release Information	Feature Description
Multicast Route Statistics	Release 7.3.1	When enabled, this feature provides statistics on the number of packets received for a multicast route. This information may be useful for monitoring and billing purposes.

Multicast route statistic provides information about the multicast routes. The multicast statistics information includes the rate at which packets are received and the number of packets received.

Cisco IOSXR Software counters are always present. To enable per-prefix counters only in hardware, use the **accounting per-prefix** command. When per-prefix counters are enabled, existing, and new (S, G) and (*, G) routes are assigned ingress counters, except for the following:

- Default multicast routes
- IPv4 (*, G) routes configured with prefix length less than 32.
- IPv6 (*, G) routes configured with prefix length less than 128.

If there is limited number of counters available and you want to enable counters on particular prefixes for troubleshooting purposes, you can configure **hw-module route-stats** to enable accounting for multicast routing for a limited number of routes.

For more information, see the **hw-module route-stats** command to configure a filter to choose which (S.G) routes have statistics enabled.

Restrictions

- Supports multicast route statistics for ingress direction only.

Configuring multicast route stats

Perform the following to configure multicast route stats:

- Configure rate per route
- Enable per-prefix counters
- Create Access Control List
- Enable multicast route statistics on a particular prefix

Configuration Example

This example shows how to enable multicast route statistics for IPv4:

```
/*Configure rate per route*/
Router# configure
Router(config)# multicast-routing
Router(config-mcast)# address-family ipv4
Router(config-mcast-default-ipv4)# rate-per-route

/*Enable per-prefix counters*/
Router# configure
Router(config)# multicast-routing
```

```
Router(config-mcast)# address-family ipv4
Router(config-mcast-default-ipv4)# accounting per-prefix

/*Create ACL*/

Router(config)# ipv4 access-list mcast-counter
Router(config-acl)# 10 permit ipv4 host 10.1.1.2 host 224.2.151.1
Router(config-acl)# 30 permit ipv4 10.1.1.0/24 232.0.4.0/22
Router(config-acl)# 50 permit ipv4 192.168.0.0/24 232.0.4.0/22
Router(config-acl)#  commit
Router(config-acl)# exit

/*Enable multicast route statistics on a particular prefixe*\

Router(config)# hw-module route-stats l3mcast vrf default ipv4 mcast-counter
```

Similarly, you can enable route statistics for IPv6 address:

```
/*Configure rate per route*/
Router# configure
Router(config)# multicast-routing
Router(config-mcast)# address-family ipv6
Router(config-mcast-default-ipv4)# rate-per-route

/*Enable per-prefix counters*/
Router# configure
Router(config)# multicast-routing
Router(config-mcast)# address-family ipv6
Router(config-mcast-default-ipv4)# accounting per-prefix

/*Create ACL*/
Router# configure
Router(config)# ipv6 access-list mcast-counter
Router(config-ipv6-acl)# 10 permit ipv6 any any
Router(config-ipv6-acl)# 20 deny udp any any
Router(config-ipv6-acl)# commit
Router(config-ipv6-acl)# exit

/*Enable multicast route statistics on a particular prefixe*\

Router(config)# hw-module route-stats l3mcast vrf default ipv6 mcast-counter
```

Verification

```
Router# show mfib route statistics location 0/RP0/CPU0
Thu Aug 13 19:16:58.321 UTC

IP Multicast Forwarding Rates
(Source Address, Group Address)
Incoming rate:
Node: (Incoming node) : pps/bps
Outgoing rate:
Node: (Outgoing node) : pps/bps

(192.168.0.0,232.0.4.0)
Incoming rate :
Node : 0/RP0/CPU0 : 749 / 1007969
Outgoing rate :
Node : 0/RP0/CPU0 : 0 / 0
RP/0/RP0/CPU0:ios#
```

To clear the Multicast Forwarding Information Base (MFIB) route packet hardware counters, use the **clear mfib platform route statistics** command.

Note To clear an ingress statistics of a route, you can get the `stats-ole` location for a specified route using the **show mrib route detail** command.

A `stats-ole` is programmed on one of the line cards for a particular route and helps report ingress statistics for a particular route.

If you know the `stats-ole` location, you can clear the ingress counters for a route on that location. If you do not know the `stats-ole` location, you can use the option `location all` instead, which helps to find the specific `stats-ole` location and clear the ingress counters.

This example shows how to find the `stats-ole` location:

```
Router # show mrib vrf vrf15 route 18.18.15.2 225.0.0.1 detail

IP Multicast Routing Information Base
Entry flags: L - Domain-Local Source, E - External Source to the Domain,
    C - Directly-Connected Check, S - Signal, IA - Inherit Accept,
    IF - Inherit From, D - Drop, MA - MDT Address, ME - MDT Encap,
    MD - MDT Decap, MT - MDT Threshold Crossed, MH - MDT interface handle
    CD - Conditional Decap, MPLS - MPLS Decap, MF - MPLS Encap, EX - Extranet
Interface flags: F - Forward, A - Accept, IC - Internal Copy,
    NS - Negate Signal, DP - Don't Preserve, SP - Signal Present,
    II - Internal Interest, ID - Internal Disinterest, LI - Local Interest,
    LD - Local Disinterest, DI - Decapsulation Interface
    EI - Encapsulation Interface, MI - MDT Interface, LVIF - MPLS Encap,
    EX - Extranet

18.18.15.2 225.0.0.1) Ver: 0x4df RPF nbr: 18.18.15.2 Flags: RPF, MRID: 60638, MCGID: 61036,

Stats T [R/S/I]: 0/11/0  /* 0/11/CPU0 is the stats-ole location. */
    Up: 01:45:14
     Incoming Interface List
                          Bundle-Ether43.80 Flags: A, Up: 01:45:14
     Outgoing Interface List
                          HundredGigE0/3/0/22.180 Flags: F NS, Up: 01:45:14
```

From the earlier example, you know that `stats-ole` location is `0/11/CPU0`. You can now clear the ingress stats using `0/11/CPU0` location.

```
Router# clear mfib platform route statistics 192.0.2.1 225.0.0.1 location 0/11/CPU0
```

Starting Cisco IOS XR Release 24.4.1, the route statistics counter is programmed on the ingress line card for the Cisco Silicon One P100 ASIC-based Systems. As traffic enters the ingress line card, the packet and byte counters will increment.

The `stats-ole` counter-based implementation is activated when at least one of the Cisco Silicon One Q100 or Q200 ASIC-based systems, along with the Cisco Silicon One P100 ASIC-based systems, is operational on a router.

When `stats-ole` counter-based implementation is active, the **show mrib route detail** command output shows the `stats-ole` location; otherwise, it shows `INVALID`.

Use the **clear mfib platform route statistics** command to clear the counters by specifying the ingress line card location.

This example shows the truncated output for the **show mrib route detail** command on the Cisco Silicon One P100 ASIC-based Systems:

```
Router # show mrib vrf vpn101 route detail
IP Multicast Routing Information Base
```

```
Entry flags:
- L - Domain-Local Source
- E - External Source to the Domain
- C - Directly-Connected Check
.
.
.

Interface flags:
- F - Forward
- A - Accept
- IC - Internal Copy
.
.
.

(*,224.0.0.0/4) Ver: 0x6119 RPF nbr: 198.51.100.1 Flags: L C RPF P, MRID: 9, MCGID: 40,
FLAGS: 0x0, Stats(F)
  Up: 00:03:12
  Outgoing Interface List
    Decapstunnel0 Flags: NS DI, Up: 00:03:07
(*,224.0.0.0/24) Ver: 0xc1d8 Flags: D P, MRID: 7, MCGID: 38, FLAGS: 0x0, Stats(F)
  Up: 00:03:12
(*,224.0.1.39) Ver: 0x653 Flags: S P, MRID: 5, MCGID: 36, FLAGS: 0x0, Stats(F)
  Up: 00:03:12
(*,224.0.1.40) Ver: 0xde67 Flags: S P, MRID: 6, MCGID: 37, FLAGS: 0x0, Stats(F)
  Up: 00:03:12
  Outgoing Interface List
    HundredGigE0/0/0/14 Flags: II LI, Up: 00:03:12
(*,232.0.0.0/8) Ver: 0x665b Flags: D P, MRID: 8, MCGID: 39, FLAGS: 0x0, Stats(F)
  Up: 00:03:12
(198.51.100.2,232.0.0.1) Ver: 0x710e RPF nbr: 203.0.113.1 Flags: RPF RPFID, MRID: 10, MCGID:
 136, FLAGS: 0x1, Stats(T): INVALID
  Up: 00:03:05
  RPF-ID: 1, Encap-ID: 0
  Incoming Interface List
    Lmdtvpn101 Flags: A LMI, Up: 00:02:43
  Outgoing Interface List
    HundredGigE0/0/0/14 Flags: F NS LI, Up: 00:03:05
```

MVPN Ingress Replication Over Dynamic TE-Tunnels

Table 21: Feature History Table

Feature Name	Release Information	Feature Description
MVPN Ingress Replication over Dynamic TE (MVPN IRoTE) Tunne-ls	Release 24.4.1	Introduced in this release on: Fixed Systems (8700 [ASIC:K100]) This feature support is now extended to the Cisco 8712-MOD-M routers.

Feature Name	Release Information	Feature Description

MVPN Ingress Replication over Dynamic TE (MVPN IRoTE) Tunne-ls	Release 24.1.1	MVPN Ingress replication over dynamic-TE tunnels enables the routing of multicast traffic through an MPLS network using RSVP-TE P2MP (point-to-multipoint) tunnels. The traffic is replicated by the ingress router before sending it to the destination devices through the TE tunnels which are created dynamically.
		When configured, this feature enables utilization of the TE tunnels for transmission of multicast traffic and ensures inter-operability with other devices that are configured with this feature in the network.
		The feature introduces these changes:
		• The **ingress replication** keywords are introduced in the following commands:
		  • mdt data
		  • mdt default
		  • mdt partitioned

Multicast VPN (MVPN) ingress replication (IR) when configured is a feature that optimizes the distribution of multicast traffic. This advanced profile leverages traffic engineering (TE) within the network underlay to maximize efficiency and performance.

One of the key benefits of this feature is that user multicast traffic is replicated at the headend of the ingress router, where the original multicast packets are replicated and sent as unicast packets to receiver nodes, or leaf nodes, via TE tunnels. This helps in streamlining traffic management by effective use of the TE tunnels and interoperability with other devices configured with this feature. Additionally, TE Fast Reroute (FRR) protection in the core enhances the network's resilience to faults and improves overall reliability. This also ensures the core network remains free of multicast traffic.

The ingress replication feature is an extension of existing MVPN profiles, such as profile 19 and 21, and is not considered a new profile. Instead, it builds upon profile 22, the P2MP TE profile, enhancing it with ingress replication capabilities.

Limitations and User Guidelines

- Sparse Mode (SM) in overlay is not supported.
- The maximum number of tunnels that can be configured is 6000.