



Multicast Configuration Guide for Cisco NCS 5500 Series Routers, IOS XR Release 6.6.x

First Published: 2019-05-30

Last Modified: 2020-01-10

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



CONTENTS

PREFACE

Preface ix

Changes to This Document **ix**

Communications, Services, and Additional Information **ix**

CHAPTER 1

New and Changed Multicast Features 1

Multicast Features Added or Modified in IOS XR Release 6.6.x **1**

CHAPTER 2

Implementing Multicast 3

Implementing Layer-3 Multicast Routing **3**

Enabling Multicast **4**

Protocol Independent Multicast **5**

PIM BFD Overview **5**

Configure PIM BFD **6**

Verification **7**

Reverse Path Forwarding **7**

Setting the Reverse Path Forwarding Statically **8**

RPF Vector Encoding Using IETF Standard **8**

Configuring RPF Vector (IETF Standard Encoding) **8**

PIM Bootstrap Router **9**

Configuring PIM Bootstrap Router **9**

PIM-Source Specific Multicast **10**

IGMPv2 **11**

Multipath Option **12**

Configuring PIM-SSM **14**

Configuring PIM Parameters **15**

Multicast Source Discovery Protocol **15**

Interconnecting PIM-SM Domains with MSDP	16
Controlling Source Information on MSDP Peer Routers	19
PIM-Sparse Mode	21
Designated Routers	22
Internet Group Management Protocol	24
Configuring Maximum IGMP Per Interface Group Limit	25
SSM Static Source Mapping	26
Information About IGMP Snooping Configuration Profiles	28
Creating Profiles	28
Attaching and Detaching Profiles	28
Changing Profiles	29
Configuring Access Control	29
IPv6 Multicast for Multiple Sources	30
Statistics for Ingress Multicast Routes	30
Configuring Statistics for Ingress Multicast Routes	31
Multicast Route Statistics	32
Restrictions for Implementing Multicast Route Statistics Feature	33
Configure Multicast Route Statistics	33
Use Case: Video Streaming	35
Multicast Label Distribution Protocol (MLDP) for Core	36
Characteristics of MLDP Profiles on Core	36
Multicast VPN Support based on Point to Multipoint Traffic Engineering (P2MPE)	37
Configuration Example for MLDP on Core	43
Label Switched Multicast (LSM) Multicast Label Distribution Protocol (mLDP) based Multicast VPN (mVPN) Support	43
Benefits of LSM mLDP based MVPN	43
Configuring MLDP MVPN	43
Packet Flow in mLDP-based Multicast VPN	44
Realizing a mLDP-based Multicast VPN	44
Characteristics of mLDP Profiles	45
Configuration rules for profiles	45
MLDP inband signaling	45
Restrictions for mLDP on Edge Routers	45
Configuration Process for MLDP MVPN (Intranet)	45

Configure VRF MLDP In-Band Signaling on Edge Routers	47
Configure Global MLDP In-band Signaling on Edge Routers	48
Configuration Examples for MVPN Profiles	49
Configuration Examples for Inband mLDP Profiles on Core Routers	49
Configuration Examples for Inband mLDP Profiles on Edge Routers	50
Verification of MLDP Configuration on Edge Routers	51
MLDP Loop-Free Alternative Fast Reroute	52
Advantages of LFA FRR	53
MLDP LFA FRR - Features	54
Limitations of LFA FRR	54
MLDP LFA FRR - Working	55
MLDP LFA FRR - Behavior	56
Configurations to Enable LFA FRR	58
Configuring Router OSPF LFA FRR	58
Configuring Router ISIS LFA FRR	64
Configuring Bidirectional Forwarding Detection	65
Configuring MPLS LFA FRR	68
<hr/>	
CHAPTER 3	Implementing Layer-2 Multicast with IGMP Snooping
	73
Implementing Layer-2 Multicast with IGMP Snooping	73
Prerequisites for IGMP Snooping	73
Restrictions for IGMP Snooping	73
Information About IGMP Snooping	74
IGMP Snooping Overview	74
Description of Basic Functions	74
High Availability Features	75
Bridge Domain Support	75
Multicast Router Port	75
Multicast Host Ports	75
Multicast Traffic Handling within a Bridge Domain with IGMP Snooping Enabled	75
Information About IGMP Snooping Configuration Profiles	76
Creating Profiles	77
Attaching and Detaching Profiles	77
Changing Profiles	77

Default IGMP Snooping Configuration Settings	78
IGMP Snooping Configuration at the Bridge Domain Level	78
IGMP Minimum Version	78
Group Membership Interval, Robustness Variable, and Query Interval	79
How to Configure IGMP Snooping	79
Creating an IGMP Snooping Profile	79
Where to Go Next	80
Attaching a Profile and Activating IGMP Snooping on a Bridge Domain	81
Detaching a Profile and Deactivating IGMP Snooping on a Bridge Domain	82
Attaching and Detaching Profiles to Ports Under a Bridge	83
Verifying Multicast Forwarding	85
Configuration Examples for IGMP Snooping	86
Configuring IGMP Snooping on Physical Interfaces Under a Bridge: Example	86
Configuring IGMP Snooping on VLAN Interfaces Under a Bridge: Example	87
Configuring IGMP Snooping on Ethernet Bundles Under a Bridge: Example	88
Configuring Multicast over Integrated Routing Bridging Active/Active Multihome	89
Verifying IGMP Snooping and EVPN Sync	91
Additional References	92

CHAPTER 4**Implementing MLD Snooping 95**

MLD Snooping	95
Prerequisites for MLD Snooping	96
Restrictions for MLD Snooping	96
Advantages of MLD Snooping	96
High Availability (HA) features for MLD	96
Bridge Domain Support for MLD	97
Multicast Router and Host Ports	97
Multicast Router Discovery for MLD	97
Multicast Traffic Handling for MLD	97
Multicast Listener Discovery over BVI	99
IPv6 Multicast Listener Discovery Snooping over BVI	99
Configuring Internal Querier for MLD Snooping	100
Creating a MLD Snooping Profile	100
Deactivating MLD Snooping on a Bridge Domain	101

Configuring Static Mrouter Ports (MLD)	102
Configuring Router Guard (MLD)	102
Configuring Immediate-leave for MLD	103
Configuring Internal Querier for MLD	104
Configuring Static Groups for MLD	105
Configuring MLD Snooping	107
Configuring MLD Snooping on Ethernet Bundles	108



Preface

This preface contains these sections:

- [Changes to This Document](#) , on page ix
- [Communications, Services, and Additional Information](#), on page ix

Changes to This Document



Note *This software release has reached end-of-life status. For more information, see the [End-of-Life and End-of-Sale Notices](#).*

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

Table 1: Changes to This Document

Date	Summary
December 2018	Initial release of this document.
May 2019	Republished for Release 6.6.2
December 2019	Republished for Release 6.6.3

Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at [Cisco Profile Manager](#).
- To get the business impact 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 Marketplace](#).
- 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 6.6.x, on page 1](#)

Multicast Features Added or Modified in IOS XR Release 6.6.x

Table 2: New and Changed Features

Feature	Description	Changed in Release	Where Documented
LFA Based FRR and Remote LFA for MLDP	LFA Based FRR and Remote LFA for MLDP	Release 6.6.3	Implementing Multicast
Multicast MLDP Profile 14 support on an Edge Router	The MLDP Profile 14 is supported when the router is configured as an edge router.	Release 6.6.3	Multicast MLDP Profile 14 support on an Edge Router
Multicast Route Statistics	Multicast route statistic feature provides information about the IPv4 multicast routes.	Release 6.6.25	Multicast Route Statistics, on page 32
IPv6 Multicast for Multiple Sources	From Release 6.6.25 onwards, IPv6 multicast supports multiple sources for a single multicast group.	Release 6.6.25	IPv6 Multicast for Multiple Sources, on page 30
EVPN: Dual PIM-DR (EVPN) – IPv4 Includes PIM-SM for Uplink – IPv4	This feature was introduced.	Release 6.6.25	Implementing Layer-2 Multicast with IGMP Snooping, on page 73
EVPN: IPv4 MC Enhanced DF election for MCAST	This feature was introduced.	Release 6.6.25	Implementing Layer-2 Multicast with IGMP Snooping, on page 73

Feature	Description	Changed in Release	Where Documented
IGMP (V2 only) Snooping on Layer 2 Transport – A/A MH (Receiver only) – PIM-ASM	This feature was introduced.	Release 6.6.25	Implementing Layer-2 Multicast with IGMP Snooping, on page 73
IGMP Layer 2 EVPN State Synchronization	This feature was introduced.	Release 6.6.25	Implementing Layer-2 Multicast with IGMP Snooping, on page 73



CHAPTER 2

Implementing Multicast

- [Implementing Layer-3 Multicast Routing](#), on page 3
- [Protocol Independent Multicast](#), on page 5
- [PIM BFD Overview](#), on page 5
- [Reverse Path Forwarding](#), on page 7
- [RPF Vector Encoding Using IETF Standard](#), on page 8
- [PIM Bootstrap Router](#), on page 9
- [PIM-Source Specific Multicast](#), on page 10
- [Multicast Source Discovery Protocol](#), on page 15
- [PIM-Sparse Mode](#), on page 21
- [Designated Routers](#), on page 22
- [Internet Group Management Protocol](#), on page 24
- [IPv6 Multicast for Multiple Sources](#), on page 30
- [Statistics for Ingress Multicast Routes](#), on page 30
- [Multicast Route Statistics](#), on page 32
- [Use Case: Video Streaming](#), on page 35
- [Multicast Label Distribution Protocol \(MLDP\) for Core](#), on page 36
- [Multicast VPN Support based on Point to Multipoint Traffic Engineering \(P2MPE\)](#), on page 37
- [Configuration Example for MLDP on Core](#), on page 43
- [Label Switched Multicast \(LSM\) Multicast Label Distribution Protocol \(mLDP\) based Multicast VPN \(mVPN\) Support](#), on page 43
- [Configure VRF MLDP In-Band Signaling on Edge Routers](#), on page 47
- [Configure Global MLDP In-band Signaling on Edge Routers](#), on page 48
- [Configuration Examples for MVPN Profiles](#), on page 49
- [Configuration Examples for Inband mLDP Profiles on Edge Routers](#), on page 50
- [Verification of MLDP Configuration on Edge Routers](#), on page 51
- [MLDP Loop-Free Alternative Fast Reroute](#), on page 52

Implementing Layer-3 Multicast Routing

Multicast routing allows a host to send packets to a subset of all hosts as a group transmission rather than to a single host, as in unicast transmission, or to all hosts, as in broadcast transmission. The subset of hosts is known as group 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.

The multicast environment consists of senders and receivers. Any host, regardless of whether it is a member of a group, can send to a group. However, only the members of a group receive the message.

The following protocols are supported to implement multicast routing:

- IGMP—IGMP is used between hosts on a network (for example, LAN) and the routers on that network to track the multicast groups of which hosts are members.
- PIM SSM— Protocol Independent Multicast in Source-Specific Multicast (PIM-SSM) has the ability to report interest in receiving packets from specific source addresses (or from all but the specific source addresses), to an IP multicast address.



Note MLD Snooping is not supported until Cisco IOS XR Release 6.5.3.

Prerequisites for Implementing Multicast Routing

- You must install and activate the multicast RPM package.
- You must be familiar with IPv4 multicast routing configuration tasks and concepts.
- Unicast routing must be operational.

Restrictions

- Multicast over access or core pseudowire is not supported.

Enabling Multicast

Configuration Example

Enables multicast routing and forwarding on all new and existing interfaces.

```
Router#config
Router(config)#multicast-routing
Router(config-mcast)#address-family ipv4
Router(config-mcast-default-ipv4)#interface all enable
*/In the above command, you can also indicate a specific interface (For example, interface
TenGigE0/0/0/3)
for enabling multicast only on that interface/*
Router(config-mcast-default-ipv4)#commit
```

Running Configuration

```
Router#show running multicast routing
multicast-routing
  address-family ipv4
    interface all enable
  !
```

Verification

Verify that the Interfaces are enabled for multicast.

```
Router#show mfib interface location 0/3/CPU0
Interface : FINT0/3/CPU0 (Enabled)
SW Mcast pkts in : 0, SW Mcast pkts out : 0
TTL Threshold : 0
Ref Count : 2
Interface : TenGigE0/3/0/0/0 (Enabled)
SW Mcast pkts in : 0, SW Mcast pkts out : 0
TTL Threshold : 0
Ref Count : 3
Interface : TenGigE0/3/0/9/0 (Enabled)
SW Mcast pkts in : 0, SW Mcast pkts out : 0
TTL Threshold : 0
Ref Count : 13
Interface : Bundle-Ether1 (Enabled)
SW Mcast pkts in : 0, SW Mcast pkts out : 0
TTL Threshold : 0
Ref Count : 4
Interface : Bundle-Ether1.1 (Enabled)
SW Mcast pkts in : 0, SW Mcast pkts out : 0
TTL Threshold : 0
```

Protocol Independent Multicast

Protocol Independent Multicast (PIM) is a multicast routing protocol used to create multicast distribution trees, which are used to forward multicast data packets.

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. Protocol Independent Multicast (PIM) is designed to send and receive multicast routing updates.

NCS 5500 series router supports Protocol Independent Multicast in Source-Specific Multicast (PIM-SSM).

PIM on Bundle-Ethernet subinterface is supported.

PIM BFD Overview

The BFD Support for Multicast (PIM) feature, also known as PIM BFD, registers PIM as a client of BFD. PIM can then utilize BFD's fast adjacency failure detection. When PIM BFD is enabled, BFD enables faster failure detection without waiting for hello messages from PIM.

At PIMs request, as a BFD client, BFD establishes and maintains a session with an adjacent node for maintaining liveness and detecting forwarding path failure to the adjacent node. PIM hellos will continue to be exchanged between the neighbors even after BFD establishes and maintains a BFD session with the neighbor. The behavior of the PIM hello mechanism is not altered due to the introduction of this feature. Although PIM depends on the Interior Gateway Protocol (IGP) and BFD is supported in IGP, PIM BFD is independent of IGP's BFD.

Protocol Independent Multicast (PIM) uses a hello mechanism for discovering new PIM neighbors between adjacent nodes. The minimum failure detection time in PIM is 3 times the PIM Query-Interval. To enable faster failure detection, the rate at which a PIM hello message is transmitted on an interface is configurable. However, lower intervals increase the load on the protocol and can increase CPU and memory utilization and cause a system-wide negative impact on performance. Lower intervals can also cause PIM neighbors to expire frequently as the neighbor expiry can occur before the hello messages received from those neighbors are

processed. When PIM BFD is enabled, BFD enables faster failure detection without waiting for hello messages from PIM.

Configure PIM BFD

This section describes how you can configure PIM BFD

```
Router# configure
Router(config)# router pim address-family ipv4
Router(config-pim-default-ipv4)# interface HundredGigE0/1/0/1
Router(config-pim-ipv4-if)# bfd minimum-interval 10
Router(config-pim-ipv4-if)# bfd fast-detect
Router(config-pim-ipv4-if)# bfd multiplier 3
Router(config-pim-ipv4)# exit
Router(config-pim-default-ipv4)# interface TenGigE0/0/0/4
Router(config-pim-ipv4-if)# bfd minimum-interval 50
Router(config-pim-ipv4-if)# bfd fast-detect
Router(config-pim-ipv4-if)# bfd multiplier 3
Router(config-pim-ipv4-if)# exit
Router(config-pim-default-ipv4)# interface TenGigE 0/0/0/4.101
Router(config-pim-ipv4-if)# bfd minimum-interval 50
Router(config-pim-ipv4-if)# bfd fast-detect
Router(config-pim-ipv4-if)# bfd multiplier 3
Router(config-pim-ipv4-if)# exit
Router(config-pim-default-ipv4)# interface Bundle-Ether 101
Router(config-pim-ipv4-if)# bfd minimum-interval 50
Router(config-pim-ipv4-if)# bfd fast-detect
Router(config-pim-ipv4-if)# bfd multiplier 3
Router(config-pim-ipv4-if)# exit
Router(config-pim-default-ipv4)# commit
```

Running Configuration

```
router pim
 address-family ipv4
  interface HundredGigE 0/1/0/1
   bfd minimum-interval 10
   bfd fast-detect
   bfd multiplier 3
  !
  interface TenGigE 0/0/0/4
   bfd minimum-interval 50
   bfd fast-detect
   bfd multiplier 3
  !
  interface TenGigE 0/0/0/4.101
   bfd minimum-interval 50
   bfd fast-detect
   bfd multiplier 3
  !
  interface Bundle-Ether 101
   bfd minimum-interval 50
   bfd fast-detect
   bfd multiplier 3
  !
 !
 !
```


!

Verification

The show outputs given in the following section display the details of the configuration of the PIM BFD, and the status of their configuration.

```
Router# show bfd session
Wed Nov 22 08:27:35.952 PST
Interface          Dest Addr      Local det time(int*mult)  State      Echo      Async
H/W              NPU
-----
Hu0/0/1/3         10.12.12.2     0s (0s*0) 90ms (30ms*3)         UP          Yes
0/0/CPU0
Hu0/0/1/2         10.12.12.2     0s (0s*0) 90ms (30ms*3)         UP          Yes
0/0/CPU0
Hu0/0/1/1         10.18.18.2     0s (0s*0) 90ms (30ms*3)         UP          Yes
0/0/CPU0
Te0/0/0/4.101    10.112.112.2   0s (0s*0) 90ms (30ms*3)         UP          Yes
0/0/CPU0
BE101             10.18.18.2     n/a      n/a                    UP          No      n/a
BE102             10.12.12.2     n/a      n/a                    UP          No      n/a
```

```
Router# show bfd client
Name              Node              Num sessions
-----
L2VPN_ATOM       0/RP0/CPU0 0
MPLS-TR          0/RP0/CPU0 0
bgp-default      0/RP0/CPU0 0
bundlemgr_distrib 0/RP0/CPU0 14
isis-1           0/RP0/CPU0 0
object_tracking  0/RP0/CPU0 0
pim6             0/RP0/CPU0 0
pim             0/RP0/CPU0 0
service-layer    0/RP0/CPU0 0
```

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.

Setting the Reverse Path Forwarding Statically

Configuration Example

The following example configures the static RPF rule for IP address 10.0.0.1:

```
Router#configure
Router(config)#multicast-routing
Router(config-mcast)#address-family ipv4
Router(config-mcast)#static-rpf 10.0.0.1 32 TenGigE 0/0/0/1 192.168.0.2
Router(config-mcast)#commit
```

Running Configuration

```
multicast-routing
  address-family ipv4
    static-rpf 10.10.10.2 32 TenGigE0/0/0/1 192.168.0.2
```

Verification

Verify that RPF is chosen according to the static RPF configuration for 10.10.10.2

```
Router#show pim rpf
Table: IPv4-Unicast-default
* 10.10.10.2/32 [0/0]
  via GigabitEthernet0/0/0/1 with rpf neighbor 192.168.0.2
```

RPF Vector Encoding Using IETF Standard

RPF vector is a PIM proxy that lets core routers without RPF information forward join and prune messages for external sources (for example, a MPLS-based BGP-free core, where the MPLS core router is without external routes learned from BGP). The RPF vector encoding is now compatible with the new IETF encoding. The new IETF standard encodes PIM messages using PIM Hello option 26.

Configuring RPF Vector (IETF Standard Encoding)

This example shows how to enable RPF encoding using IETF standard:

```
(config)# router pim
(config-pim-default-ipv4)# address-family ipv4
```

```
(config-pim-default-ipv4)# rpf-vector use-standard-encoding
!
(config)# multicast-routing
(config-mcast)# interface TenGigE0/1/0/7/0
(config-mcast)# interface TenGigE0/0/0/0/2
```

Verification

```
Router#show pim neighbor
Tue Apr 17 10:15:40.961 PDT
```

```
PIM neighbors in VRF default
Flag: B - Bidir capable, P - Proxy capable, DR - Designated Router,
      E - ECMP Redirect capable
      * indicates the neighbor created for this router
```

Neighbor Address	Interface	Uptime	Expires	DR pri	Flags
25.25.25.1	TenGigE0/1/0/7/0	1w3d	00:01:36	1	B P
25.25.25.2*	TenGigE0/1/0/7/0	1w3d	00:01:41	1 (DR)	B P E
32.32.32.2*	TenGigE0/0/0/0/2	1w4d	00:01:40	1	B P E
32.32.32.3	TenGigE0/0/0/0/2	1w4d	00:01:42	1 (DR)	B P

In the above output, you can see "P" tag on the multicast enabled interfaces.

PIM Bootstrap Router

The PIM bootstrap router (BSR) provides a fault-tolerant, automated RP discovery and distribution mechanism that simplifies the Auto-RP process. This feature is enabled by default allowing routers to dynamically learn the group-to-RP mappings.

PIM uses the BSR to discover and announce RP-set information for each group prefix to all the routers in a PIM domain. This is the same function accomplished by Auto-RP, but the BSR is part of the PIM specification. The BSR mechanism interoperates with Auto-RP on Cisco routers.

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.

Configuring PIM Bootstrap Router

Configuration Example

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

```
Router#config
Router(config)#router pim
```

```
Router(config-pim-default-ipv4)#bsr candidate-bsr 1.1.1.1 hash-mask-len 30 priority 1
Router(config-pim-default-ipv4-if)#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 1.1.1.1. This rendezvous point is responsible for the groups with the prefix 239.

```
Router#config
Router(config)#router pim
Router(config-pim-default-ipv4)#bsr candidate-rp 1.1.1.1 group-list 4 priority 192 interval
60
```

```
Router(config-pim-default-ipv4)#exit
Router(config)#ipv4 access-list 4
Router(config-ipv4-acl)#permit ipv4 any 239.0.0.0 0.255.255.255
Router(config-ipv4-acl)#commit
```

Running Configuration

```
Router#show run router pim
router pim
 address-family ipv4
   bsr candidate-bsr 1.1.1.1 hash-mask-len 30 priority 1
   bsr candidate-rp 1.1.1.1 group-list 4 priority 192 interval 60
```

Verification

```
Router#show pim rp mapping
PIM Group-to-RP Mappings
Group(s) 239.0.0.0/8
  RP 1.1.1.1 (?), v2
    Info source: 1.1.1.1 (?), elected via bsr, priority 192, holdtime 150
    Uptime: 00:02:50, expires: 00:01:54
```

```
Router#show pim bsr candidate-rp
PIM BSR Candidate RP Info
Cand-RP      mode  scope  priority  uptime      group-list
1.1.1.1      BD    16     192      00:04:06    4
```

```
Router#show pim bsr election
PIM BSR Election State
Cand/Elect-State      Uptime  BS-Timer  BSR                                     C-BSR
Elected/Accept-Pref  00:03:49 00:00:25 1.1.1.1 [1, 30]                       1.1.1.1 [1, 30]
```

PIM-Source Specific Multicast

When PIM is used in SSM mode, multicast 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 and FF3x::/32 for IPv6. To configure these values, use the **ssm range** command.



Note PIM-SSM supports IPv6 from Cisco IOS XR Release 6.2.2

- 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 broadcast traffic. The source is provided by interested receivers through IGMPv3 membership reports.

IGMPv2

To support IGMPv2, SSM mapping configuration must be added while configuring IGMP to match certain sources to group range.

Configuring Example

Configures the access-list (mcl):

```
Router#configure
Router(config)#ipv4 access-list mcl
Router(config-ipv4-acl)#permit ipv4 any 232.1.1.0 0.0.0.255
Router(config-ipv4-acl)#commit
```

Configures the multicast source (1.1.1.1) as part of a set of sources that map SSM groups described by the specified access-list (mc1):

```
Router#configure
Router(config)#router igmp
Router(config-igmp)#ssm map static 1.1.1.1 mc1
Router(config-igmp)#commit
```

Running Configuration

```
Router#show run router igmp
router igmp
ssm map static 1.1.1.1 mc1
```

Multipath Option

The multipath option is available under `router pim` configuration mode. After multipath option is enabled, SSM selects different path to reach same destination instead of choosing common path. The multipath option helps load balance the SSM traffic.

Configuring Multipath Option

```
Router#configure
Router(config)#router pim address-family ipv4
Router(config-pim-default-ipv4)#multipath hash source
Router(config-pim-default-ipv4)#commit
```

Running Configuration

```
Router#show running router pim
router pim
  address-family ipv4
    dr-priority 100
    multipath hash source /*SSM traffic takes different path to reach same destination
based on source hash value.*/
```

Verification

The Bundle-Ether132 and TenGigE0/4/0/18/0.132 are two paths to reach the destination router Turin-56. Since we have enabled multipath option, the source has two IP addresses 50.11.30.12 and 50.11.30.11. The Multicast traffic from two sources take two different paths Bundle-Ether132 and TenGigE0/4/0/18/0.132 to reach same destination.

This show run output shows that Bundle-Ether132 and TenGigE0/4/0/18/0.132 are connected to same destination router Turin-56:

```
Router#show run int TenGigE0/1/0/6/3.132
interface TenGigE0/1/0/6/3.132
  description Connected to Turin-56 ten0/0/0/19.132
  ipv4 address 13.0.2.1 255.255.255.240
  ipv6 address 2606::13:0:2:1/120
  encapsulation dot1q 132
!
```

```
Router#show run int be132
interface Bundle-Ether132
  description Bundle between Fretta-56 and Turin-56
  ipv4 address 28.0.0.1 255.255.255.240
```

```

ipv6 address 2606::28:0:0:1/120
load-interval 30

```

```
Router#show mrib route 50.11.30.11 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, 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

```

```
(50.11.30.11,225.255.11.1) Ver: 0x523cc294 RPF nbr: 50.11.30.11 Flags: L RPF, FGID: 11453,
-1, -1
```

```
Up: 4d15h
```

```
Incoming Interface List
```

```
HundredGigE0/4/0/10.1130 Flags: A, Up: 4d15h
```

```
Outgoing Interface List
```

```
FortyGigE0/1/0/5 Flags: F NS, Up: 4d15h
```

```
TenGigE0/4/0/6/0 Flags: F NS, Up: 4d15h
```

```
TenGigE0/1/0/6/3.132 Flags: F NS, Up: 4d15h
```

```
TenGigE0/4/0/18/0.122 Flags: F NS, Up: 4d15h
```

```
Router#show mrib route 50.11.30.12 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, 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

```

```
(50.11.30.12,226.255.12.1) Ver: 0x5fe02e5b RPF nbr: 50.11.30.12 Flags: L RPF, FGID: 12686,
-1, -1
```

```
Up: 4d15h
```

```
Incoming Interface List
```

```
HundredGigE0/4/0/10.1130 Flags: A, Up: 4d15h
```

```
Outgoing Interface List
```

```
Bundle-Ether121 Flags: F NS, Up: 4d15h
```

```
Bundle-Ether132 Flags: F NS, Up: 4d15h
```

```
FortyGigE0/1/0/5 Flags: F NS, Up: 4d15h
```

```
TenGigE0/4/0/6/0.117 Flags: F NS, Up: 4d15h
```

Configuring PIM-SSM

Configuration Example

Configures SSM service for the IPv4 address range defined by access list 4.

```
Router#config
Router(config)#ipv4 access-list 4
Router(config-ipv4-acl)#permit ipv4 any 224.2.151.0 0.0.0.255
Router(config-ipv4-acl)#exit
Router(config)#multicast-routing
Router(config-mcast)#address-family ipv4
Router(config-mcast-default-ipv4)#ssm range 4
Router(config-mcast-default-ipv4)#commit
Router(config-mcast-default-ipv4)#end
```

Configures SSM service for the IPv6 address range defined by access list 6.

```
Router#config
Router(config)#ipv6 access-list 6
Router(config-ipv6-acl)#permit ipv6 any ff30:0:0:2::/32
Router(config-ipv6-acl)#exit
Router(config)#multicast-routing
Router(config-mcast)#address-family ipv6
Router(config-mcast-default-ipv6)#ssm range 6
Router(config-mcast-default-ipv6)#commit
Router(config-mcast-default-ipv6)#end
```

Running Configuration

```
Router#show running multicast-routing
multicast-routing
  address-family ipv4
    ssm range 4
  interface all enable
!
```

```
Router#show running multicast-routing
multicast-routing
  address-family ipv6
    ssm range 6
  interface all enable
!
```

Verification

Verify if the SSM range is configured according to the set parameters:

```
Router#show access-lists 4
ipv4 access-list 4
  10 permit ipv4 any 224.2.151.0 0.0.0.255
```

/Verify if the SSM is configured for 224.2.151.0/24/:

```
Router#show pim group-map
IP PIM Group Mapping Table
(* indicates group mappings being used)
Group Range      Proto Client  Groups RP address  Info
224.0.1.39/32*  DM    perm     1     0.0.0.0
224.0.1.40/32*  DM    perm     1     0.0.0.0
224.0.0.0/24*   NO    perm     0     0.0.0.0
224.2.151.0/24* SSM   config   0     0.0.0.0
```


Configuring PIM Parameters

To configure PIM-specific parameters, the router pim configuration mode is used. The default configuration prompt is for IPv4 and will be seen as config-pim-default-ipv4. To ensure the election of a router as PIM DR on a LAN segment, use the **dr-priority** command. The router with the highest DR priority will win the election. By default, at a preconfigured threshold, the last hop router can join the shortest path tree to receive multicast traffic. To change this behavior, use the command **spt-threshold infinity** under the router pim configuration mode. This will result in the last hop router permanently joining the shared tree. The frequency at which a router sends PIM hello messages to its neighbors can be configured by the hello-interval command. By default, PIM hello messages are sent once every 30 seconds. If the hello-interval is configured under router pim configuration mode, all the interfaces with PIM enabled will inherit this value. To change the hello interval on the interface, use the **hello-interval** command under interface configuration mode, as follows:

Configuration Example

```
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)#spt-threshold infinity
Router(config-pim-default-ipv4)#interface TenGigE0/0/0/1
Router(config-pim-ipv4-if)#dr-priority 4
Router(config-pim-ipv4-if)#hello-interval 45
Router(config-pim-ipv4-if)#commit
```

Running Configuration

```
Router#show run router pim
router pim
  address-family ipv4
    dr-priority 2
    spt-threshold infinity
  interface TenGigE0/0/0/1
    dr-priority 4
    hello-interval 45
```

Verification

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

```
Router#show pim interface te0/0/0/1
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
```

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.

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.

Interconnecting PIM-SM Domains with MSDP

To set up an MSDP peering relationship with MSDP-enabled routers in another domain, you configure an MSDP peer to the local router.

If you do not want to have or cannot have a BGP peer in your domain, you could define a default MSDP peer from which to accept all Source-Active (SA) messages.

Finally, you can change the Originator ID when you configure a logical RP on multiple routers in an MSDP mesh group.

Before you begin

You must configure MSDP default peering, if the addresses of all MSDP peers are not known in BGP or multiprotocol BGP.

SUMMARY STEPS

1. **configure**
2. **interface** *type interface-path-id*
3. **ipv4 address** *address mask*
4. **exit**
5. **router msdp**
6. **default-peer** *ip-address* [**prefix-list** *list*]
7. **originator-id** *type interface-path-id*
8. **peer** *peer-address*
9. **connect-source** *type interface-path-id*
10. **mesh-group** *name*
11. **remote-as** *as-number*
12. **commit**
13. **show msdp** [**ipv4**] **globals**

14. `show msdp [ipv4] peer [peer-address]`
15. `show msdp [ipv4] rpf rpf-address`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>configure</code>	
Step 2	<p><code>interface type interface-path-id</code></p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config)# interface loopback 0</pre>	<p>(Optional) Enters interface configuration mode to define the IPv4 address for the interface.</p> <p>Note This step is required if you specify an interface type and number whose primary address becomes the source IP address for the TCP connection.</p>
Step 3	<p><code>ipv4 address address mask</code></p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.0.1.3 255.255.255.0</pre>	<p>(Optional) Defines the IPv4 address for the interface.</p> <p>Note This step is required only if you specify an interface type and number whose primary address becomes the source IP address for the TCP connection. See optional for information about configuring the connect-source command.</p>
Step 4	<p><code>exit</code></p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config-if)# end</pre>	Exits interface configuration mode.
Step 5	<p><code>router msdp</code></p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config)# router msdp</pre>	Enters MSDP protocol configuration mode.
Step 6	<p><code>default-peer ip-address [prefix-list list]</code></p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config-msdp)# default-peer 172.23.16.0</pre>	(Optional) Defines a default peer from which to accept all MSDP SA messages.
Step 7	<p><code>originator-id type interface-path-id</code></p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config-msdp)# originator-id /1/1/0</pre>	(Optional) Allows an MSDP speaker that originates a (Source-Active) SA message to use the IP address of the interface as the RP address in the SA message.

	Command or Action	Purpose
Step 8	peer <i>peer-address</i> Example: RP/0/RP0/CPU0:router(config-msdp)# peer 172.31.1.2	Enters MSDP peer configuration mode and configures an MSDP peer. <ul style="list-style-type: none"> • Configure the router as a BGP neighbor. • If you are also BGP peering with this MSDP peer, use the same IP address for MSDP and BGP. You are not required to run BGP or multiprotocol BGP with the MSDP peer, as long as there is a BGP or multiprotocol BGP path between the MSDP peers.
Step 9	connect-source <i>type interface-path-id</i> Example: RP/0/RP0/CPU0:router(config-msdp-peer)# connect-source loopback 0	(Optional) Configures a source address used for an MSDP connection.
Step 10	mesh-group <i>name</i> Example: RP/0/RP0/CPU0:router(config-msdp-peer)# mesh-group internal	(Optional) Configures an MSDP peer to be a member of a mesh group.
Step 11	remote-as <i>as-number</i> Example: RP/0/RP0/CPU0:router(config-msdp-peer)# remote-as 250	(Optional) Configures the remote autonomous system number of this peer.
Step 12	commit	
Step 13	show msdp [ipv4] globals Example: RP/0/RP0/CPU0:router# show msdp globals	Displays the MSDP global variables.
Step 14	show msdp [ipv4] peer [peer-address] Example: RP/0/RP0/CPU0:router# show msdp peer 172.31.1.2	Displays information about the MSDP peer.
Step 15	show msdp [ipv4] rpf rpf-address Example: RP/0/RP0/CPU0:router# show msdp rpf 172.16.10.13	Displays the RPF lookup.

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.

SUMMARY STEPS

1. **configure**
2. **router msdp**
3. **sa-filter** {in | out} {ip-address | peer-name} [list access-list-name] [rp-list access-list-name]
4. **cache-sa-state** [list access-list-name] [rp-list access-list-name]
5. **ttl-threshold** ttl-value
6. **exit**
7. **ipv4 access-list** name [sequence-number] **permit** source [source-wildcard]
8. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	router msdp Example: RP/0/RP0/CPU0:router(config)# router msdp	Enters MSDP protocol configuration mode.

	Command or Action	Purpose
Step 3	<p>sa-filter {in out} {ip-address peer-name} [list access-list-name] [rp-list access-list-name]</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config-msdp)# sa-filter out router.cisco.com list 100</pre>	<p>Configures an incoming or outgoing filter list for messages received from the specified MSDP peer.</p> <ul style="list-style-type: none"> If you specify both the list and rp-list keywords, all conditions must be true to pass any source, group (S, G) pairs in outgoing Source-Active (SA) messages. You must configure the ipv4 access-list command in Step 7, on page 20. If all match criteria are true, a permit from the route map passes routes through the filter. A deny filters routes. This example allows only (S, G) pairs that pass access list 100 to be forwarded in an SA message to the peer named router.cisco.com.
Step 4	<p>cache-sa-state [list access-list-name] [rp-list access-list-name]</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config-msdp)# cache-sa-state list 100</pre>	<p>Creates and caches source/group pairs from received Source-Active (SA) messages and controls pairs through access lists.</p>
Step 5	<p>t11-threshold ttl-value</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config-msdp)# t11-threshold 8</pre>	<p>(Optional) Limits which multicast data is sent in SA messages to an MSDP peer.</p> <ul style="list-style-type: none"> Only multicast packets with an IP header TTL greater than or equal to the <i>ttl-value</i> argument are sent to the MSDP peer specified by the IP address or name. Use this command if you want to use TTL to examine your multicast data traffic. For example, you could limit internal traffic to a TTL of 8. If you want other groups to go to external locations, send those packets with a TTL greater than 8. This example configures a TTL threshold of eight hops.
Step 6	<p>exit</p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router(config-msdp)# exit</pre>	<p>Exits the current configuration mode.</p>
Step 7	<p>ipv4 access-list name [sequence-number] permit source [source-wildcard]</p> <p>Example:</p>	<p>Defines an IPv4 access list to be used by SA filtering.</p> <ul style="list-style-type: none"> In this example, the access list 100 permits multicast group 239.1.1.1.

	Command or Action	Purpose
	RP/0/RP0/CPU0:router(config)# ipv4 access-list 100 20 permit 239.1.1.1 0.0.0.0	<ul style="list-style-type: none"> The ipv4 access-list command is required if the keyword list is configured for SA filtering in Step 3, on page 20.
Step 8	commit	

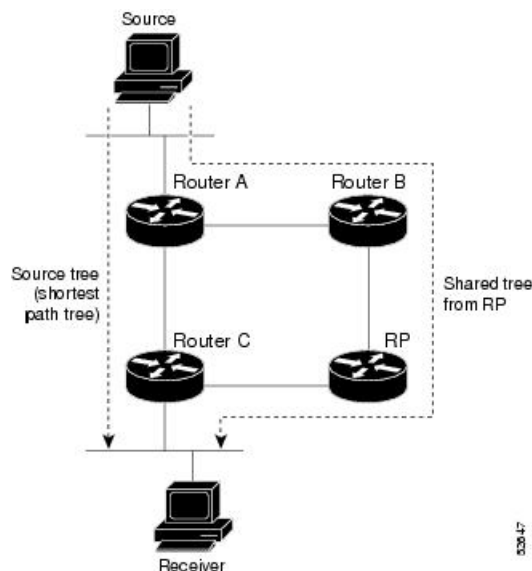
PIM-Sparse Mode

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.

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

Figure 1: 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 set up 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 Figure 4: Shared Tree and Source Tree

(Shortest Path Tree), above. Data from senders is delivered to the RP for distribution to group members joined to the shared tree.

Unless the 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.
9. 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.



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

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.

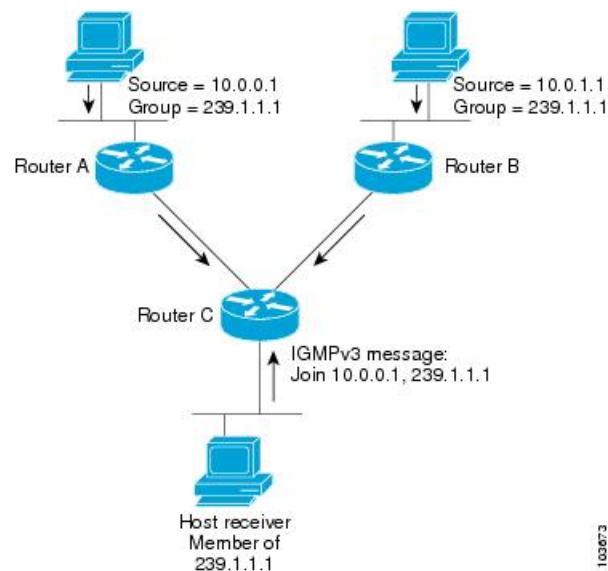
If there are multiple PIM-SM routers on a LAN, a designated router must be elected to avoid duplicating multicast traffic for connected hosts. The PIM router with the highest IP address becomes the DR for the LAN unless you choose to force the DR election by use of the **dr-priority** command. The DR 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 2: 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 TenGigE0/0/0/1
  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
```

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 messages; that is, router queries and host reports.

A set of routers and hosts that receive multicast data streams from the same source is called a multicast group. Hosts use IGMP 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 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.

NCS 5500 supports IGMPv3 by default. No configuration is required. IGMP Version 3 permits joins and leaves for certain source and group pairs instead of requesting traffic from all sources in the multicast group.

Restrictions

IGMP snooping under VPLS bridge domain is not supported.

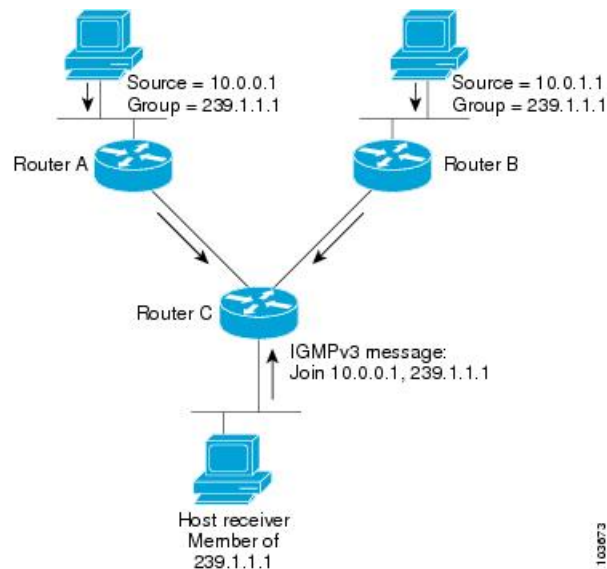
Functioning of IGMP Routing

The following image "IGMP Singaling" , 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 3: IGMP Signaling



Configuring Maximum IGMP Per Interface Group Limit

The IGMP Per Interface States Limit sets a limit on creating OIF 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.

- 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.

Configuration Example

Configures all interfaces with 4000 maximum groups per interface except TenGigE interface 0/0/0/6, which is set to 3000:

```
Router#config
Router(config)#router igmp
Router(config-igmp)#maximum groups-per-interface 4000
Router(config-igmp)#interface TenGigE0/0/0/6
Router(config-igmp-default-if)#maximum groups-per-interface 3000
Router(config-igmp-default-if)#commit
```

Running Configuration

```
router igmp
 interface TenGigE0/0/0/6
   maximum groups-per-interface 3000
 !
 maximum groups-per-interface 4000
 !
```

Verification

```
Router#show igmp summary
Robustness Value 2
No. of Group x Interfaces 37
Maximum number of Group x Interfaces 50000
Supported Interfaces : 9
Unsupported Interfaces: 0
Enabled Interfaces : 8
Disabled Interfaces : 1
MTE tuple count : 0
Interface                Number  Max #
                          Groups   Groups
Loopback0                 4       4000
TenGigE0/0/0/0            5       4000
TenGigE0/0/0/1            5       4000
TenGigE0/0/0/2            0       4000
TenGigE0/0/0/3            5       4000
TenGigE0/0/0/6            5       3000
TenGigE0/0/0/18           5       4000
TenGigE0/0/0/19           5       4000
TenGigE0/0/0/6.1         3       4000
```

SSM Static Source Mapping

Configure a source (1.1.1.1) as part of a set of sources that map SSM groups described by the specified access-list (4).

Configuration Example

```

Router#configure
Router(config)#ipv4 access-list 4
Router(config-ipv4-acl)#permit ipv4 any 229.1.1.0 0.0.0.255
Router(config-ipv4-acl)#exit
Router(config)# multicast-routing
Router(config-mcast)#address-family ipv4
Router(config-mcast-default-ipv4)#ssm range 4
Router(config-mcast-default-ipv4)#exit
Router(config-mcast)#exit
Router(config)#router igmp
Router(config-igmp)#ssm map static 1.1.1.1 4
*/Repeat the above step as many times as you have source addresses to include in the set
for SSM mapping/*
Router(config-igmp)#interface TenGigE0/0/0/3
Router(config-igmp-default-if)#static-group 229.1.1.1
Router(config-igmp-default-if)#commit

```

Running Configuration

```

Router#show run multicast-routing
multicast-routing
  address-family ipv4
    ssm range 4
  interface all enable
  !
  !
Router#show access-lists 4
ipv4 access-list 4
  10 permit ipv4 any 229.1.1.0 0.0.0.255

Router#show run router igmp
router igmp
  interface TenGigE0/0/0/3
  static-group 229.1.1.1
  !
  ssm map static 1.1.1.1 4

```

Verification

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

```

Router#show mrib route 229.1.1.1 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, 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
(1.1.1.1,229.1.1.1) RPF nbr: 1.1.1.1 Flags: RPF
Up: 00:01:11
Incoming Interface List

```

```

Loopback0 Flags: A, Up: 00:01:11
Outgoing Interface List
TenGigE0/0/0/3 Flags: F NS LI, Up: 00:01:11

```

Information About 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. An empty profile enables the default configuration options and settings for IGMP snooping, as listed in the *Default IGMP Snooping Configuration Settings*.

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 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. Profiles can be attached to ports under the bridge, one profile per port.
- 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.



Note

- IEEE 802.1Q, Q-in-Q, default and untagged encapsulations are supported.
- IEEE 802.1ad and other encapsulations are not supported.
- IEEE 802.1ad is not supported with BVI.

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 **mldp snooping profile** command in l2vpn bridge group bridge domain configuration mode. To attach a profile to a port, use the **mldp 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, MLDP snooping is deactivated in the bridge domain.
- If you detach a profile from a port, MLDP 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 you need to change an active profile, you must detach it from all bridges or ports, change it, 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.

Configuring Access Control

Access control configuration is the configuration of access groups and weighted group limits.

The role of access groups in IGMP v2/v3 message filtering is to permit or deny host membership requests for multicast groups (*,G) and multicast source groups (S,G). This is required to provide blocked and allowed list access to IPTV channel packages.

Weighted group limits restrict the number of IGMP v2/v3 groups, in which the maximum number of concurrently allowed multicast channels can be configured on a per EFP- and per PW-basis.

IGMP Snooping Access Groups

Although Layer-3 IGMP routing also uses the **igmp access-group** command in support of access groups, the support is not the same in Layer-2 IGMP, because the Layer-3 IGMP routing access group feature does not support source groups.

Access groups are specified using an extended IP access list referenced in an IGMP snooping profile that you attach to a bridge domain or a port.



Note A port-level access group overrides any bridge domain-level access group.

The **access-group** command instructs IGMP snooping to apply the specified access list filter to received membership reports. By default, no access list is applied.

Changes made to the access-list referenced in the profile (or a replacement of the access-list referenced in the igmp snooping profile) will immediately result in filtering the incoming igmp group reports and the existing group states accordingly, without the need for a detach-reattach of the igmp snooping profile in the bridge-domain, each time such a change is made.

IGMP Snooping Group Weighting

To limit the number of IGMP v2/v3 groups, in which the maximum number of concurrently allowed multicast channels must be configurable on a per EFP-basis and per PW-basis, configure group weighting.

IGMP snooping limits the membership on a bridge port to a configured maximum, but extends the feature to support IGMPv3 source groups and to allow different weights to be assigned to individual groups or source

groups. This enables the IPTV provider, for example, to associate standard and high-definition IPTV streams, as appropriate, to specific subscribers.

This feature does not limit the actual multicast bandwidth that may be transmitted on a port. Rather, it limits the number of IGMP groups and source-groups, of which a port can be a member. It is the responsibility of the IPTV operator to configure subscriber membership requests to the appropriate multicast flows.

The **group policy** command, which is under `igmp-snooping-profile` configuration mode, instructs IGMP snooping to use the specified route policy to determine the weight contributed by a new `<*,G>` or `<S,G>` membership request. The default behavior is for there to be no group weight configured.

The **group limit** command specifies the group limit of the port. No new group or source group is accepted if its contributed weight would cause this limit to be exceeded. If a group limit is configured (without group policy configuration), a `<S/*,G>` group state will have a default weight of 1 attributed to it.



Note By default, each group or source-group contributes a weight of 1 towards the group limit. Different weights can be assigned to groups or source groups using the `group policy` command.

The group limit policy configuration is based on these conditions:

- Group weight values for `<*,G>` and `<S,G>` membership are configured in a Route Policy, that is included in an `igmp snooping profile` attached to a BD or port.
- Port level weight policy overrides any bridge domain level policy, if `group-limit` is set and `route-policy` is configured.
- If there is no policy configured, each group weight is counted equally and is equal to 1.
- If policy has been configured, all matching groups get weight of 1 and un-matched groups have 0 weight.

IPv6 Multicast for Multiple Sources



Note When a router has LCs (with and without external TCAMs), it operates with default IPv6 multicast route scale, which is programmed on the LC without an external TCAM.

Statistics for Ingress Multicast Routes

Multicast and interface statistics are often used for accounting purpose. By default Multicast Forwarding Information Base (MFIB) does not store multicast route statistics. In order to enable MFIB to create and store multicast route statistics for ingress flows use the **hw-module profile mfib statistics** command in the configuration mode. Use `No` form of the latter command to disable logging of multicast route statistics. For the configuration to take effect, you must reload affected line cards after executing the command .



Note The MFIB counter can store a maximum of 2000 multicast routes, if the count exceeds beyond 2000 routes, then statistics are overwritten.

This table lists commands used to display or reset multicast route statistics stored in MFIB.

Table 3: show and clear commands for Multicast Statistics

Command	Description
show mfib hardware route statistics location <node-id>	Displays platform-specific MFIB information for the packet and byte counters for all multicast route.
show mfib hardware route <source-address> location <node-id>	Displays platform-specific MFIB information for the packet and byte counters for multicast routes originating from the specified source.
Show mfib hardware route <source-address><group-address> location <node-id>	Displays platform-specific MFIB information for the packet and byte counters for multicast routes originating from the specified source and belonging to specified multicast group.
clear mfib hardware ingress route statistics location <node-id>	Resets allocated counter values regardless of the MFIB hardware statistics mode from the designated node.
clear mfib hardware route <source-address> location <node-id>	Resets allocated counter values regardless of the MFIB hardware statistics mode from the designated node for specified multicast route source.
clear mfib hardware route <source-address><group-address> location <node-id>	Resets allocated counter values regardless of the MFIB hardware statistics mode from the designated node for specified multicast route source and multicast group.



Note To program IPv6 multicast routes in external TCAM, use the following commands:

- **hw-module profile tcam fib v6mcast**
- **hw-module profile tcam fib ipv6 unicast**

Configuring only **hw-module profile tcam fib v6mcast percent value** causes an unexpected behaviour for 5501-SE.

Configuring Statistics for Ingress Multicast Routes

Configuration Example

In this example you will enable MRIB route statistics logging for ingress multicast routes for all locations:

```

Router#config
Router(config)#hw-module profile mfib statistics
Router(config)#commit
Router(config)#exit
Router#admin
Router(admin)#reload location all /*Reloads all line cards. This is required step,
else the multicast route statistics will not be created.*/

```

Running Configuration

```

Router#show running config
hw-module profile mfib statistics
!

```

Verification

The below show commands display the multicast statistics for source (192.0.2.2), group (226.1.2.1) and node location 0/1/cpu0 for ingress route:



Note The multicast egress statistics per flow (per SG) is not supported. But egress interface level multicast statistics is supported. Also the drop statistics is not supported.

```

Router#show mfib hardware route statistics location 0/1/cpu0
(192.0.2.2, 226.1.2.1) :: Packet Stats 109125794 Byte Stats 23680297298
(192.0.2.2, 226.1.2.2) :: Packet Stats 109125760 Byte Stats 23680289920
(192.0.2.2, 226.1.2.3) :: Packet Stats 109125722 Byte Stats 23680281674
(192.0.2.2, 226.1.2.4) :: Packet Stats 109125683 Byte Stats 23680273211
(192.0.2.2, 226.1.2.5) :: Packet Stats 109125644 Byte Stats 23680264748
(192.0.2.2, 226.1.2.6) :: Packet Stats 109129505 Byte Stats 23681102585
(192.0.2.2, 226.1.2.7) :: Packet Stats 109129470 Byte Stats 23681094990
(192.0.2.2, 226.1.2.8) :: Packet Stats 109129428 Byte Stats 23681085876
(192.0.2.2, 226.1.2.9) :: Packet Stats 109129385 Byte Stats 23681076545
(192.0.2.2, 226.1.2.10) :: Packet Stats 109129336 Byte Stats 23681065912

```

```

Router#show mfib hardware route statistics 192.0.2.2 location 0/1/cpu0
(192.0.2.2, 226.1.2.1) :: Packet Stats 109184295 Byte Stats 23692992015
(192.0.2.2, 226.1.2.2) :: Packet Stats 109184261 Byte Stats 23692984637
(192.0.2.2, 226.1.2.3) :: Packet Stats 109184223 Byte Stats 23692976391
(192.0.2.2, 226.1.2.4) :: Packet Stats 109184184 Byte Stats 23692967928
(192.0.2.2, 226.1.2.5) :: Packet Stats 109184145 Byte Stats 23692959465
(192.0.2.2, 226.1.2.6) :: Packet Stats 109184106 Byte Stats 23692951002
(192.0.2.2, 226.1.2.7) :: Packet Stats 109184071 Byte Stats 23692943407
(192.0.2.2, 226.1.2.8) :: Packet Stats 109184029 Byte Stats 23692934293
(192.0.2.2, 226.1.2.9) :: Packet Stats 109183986 Byte Stats 23692924962

```

```

Router#show mfib hardware route statistics 192.0.2.2 226.1.2.1 location 0/1/cpu0
(192.0.2.2, 226.1.2.1) :: Packet Stats 109207695 Byte Stats 23698069815

```

Multicast Route Statistics

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

Before enabling multicast route statistics, you must configure an ACL to specify which of the IP route statistics to be captured.

Restrictions for Implementing Multicast Route Statistics Feature

These are the points that you should consider before implementing multicast route statistics feature:

- Multicast route statistics are available for <S,G> routes only. The statistics for <*,G> routes are not available.
- IPv6 multicast route statistics are not supported.
- Multicast route statistics for egress direction is not supported.
- When ACL is mapped with **hw-module router-stats** configuration, you can't modify the ACL. To modify ACLs that are mapped with router-stats, remove the existing **hw-module router-stats** configuration and update the ACL entries. Then, configure the **hw-module router-stats** again.

This feature supports only:

- L3 Multicast traffic.
- Default VRFs.

Configure Multicast Route Statistics

Configuring multicast route statistics includes these main tasks:

- Configuring an ACL
- Enabling multicast route statistics for the configured ACLs

```
RP0/0/RP0/CPU0:router# configure

/* Configure an ACL matching the (S,G) routes for which statistics have to be captured:*/
RP0/0/RP0/CPU0:router(config)# ipv4 access-list mcast-counter
RP0/0/RP0/CPU0:router(config-acl)# 10 permit ipv4 host 10.1.1.2 host 224.2.151.1
RP0/0/RP0/CPU0:router(config-acl)# 30 permit ipv4 10.1.1.0/24 232.0.4.0/22
RP0/0/RP0/CPU0:router(config-acl)# 50 permit ipv4 192.168.0.0/24 232.0.4.0/22
RP0/0/RP0/CPU0:router(config-acl)# commit
RP0/0/RP0/CPU0:router(config-acl)# exit

/* Enable multicast route statistics for the configured ACL in the ingress direction on the
default VRF. */
RP0/0/RP0/CPU0:router(config)# hw-module route-stats l3mcast vrf default ipv4 egress
mcast-router
RP0/0/RP0/CPU0:router(config)# hw-module route-stats l3mcast vrf default ipv4 ingress
mcast-router
```



Note

- If you are enabling the route stats for a router on the global table, use **vrf default**. If you are enabling the route stats for specific vrf, use the **vrf vrfname** option.
- In case, you want to enable route stats for all tables, do not use the **vrf**.

For example:

```
RP0/0/RP0/CPU0:router(config)#hw-module route-stats l3mcast ipv4 ingress mcast-counter
```

Verification

Use the **show mfib route rate** command to verify if the multicast route information is captured for the traffic that matches the ACL:



Note The ingress stats are always per S, G.

```
RP0/0/RP0/CPU0:router# show mfib route rate
Thu Aug 16 18:04:47.312 PDT

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

(10.1.1.2,232.0.0.1)
  Incoming rate :
    Node : 0/0/CPU0 : 4593 / 18153671
  Outgoing rate :
    Node : 0/0/CPU0 : 0 / 0
```

The above output shows that the multicast source **10.1.1.2** is sending packets to multicast group **232.0.0.1** and is received at **4593** pps.

IPv6 Egress Multicast Route Statistics Example

```
RP0/0/RP0/CPU0:router# configure

/* Configure an ACL matching the (S,G) routes for which statistics have to be captured:*/
RP0/0/RP0/CPU0:router(config)# ipv6 access-list 12
RP0/0/RP0/CPU0:router(config-acl)# 10 permit ipv6 any ff33:1:3::1/48
RP0/0/RP0/CPU0:router(config-acl)#commit
RP0/0/RP0/CPU0:router(config-acl)#exit

/* Enable multicast route statistics for the configured ACL in the egress direction on the
   named VRF. */
RP0/0/RP0/CPU0:router(config)# hw-module route-stats l3mcast vrf vrf1 ipv6 egress 12
```

For the information on the interface accounting stats, use the show interface accounting command. The following show command displays interface accounting stats for ingress:

```
Router# show int tenGigE 0/0/0/15 accounting
Mon Nov 12 10:26:20.592 UTC
TenGigE0/0/0/15
  Protocol          Pkts In      Chars In      Pkts Out      Chars Out
  IPV6_MULTICAST    22125711958  1814308380556 0               0
  IPV6_ND           0            0              1243          128960
```

Cisco IOS XR Release 7.4.1 and later support YANG data model for multicast interface counters.

- Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface/protocols/protocol
- Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface[interface-name=TenGigE0/0/0/18]/protocols/protocol



Note The YANG model does not support ingress and egress multicast route stats.

The following show command displays interface accounting stats for egress:

```
Router# show interfaces bundle-ether 100.1001 accounting rates
Mon Aug 26 15:56:41.738 IST
Bundle-Ether100.1001
```

Protocol	Ingress		Egress	
	Bits/sec	Pkts/sec	Bits/sec	Pkts/sec
IPV4_MULTICAST	0	0	11455000	990
IPV6_MULTICAST	0	0	11455000	990
ARP	0	0	0	0
IPV6_ND	0	0	0	0

Use Case: Video Streaming

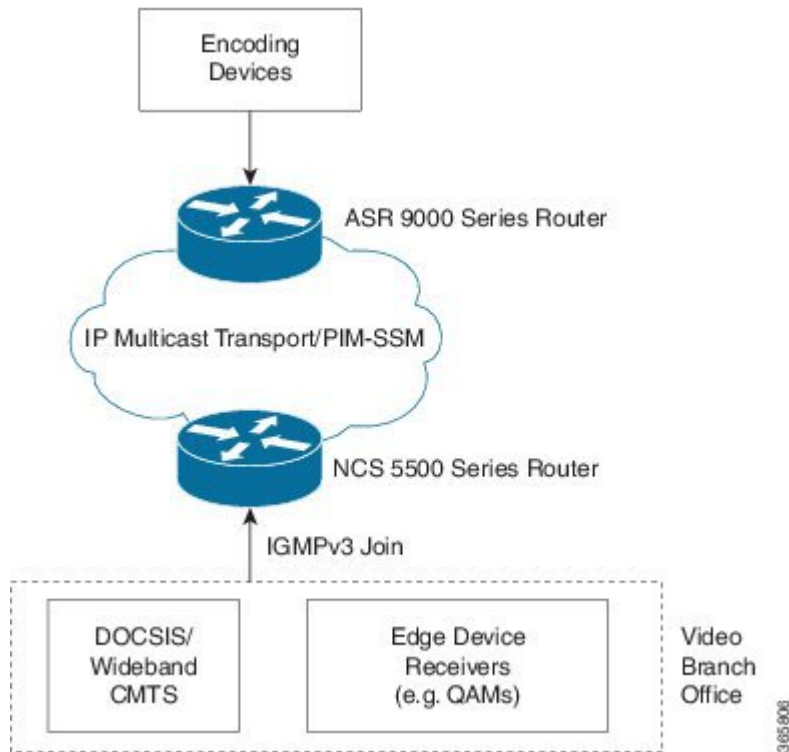
In today's broadcast video networks, proprietary transport systems are used to deliver entire channel line-ups to each video branch office. IP based transport network would be a cost efficient/convenient alternative to deliver video services combined with the delivery of other IP based services. (Internet delivery or business services)

By its very nature, broadcast video is a service well-suited to using IP multicast as a more efficient delivery mechanism to reach end customers.

The IP multicast delivery of broadcast video is explained as follows:

1. Encoding devices in digital primary headends, encode one or more video channels into a Moving Pictures Expert Group (MPEG) stream which is carried in the network via IP multicast.
2. Devices at video branch office are configured by the operator to request the desired multicast content via IGMP joins.
3. The network, using PIM-SSM as its multicast routing protocol, routes the multicast stream from the digital primary headend to edge device receivers located in the video branch office. These edge devices could be edge QAM devices which modulate the MPEG stream for an RF frequency, or CMTS for DOCSIS.

Figure 4: Video Streaming



Multicast Label Distribution Protocol (MLDP) for Core

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.

Characteristics of MLDP Profiles on Core

The following MLDP profiles are supported when the router is configured as a core router:

- Profile 5—Partitioned MDT - MLDP P2MP - BGP-AD - PIM C-mcast Signaling
- Profile 6—VRF MLDP - In-band Signaling
- Profile 7—Global MLDP In-band Signaling
- Profile 12—Default MDT - MLDP - P2MP - BGP-AD - BGP C-mcast Signaling
- Profile 17—Default MDT - MLDP - P2MP - BGP-AD - PIM C-mcast Signaling

Point-to-Multipoint Profiles on Core and Edge Routers

The following profiles are supported when the router is configured as a core router and edge router for p2mp:

- Profile 8—Global P2MP-TE
- Profile 10—VRF Static-P2MP-TE with BGP AD

Multicast VPN Support based on Point to Multipoint Traffic Engineering (P2MPE)

Table 4: Feature History Table

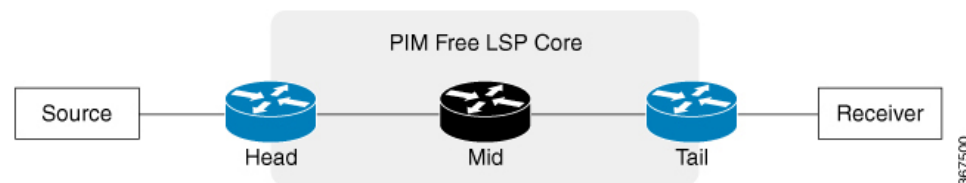
Feature Name	Release Information	Feature Description
Multicast VPN Support based on Point to Multipoint Traffic Engineering (P2MPE)	Release 7.4.1	This feature is now supported on routers that have the Cisco NC57 line cards installed and operate in native and compatibility mode.

To carry multicast traffic in service provider networks, a multicast protocol like PIM needs to be deployed to set up forwarding paths in the service provider core. However for an MPLS backbone network, service providers can use label encapsulation instead of IP tunneling. This approach helps to reduce the control traffic overhead on the service provider core and also leverages the MPLS traffic engineering and protection features.

The label encapsulation could be either point-to-multipoint (P2MP) label switched paths (LSPs) or multipoint-to-multipoint (MP2MP) LSPs. For creating multicast LSPs, RSVP-TE protocol extensions can be used. The RSVP-TE protocol is extended to signal P2MP LSPs across the MPLS networks. P2MP-TE feature enables transporting multicast traffic through a PIM free service provider core using P2MP-TE tunnels.

The following figure explains the topology that is used in this feature.

Figure 5: PIM Free LSP Core



In this figure, the following terminologies are used:

- Head—A router on which a TE tunnel is configured.
- Tail—The router on which the TE tunnel terminates.
- Mid—A router through which the TE tunnel passes.

A Multicast VPN (mVPN) profile is configured for the global context or per VRF. Different mVPN profiles can be applied depending on where the multicast streams need to be transported.

The following mVPN profiles are supported for the P2MP-TE feature:

- mVPN profile 8 for global context
- mVPN profile 10 for L3VPN context

Restrictions and Usage Guidelines

The following restrictions and guidelines apply for this feature:

- Only Source-Specific Multicast (SSM) traffic is supported.
- For profile 8, both IPv4 and IPv6 are supported.
- For profile 10, only IPv4 is supported.
- Fast Reroute (FRR) for P2MP-TE tunnel is not supported.
- BVI interface is not supported.
- FRR for P2MP RSVP currently is not supported.
- FRR for multicast traffic with profile 6,7,14 is supported.

Configuration Example: P2MP-TE Profile 8

This example shows the P2MP-TE configuration for profile 8. You need to configure the head, mid, and tail routers in the P2MP tunnel.

The head router configuration is given as follows. This configuration includes IGP, MPLS-TE tunnel, and multicast configurations. You should also configure LDP and RSVP while configuring this feature.

The running configuration for the head router is given as follows.

The mid router only requires MPLS-TE, RSVP and an IGP like OSPF configurations. The running configuration for the mid router is given as follows:

The tail router configuration is given as follows. This configuration includes IGP, MPLS-TE tunnel and multicast configurations. Similar to head router, you should also configure RSVP and LDP while configuring this feature.

The running configuration for the tail router is given as follows:

Configuration Example: P2MP-TE Profile 10

This example shows the P2MP-TE configuration for profile 10. You need to configure the head, mid, and tail routers.

The head router configuration is given as follows. This configuration includes IGP, L3VPN, and multicast configurations. You should also configure MPLS-TE, LDP, and RSVP while configuring this feature.

The running configuration for the head router is given as follows.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# router ospf 1
RP/0/RP0/CPU0:router(config-router)# area 0
RP/0/RP0/CPU0:router(config-ospf-ar)# mpls traffic-eng
RP/0/RP0/CPU0:router(config-ospf-ar-mpls-te)# exit
RP/0/RP0/CPU0:router(config-ospf-ar)# interface Loopback0
RP/0/RP0/CPU0:router(config-ospf-ar-if)# exit
RP/0/RP0/CPU0:router(config-ospf-ar)# interface TenGigE0/0/0/3
```



```

RP/0/RP0/CPU0:router(config-ospf-ar-if)# cost 1
RP/0/RP0/CPU0:router(config-ospf-ar-if)# network point-to-point
RP/0/RP0/CPU0:router(config-ospf-ar-if)# exit
RP/0/RP0/CPU0:router(config-ospf-ar)# exit
RP/0/RP0/CPU0:router(config-ospf)# mpls traffic-eng router-id loopback 0
RP/0/RP0/CPU0:router(config-ospf)# exit
RP/0/RP0/CPU0:router(config)# vrf vpn_2
RP/0/RP0/CPU0:router(config-vrf)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-vrf-af)# import route-target 100:2
RP/0/RP0/CPU0:router(config-vrf-af)# export route-target 120:2
RP/0/RP0/CPU0:router(config)# interface TengigE0/0/0/6
RP/0/RP0/CPU0:router(config-if)# vrf vpn_2
RP/0/RP0/CPU0:router(config-if-vrf)# ipv4 address 10.0.0.1 255.255.255.0
RP/0/RP0/CPU0:router(config)# route-policy pass-all
RP/0/RP0/CPU0:router(config)# pass
RP/0/RP0/CPU0:router(config)# end-policy
RP/0/RP0/CPU0:router(config)# router bgp 1
RP/0/RP0/CPU0:router(config-bgp)# bgp router-id 10.2.2.2
RP/0/RP0/CPU0:router(config-bgp)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-bgp-af)# address-family vpnv4 unicast
RP/0/RP0/CPU0:router(config-bgp-af)# address-family ipv4 mvpn
RP/0/RP0/CPU0:router(config-bgp)# neighbor 10.1.1.1
RP/0/RP0/CPU0:router(config-bgp-nbr)# remote-as 1
RP/0/RP0/CPU0:router(config-bgp-nbr)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-bgp-nbr-af)# route-policy pass-all in
RP/0/RP0/CPU0:router(config-bgp-nbr-af)# route-policy pass-all out
RP/0/RP0/CPU0:router(config-bgp-nbr-af)# address-family vpnv4 unicast
RP/0/RP0/CPU0:router(config-bgp-nbr-af)# route-policy pass-all in
RP/0/RP0/CPU0:router(config-bgp-nbr-af)# route-policy pass-all out
RP/0/RP0/CPU0:router(config-bgp-nbr)# address-family ipv4 mvpn
RP/0/RP0/CPU0:router(config)# multicast-routing
RP/0/RP0/CPU0:router(config-mcast)# address-family ipv4
RP/0/RP0/CPU0:router(config-mcast-default-ipv4)# interface Loopback0
RP/0/RP0/CPU0:router(config-mcast-default-ipv4-if)# enable
RP/0/RP0/CPU0:router(config-mcast-default-ipv4-if)# exit
RP/0/RP0/CPU0:router(config-mcast-default-ipv4)# mdt source Loopback0
RP/0/RP0/CPU0:router(config-mcast)# vrf vpn_2
RP/0/RP0/CPU0:router(config-mcast-vpn_2)# address-family ipv4
RP/0/RP0/CPU0:router(config-mcast-vpn_2-ipv4)# mdt source loopback0
RP/0/RP0/CPU0:router(config-mcast-vpn_2-ipv4)# core-tree-protocol rsvp-te
RP/0/RP0/CPU0:router(config-mcast-vpn_2-ipv4)# rate-per-route
RP/0/RP0/CPU0:router(config-mcast-vpn_2-ipv4)# interface all enable
RP/0/RP0/CPU0:router(config-mcast-vpn_2-ipv4)# bgp auto-discovery p2mp-te
RP/0/RP0/CPU0:router(config)# router igmp
RP/0/RP0/CPU0:router(config-igmp)# vrf vpn_2
RP/0/RP0/CPU0:router(config-igmp-vpn_2)# interface TenGigE0/0/0/6
RP/0/RP0/CPU0:router(config-igmp-vpn_2-if)# version 3
RP/0/RP0/CPU0:router(config-igmp-vpn_2-if)# exit
RP/0/RP0/CPU0:router(config)# router pim
RP/0/RP0/CPU0:router(config-pim)# vrf vpn_2
RP/0/RP0/CPU0:router(config-pim-vpn_2)# address-family ipv4
RP/0/RP0/CPU0:router(config-pim-vpn_2-ipv4)# interface TenGigE0/0/0/6
RP/0/RP0/CPU0:router(config-pim-vpn_2-ipv4-if)# enable
RP/0/RP0/CPU0:router(config)#router bgp 1
RP/0/RP0/CPU0:router(config-bgp)# bgp router-id 192.168.1.2
RP/0/RP0/CPU0:router(config-bgp)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-bgp-af)# address-family vpnv4 unicast
RP/0/RP0/CPU0:router(config-bgp-af)# address-family ipv4 mvpn
RP/0/RP0/CPU0:router(config-bgp)# neighbor 192.168.1.1
RP/0/RP0/CPU0:router(config-bgp-nbr)# remote-as 2002
RP/0/RP0/CPU0:router(config-bgp-nbr)# address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-bgp-nbr-af)# route-policy pass-all in
RP/0/RP0/CPU0:router(config-bgp-nbr-af)# route-policy pass-all out
RP/0/RP0/CPU0:router(config-bgp-nbr)# address-family vpnv4 unicast

```

```

RP/0/RP0/CPU0:router(config-bgp-nbr-af)# route-policy pass-all in
RP/0/RP0/CPU0:router(config-bgp-nbr-af)# route-policy pass-all out
RP/0/RP0/CPU0:router(config-bgp-nbr)# address-family ipv4 mvpn
RP/0/RP0/CPU0:router(config-bgp)# vrf vpn_2
RP/0/RP0/CPU0:router(config-bgp-vrf)#rd 100:2
RP/0/RP0/CPU0:router(config-bgp-vrf)#address-family ipv4 unicast
RP/0/RP0/CPU0:router(config-bgp-vrf-af)#label mode per-prefix
RP/0/RP0/CPU0:router(config-bgp-vrf-af)#redistribute connected
RP/0/RP0/CPU0:router(config-bgp-vrf-af)#exit
RP/0/RP0/CPU0:router(config-bgp-vrf)# address-family ipv4 mvpn

```

Running configuration for the tail router is given as follows:

```

interface Loopback0
  ipv4 address 10.2.2.2 255.255.255.255
!
interface TenGigE0/0/0/3
  ipv4 address 10.3.0.2 255.255.255.0
!
router ospf 1
  area 0
    mpls traffic-eng
    interface Loopback0
    !
    interface TenGigE0/0/0/3
      cost 1
      network point-to-point
    !
  !
  mpls traffic-eng router-id Loopback0
  !
  rsvp
    interface TenGigE0/0/0/3
      bandwidth percentage 100
    !
  !
  mpls traffic-eng
    interface TenGigE0/0/0/3
    !
  mpls ldp
    discovery
      targeted-hello interval 10
    !
    router-id 10.2.2.2
    address-family ipv4
      discovery targeted-hello accept
    !
    interface TenGigE0/0/0/3
    !
  ! vrf vpn_2
    address-family ipv4 unicast
      import route-target
        100:2
      export route-target
        100:2

interface TenGigE0/0/0/6
  vrf vpn_2
  ipv4 address 10.6.0.2 255.255.255.0

route-policy pass-all
  pass
end-policy

```

```

router bgp 1
  bgp router-id 10.2.2.2
  address-family ipv4 unicast
  address-family vpnv4 unicast
  address-family ipv4 mvpn
  neighbor 10.1.1.1
    remote-as 1
  update-source Loopback0
  address-family ipv4 unicast
    route-policy pass-all in
    route-policy pass-all out
  address-family vpnv4 unicast
    route-policy pass-all in
    route-policy pass-all out
  address-family ipv4 mvpn
vrf vpn_2
  rd 100:2
  address-family ipv4 unicast
    label mode per-prefix
    redistribute connected
  address-family ipv4 mvpn
!
multicast-routing
  address-family ipv4
    interface Loopback0
      enable
    !
    mdt source Loopback0
    !
  vrf vpn_2
    address-family ipv4
      mdt source Loopback0
      core-tree-protocol rsvp-te
      rate-per-route
      interface all enable
      bgp auto-discovery p2mp-te
    !
  !
router igmp
  vrf vpn_2
    interface TenGigE0/0/0/6
      version 3
    !
  !
router pim
  vrf vpn_2
    address-family ipv4
      interface TenGigE0/0/0/6
        enable
    !
  !
!

```

Verification: P2MP-TE

This example shows how to verify if the multicast control state is correct on the head router using the **show mrib vrf vpn_2 route** command.

```

RP/0/RP0/CPU0:router# show mrib vrf vpn_2 route

(10.0.0.100,232.0.0.1) RPF nbr: 10.0.0.100 Flags: RPF
Up: 00:00:38
Incoming Interface List

```

```
TenGigE0/0/0/0 Flags: A, Up: 00:00:38
Outgoing Interface List
Tunnel-mte2 Flags: F NS LI LVIF, Up: 00:00:38
```

You can also verify the multicast control state on the tail router.

```
RP/0/RP0/CPU0:router# show mrib vrf vpn_2 route

(10.0.0.100,232.0.0.1) RPF nbr: 10.1.1.1 Flags: RPF
Up: 00:03:55
Outgoing Interface List
TenGigE0/0/0/6 Flags: F NS LI, Up: 00:03:55
```

This example shows how to check if the TE tunnel is established on the head router by using the **show mpls traffic-eng tunnels p2mp** command.

```
RP/0/RP0/CPU0:router# show mpls traffic-eng tunnels p2mp 2

Name: tunnel-mte2
Signalled-Name: head_mt2
Status:
  Admin: up Oper: up (Up for 00:09:37)
  Config Parameters:
    Bandwidth: 0 kbps (CT0) Priority: 7 7 Affinity: 0x0/0xffff
    Interface Bandwidth: 0 kbps
    Metric Type: TE (global)
    Fast Reroute: Not Enabled, Protection Desired: None
    Record Route: Not Enabled
    Reoptimization after affinity failure: Enabled
    Destination summary: (1 up, 0 down, 0 disabled) Affinity: 0x0/0xffff
    Auto-bw: disabled
    Destination: 10.2.2.2
      State: Up for 00:09:37
      Path options:
        path-option 1 dynamic [active]
  Current LSP:
    lsp-id: 10002 p2mp-id: 2 tun-id: 2 src: 10.1.1.1 extid: 10.1.1.1
    LSP up for: 00:09:37 (since Fri May 25 22:32:03 UTC 2018)
    Reroute Pending: No
    Inuse Bandwidth: 0 kbps (CT0)
    Number of S2Ls: 1 connected, 0 signaling proceeding, 0 down
  Destination 2.2.2.2 Signaling Status: connected
    S2L up for: 00:09:37 (since Fri May 25 22:32:03 UTC 2018)
    Sub Group ID: 1 Sub Group Originator ID: 10.1.1.1
    Path option path-option 1 dynamic (path weight 2)
    Path info (OSPF 1 area 0)
      10.0.0.5
      10.0.0.2
      10.2.2.2
  Reoptimized LSP (Install Timer Remaining 0 Seconds):
    None
  Cleaned LSP (Cleanup Timer Remaining 0 Seconds):
    None
Displayed 1 (of 101) heads, 0 (of 0) midpoints, 0 (of 0) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
```

This example shows how to verify the label assignment on the head router using the **show mpls forwarding p2mp** command.

```
RP/0/RP0/CPU0:router# show mpls forwarding p2mp
```

Local Label	Outgoing Label	Prefix or ID	Outgoing Interface	Next Hop	Bytes Switched
64106	64008	P2MP TE: 2	TenGigE0/0/0/2	10.0.0.5	0

Configuration Example for MLDP on Core

```
mpls ldp
 mldp
  logging notifications
  address-family ipv4
  !
  !
  !
```

Label Switched Multicast (LSM) Multicast Label Distribution Protocol (mLDP) based Multicast VPN (mVPN) Support

Table 5: Feature History Table

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. mLDP is supported on both core and edge routers.

When router is positioned as the core router running mLDP, it only supports the Profiles 5, 6, 7, 12, 14, and 17 irrespective of the profiles supported on the edge router.

When router is positioned as the edge router running mLDP, it only supports the Profiles 6 and 7.

For more information about the characteristics of each of the mLDP Profiles, see *Characteristics of mLDP Profiles* section in the *Implementing Layer-3 Multicast Routing on Cisco IOS XR Software* chapter of the *Multicast Configuration Guide for Cisco ASR 9000 Series Routers, IOS XR Release 6.5.x*.

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 applies the benefits of MPLS to IP multicast such as Fast ReRoute (FRR) and
- It eliminates the complexity associated PIM.

Configuring MLDP MVPN

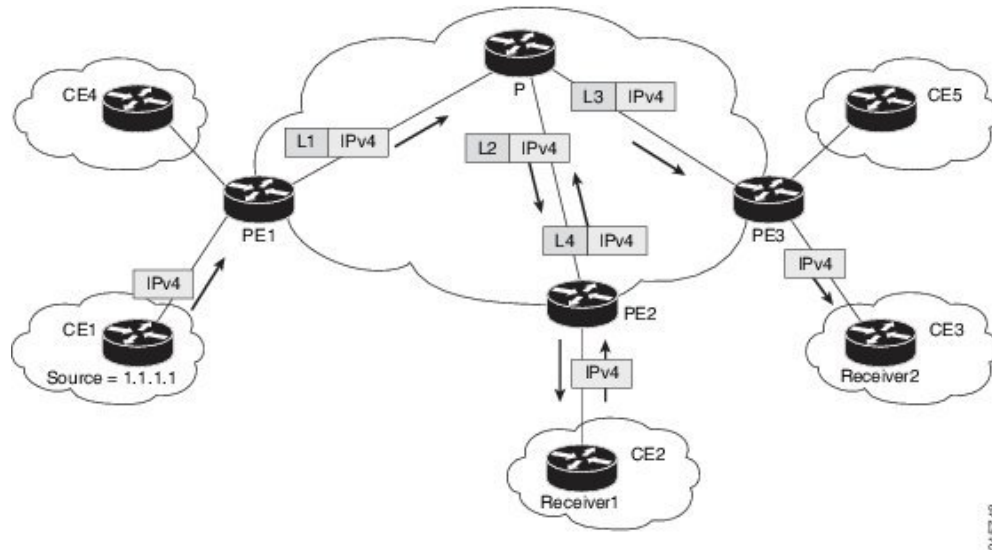
The MLDP MVPN configuration enables IPv4 multicast packet delivery using MPLS. This configuration uses MPLS labels to construct default and data Multicast Distribution Trees (MDTs). The MPLS replication is used as a forwarding mechanism in the core and edge network. For MLDP MVPN configuration to work,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



Note When a Cisco NCS5500 Series Routers, Cisco NCS560 Series Routers, or Cisco NCS540 series Routers is positioned as terminal node, it drops the IPv6 traffic that it receives from Cisco ASR 9000 Series Routers which is acting as a head node.

Figure 6: MLDP based MPLS Network for Core and Edge 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.

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 GRE/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).

Characteristics of mLDP Profiles

The characteristics of various mLDP profiles are listed in this section.

Configuration rules for profiles

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)).

The details of the MLDP profiles are discussed in the *Multicast Configuration Guide for Cisco NCS 5500 Series Routers*

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.
- IPv6 BVI is not supported.
- Netflow for MPLS-encapsulated multicast packets is not supported.
- MLDP Fast-Reroute is not supported.

Configuration Process for MLDP MVPN (Intranet)

These steps provide a broad outline of the different configuration process of MLDP MVPN for intranet:

- Enabling MPLS MLDP
 - configure
 - mpls ldp mldp
- Configuring a VRF entry
 - configure
 - vrf *vrf_name*
 - address-family ipv4/ipv6 unicast
 - import route-target route-target-ext-community
 - export route-target route-target-ext-community
- Configuring VPN ID
 - configure
 - vrf *vrf_name*
 - vpn id *vpn_id*
- Configuring MVPN Routing and Forwarding instance
 - configure
 - multicast-routing vrf *vrf_name*
 - address-family ipv4
 - mdt default mldp ipv4 *root-node*
- Configuring the Route Distinguisher
 - configure
 - router bgp *AS Number*
 - vrf *vrf_name*
 - rd *rd_value*
- Configuring Data MDTs (optional)
 - configure
 - multicast-routing vrf *vrf_name*
 - address-family ipv4
 - mdt data <1-255>
- Configuring BGP MDT address family
 - configure

- router bgp *AS Number*
- address-family ipv4 mdt
- Configuring BGP vpnv4 address family
 - configure
 - router bgp *AS Number*
 - address-family vpnv4 unicast
- Configuring BGP IPv4 VRF address family
 - configure
 - router bgp *AS Number*
 - vrf *vrf_name*
 - address-family ipv4 unicast
- Configuring PIM SM/SSM Mode for the VRFs
 - configure
 - router pim
 - vrf *vrf_name*
 - address-family ipv4
 - rpf topology route-policy *rosen_mvpn_mldp*

For each profile, a different route-policy is configured.

- Configuring route-policy
 - route-policy *rosen_mvpn_mldp*
 - set core-tree *tree-type*
 - pass
 - end-policy



Note The configuration of the above procedures depends on the profile used for each configuration.

Configure VRF MLDP In-Band Signaling on Edge Routers

To configure VRF MLDP in-band signaling (Profile 6) on edge routers, you must complete the following tasks:

1. Assign a route policy in PIM to select a reverse-path forwarding (RPF) topology.

2. Configure route policy to set the Multicast Distribution Tree (MDT) type to MLDP inband.
3. Enable MLDP-inband signaling in multicast routing.
4. Enable MPLS for MLDP.

Configuration

```

/* Assign a route policy in PIM to select a reverse-path forwarding (RPF) topology */
RP/0/RP0/CPU0:router(config)#router pim
RP/0/RP0/CPU0:router(config-pim)#vrf one
RP/0/RP0/CPU0:router(config-pim-one)#address-family ipv4
RP/0/RP0/CPU0:router(config-pim-one-ipv4)#rpf topology route-policy rpf-vrf-one

/* Configure route policy to set the MDT type to MLDP inband */
RP/0/RP0/CPU0:router(config)#route-policy rpf-vrf-one
RP/0/RP0/CPU0:router(config-rpl)#set core-tree mldp-inband
RP/0/RP0/CPU0:router(config-rpl)#end-policy

/* Enable MLDP-inband signaling in multicast routing */
RP/0/RP0/CPU0:router(config)#multicast-routing
RP/0/RP0/CPU0:router(config-mcast)#vrf one
RP/0/RP0/CPU0:router(config-mcast-one)#address-family ipv4
RP/0/RP0/CPU0:router(config-mcast-one-ipv4)#mdt source loopback 0
RP/0/RP0/CPU0:router(config-mcast-one-ipv4)#mdt mldp in-band-signaling ipv4
RP/0/RP0/CPU0:router(config-mcast-one-ipv4)#interface all enable

/* Enable MPLS MLDP */
RP/0/RP0/CPU0:router(config)#mpls ldp
RP/0/RP0/CPU0:router(config-ldp)#mldp

```

Configure Global MLDP In-band Signaling on Edge Routers

To configure global MLDP in-band signaling (Profile 7) on edge routers, you must complete the following tasks:

1. Assign a route policy in PIM to select a reverse-path forwarding (RPF) topology.
2. Configure route policy to set the MDT type to MLDP Inband.
3. Enable MLDP inband signaling in multicast routing.
4. Enable MPLS MLDP.

Configuration

```

/* Assign a route policy in PIM to select a reverse-path forwarding (RPF) topology */
RP/0/RP0/CPU0:router(config)#router pim
RP/0/RP0/CPU0:router(config-pim)#address-family ipv4
RP/0/RP0/CPU0:router(config-pim-default-ipv4)#rpf topology route-policy rpf-global
RP/0/RP0/CPU0:router(config-pim-default-ipv4)#interface TenGigE 0/0/0/21
RP/0/RP0/CPU0:router(config-pim-ipv4-if)#enable

/* Configure route policy to set the MDT type to MLDP inband */

```

```

RP/0/RP0/CPU0:router(config)#route-policy rpf-global
RP/0/RP0/CPU0:router(config-rpl)#set core-tree mldp-inband
RP/0/RP0/CPU0:router(config-rpl)#end-policy

/* Enable MLDP-inband signaling in multicast routing */

RP/0/RP0/CPU0:router(config)#multicast-routing
RP/0/RP0/CPU0:router(config-mcast)#address-family ipv4
RP/0/RP0/CPU0:router(config-mcast-default-ipv4)#interface loopback 0
RP/0/RP0/CPU0:router(config-mcast-default-ipv4-if)#enable
RP/0/RP0/CPU0:router(config-mcast-default-ipv4-if)#exit
RP/0/RP0/CPU0:router(config-mcast-default-ipv4)#mdt source loopback 0
RP/0/RP0/CPU0:router(config-mcast-default-ipv4)#mdt mldp in-band-signaling ipv4
RP/0/RP0/CPU0:router(config-mcast-default-ipv4)#interface all enable

/* Enable MPLS MLDP */

RP/0/RP0/CPU0:router(config)#mpls ldp
RP/0/RP0/CPU0:router(config-ldp)#mldp

```

Configuration Examples for MVPN Profiles

This section provides profile-wise configuration examples for the various MVPN profiles.

Configuration Examples for Inband mLDP Profiles on Core Routers

Profile-6: VRF Inband mLDP

```

router bgp 100
 mvpn
 !
 multicast-routing
  vrf v61
  address-family ipv4
   mdt source Loopback0
   mdt mtu 1600
   mdt mldp in-band-signaling ipv4
   interface all enable
  !
  address-family ipv6
   mdt mtu 1600
   mdt mldp in-band-signaling ipv4
   interface all enable
  !
 !
router pim
 vrf v61
  address-family ipv4
   rpf topology route-policy mldp-inband
  !
  address-family ipv6
   rpf topology route-policy mldp-inband
  !
 !
route-policy mldp-inband
 set core-tree mldp-inband
end-policy
!

```

Profile-7: Global Inband mLDP

```

multicast-routing
 address-family ipv4
   mdt source Loopback0
   mdt mldp in-band-signaling ipv4
   ssm range Global-SSM-Group
 interface all enable
 !
 address-family ipv6
   mdt source Loopback0
   mdt mldp in-band-signaling ipv4
   ssm range Global-SSM-Group-V6
 interface all enable
 !
 router pim
 address-family ipv4
   rpf topology route-policy mldp-inband
 !
 address-family ipv6
   rpf topology route-policy mldp-inband
 !
 !
 route-policy mldp-inband
   set core-tree mldp-inband
 end-policy
 !

```

Configuration Examples for Inband mLDP Profiles on Edge Routers

Running Configuration for VRF MLDP In-Band Signaling (Profile 6)

```

router pim
 vrf one
 address-family ipv4
   rpf topology route-policy rpf-vrf-one

 route-policy rpf-vrf-one
   set core-tree mldp-inband
 end-policy

multicast-routing
 vrf one
 address-family ipv4
   mdt source Loopback0
   mdt mldp in-band-signaling ipv4
 interface all enable

mpls ldp
 mldp

```

Running Configuration for Global MLDP In-band Signaling (Profile 7)

```

router pim
 address-family ipv4
   rpf topology route-policy rpf-global
 interface TenGigE0/0/0/21
   enable

route-policy rpf-global

```

```

    set core-tree mldp-inband
end-policy

multicast-routing
address-family ipv4
interface Loopback0
    enable
!
mdt source Loopback0
mdt mldp in-band-signaling ipv4
interface all enable
!
mpls ldp
mldp

```

Verification of MLDP Configuration on Edge Routers

Use the following commands to verify the MLDP configuration on edge routers.

To check the MLDP neighbors, use the **show mpls mldp neighbor** command.

```

RP/0/RP0/CPU0:Head# show mpls mldp neighbors
mLDP neighbor database
MLDP peer ID      : 2.2.2.2:0, uptime 07:47:59 Up,
  Capabilities    : GR, Typed Wildcard FEC, P2MP, MP2MP
  Target Adj      : No
  Upstream count  : 1
  Branch count    : 1
  LDP GR          : Enabled
                  : Instance: 1
  Label map timer : never
  Policy filter in :
  Path count      : 1
  Path(s)         : 12.1.1.2          TenGigE0/0/1/0/3.2000 LDP
  Adj list        : 12.1.1.2          TenGigE0/0/1/0/3.2000
  Peer addr list  : 2.25.32.2
                  : 2.2.2.2
                  : 11.1.1.1
                  : 12.1.1.2
                  : 13.10.1.1

```

To display the contents of the Label Information Base (LIB), use the **show mpls mldp bindings** command.

```

RP/0/RP0/CPU0:Head#show mpls mldp bindings
mLDP MPLS Bindings database

LSP-ID: 0x00001 Paths: 7 Flags:
0x00001 P2MP 5.5.5.5 [vpn6 1:1 2015:1:1::3 ff3e::1]
  Local Label: 70009
  Remote Label: 64018 NH: 12.1.1.2 Inft: TenGigE0/0/1/0/3.2000
  Remote Label: 64022 NH: 50.1.1.1 Inft: TenGigE0/0/1/3/0
  Remote Label: 30002 NH: 30.10.1.2 Inft: Bundle-Ether56
  Remote Label: 64023 NH: 60.1.1.2 Inft: HundredGigE0/0/1/1
  Remote Label: 64024 NH: 70.1.1.1 Inft: TenGigE0/0/1/2/0
  Remote Label: 64022 NH: 40.1.1.1 Inft: TenGigE0/0/0/18

```

To display the MLDP event traces, use the **show mpls mldp trace** command.

```

RP/0/RP0/CPU0:Head#show mpls mldp trace
3535 wrapping entries (631040 possible, 35584 allocated, 0 filtered, 3535 total)

```

```

May 30 23:30:21.121 MLDP GLO 0/RP0/CPU0 t6746 GEN : Trace pre-init iox success
May 30 23:30:21.121 MLDP GLO 0/RP0/CPU0 t6746 GEN : Debug pre-init iox success
May 30 23:30:21.121 MLDP GLO 0/RP0/CPU0 t6746 GEN : API pre-init iox success
May 30 23:30:21.121 MLDP GLO 0/RP0/CPU0 t6746 GEN : Bitfield pre-init iox success
May 31 12:08:39.465 MLDP GLO 0/RP0/CPU0 t6746 GEN : mldp_evm 0x563de8f01698 allocated
May 31 12:08:39.465 MLDP GLO 0/RP0/CPU0 t6746 GEN : EVM init iox success
May 31 12:08:39.472 MLDP GLO 0/RP0/CPU0 t6746 GEN : Registered EDM on active success
May 31 12:08:39.472 MLDP GLO 0/RP0/CPU0 t6746 GEN : EDM Ac/St init iox again
May 31 12:08:39.472 MLDP GLO 0/RP0/CPU0 t6746 GEN : Registered EDM Location on active
success
May 31 12:08:39.472 MLDP GLO 0/RP0/CPU0 t6746 GEN : EDM Loc init iox success
May 31 12:08:39.475 MLDP GLO 0/RP0/CPU0 t6746 GEN : LMRIB init iox success
May 31 12:08:39.475 MLDP GLO 0/RP0/CPU0 t18944 MRIB : MRIB connection established
May 31 12:08:39.475 MLDP GLO 0/RP0/CPU0 t6746 GEN : Interface manager init iox success
May 31 12:08:39.475 MLDP GLO 0/RP0/CPU0 t6746 GEN : Async init iox success
May 31 12:08:39.475 MLDP GLO 0/RP0/CPU0 t6746 GEN : Boolean init iox success
May 31 12:08:39.475 MLDP GLO 0/RP0/CPU0 t6746 GEN : Timers init iox success
May 31 12:08:39.479 MLDP GLO 0/RP0/CPU0 t6746 GEN : RUMP init iox success
May 31 12:08:39.479 MLDP GLO 0/RP0/CPU0 t6746 GEN : Chunks init iox success
May 31 12:08:39.509 MLDP ERR 0/RP0/CPU0 t6746 RIB : RIB not ready
May 31 12:08:39.509 MLDP ERR 0/RP0/CPU0 t6746 RIB : RIB not ready
May 31 12:08:39.512 MLDP GLO 0/RP0/CPU0 t6746 GEN : mldp_ens_event_ctx_chunk is NULL
May 31 12:08:39.512 MLDP GLO 0/RP0/CPU0 t6746 GEN : Context Table init iox success
May 31 12:08:39.512 MLDP GLO 0/RP0/CPU0 t6746 GEN : mldp_rib_main_evm 0x563de8fd23e8
allocated
May 31 12:08:39.512 MLDP GLO 0/RP0/CPU0 t6746 GEN : RIB Thread EVM init rib success
May 31 12:08:39.512 MLDP GLO 0/RP0/CPU0 t6746 GEN : RIB Thread Chunk init rib success
May 31 12:08:39.512 MLDP GLO 0/RP0/CPU0 t6746 GEN : RIB Thread queue init rib success
May 31 12:08:39.512 MLDP GLO 0/RP0/CPU0 t6746 RIB : Bound to RIB, fd: 354

```

MLDP Loop-Free Alternative Fast Reroute

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). 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 precomputed, 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 (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 on the changed network topology (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.

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 also 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.

For information on use cases, see the [MLDP with Flex- Algo in Service Provider Networks White Paper](#).

MLDP LFA FRR with Flexible Algorithm

The MLDP LFA FRR with Flexible Algorithm uses the segment routed (SR) LFA FRR-selected primary and backup paths to the peers and emulates a multicast distribution tree, instead of multicast label-switched paths (LSP). It helps in having a more efficient FRR with low-latency routing, live-live disjoint paths, or constraining multicast flows to a specific region. Interior Gateway Protocol (IGP) calculates LFA path for each learned node SID within the IGP domain.

Supported MLDP Profiles

The list of supported MLDP profiles is:

Advantages of LFA FRR

The following are the advantages of the LFA FRR solution:

- The backup path for the traffic flow is pre-computed.
- 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 10s of milliseconds can be achieved. Prefix independency is the key for a fast switchover in the forwarding table.
- 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

MLDP LFA FRR - Features

The following are the features of mLDP LFA FRR solution:

- Supports both IPv4 and IPv6 traffic
- Supports Profile 14 mLDP profile
- Supports the LAG interfaces and sub-interfaces in the core
- Supports both ISIS and OSPF routing protocols
- 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



Note ECMP primary and backup paths are not supported.

Limitations of LFA FRR

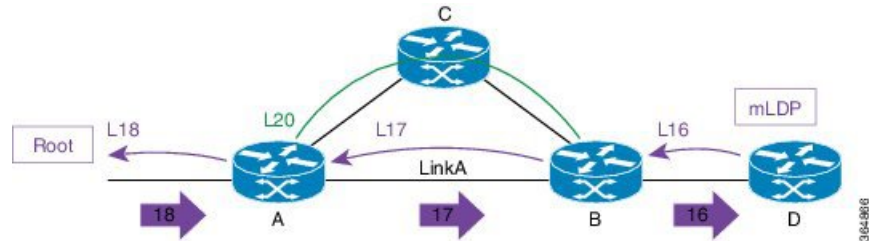
The following are some of the known limitations of the LFA FRR solution:

- When a failure that is more extensive than that which the alternate was intended to protect occurs, there is the possibility of temporarily looping traffic (micro looping until Control Plane Convergence).
- Topology dependent. For example, either MPLS or MLDP dependent.
- Complex implementation.
- The solution is currently not supported on all platforms.

MLDP LFA FRR - Working

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

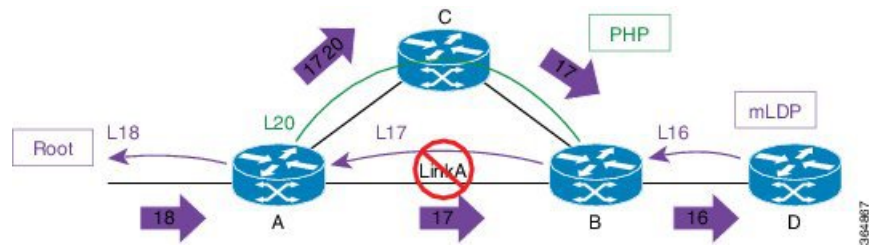
Figure 7: MLDP LFA FRR - Setup



In this figure:

1. Router A is the source provider edge router, and the next Hop is Router B.
2. 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.
3. MLDP LSP is build from D, B, and A towards the root.
- 4.

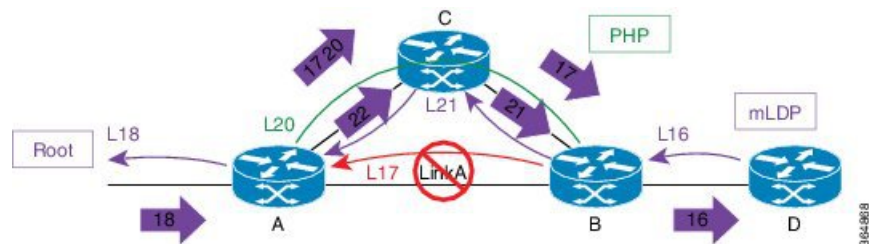
Figure 8: Link Failure



When a link failure occurs on Link A:

1. Traffic over Link A is rerouted over the backup tunnel by imposing the traffic engineering (TE) label 20 towards mid Router C.
2. Router C performs penultimate hop popping (PHP) and removes the outer label 20.
3. Router B receives the mLDP packets with label 17 and forwards to Router D.

Figure 9: Re-optimization - Make-Before-Break



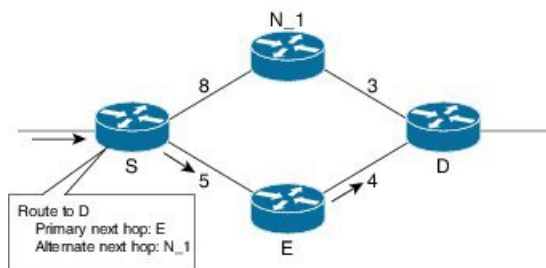
During re-optimization:

1. 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.
2. Router A forwards packets natively with old label 17 and also new label 22.
3. Router B drops traffic carried from new label 22 and forwards traffic with label 17.
4. Router B uses make-before-break (MBB) trigger to switch from either physical or bundle interface to native, label 17 to 21.
5. Router B prunes off the physical or bundle interface with a label withdraw to router A.

MLDP LFA FRR - Behavior

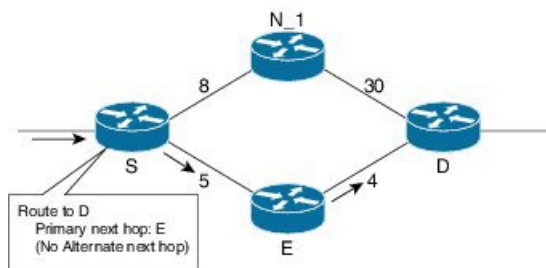
In the following scenarios, S is source router, D is the destination router, E is primary next hop, and N_1 is the alternative next hop.

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.

LFA Criteria

In the above example, the LFA criteria of whether N is to be the LFA next-hop is met, when:

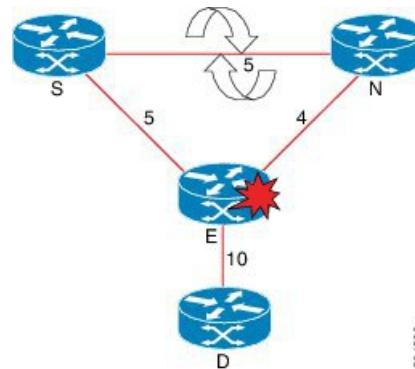
Cost of path (N_1, D) < Cost of path (N_1, S) + Cost of path (E, S) + Cost of path (D, E)

Downstream Path criteria, which is subset of LFA, is met when:

Cost of path (N_1, D) < Cost of path (E, S) + Cost of path (D, E)

Link Protecting LFA

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 cannot use S. Therefore, N would have no alternate and would discard the traffic, thus avoiding the micro-looping.

Node Protecting LFA

Link and node protecting LFA guarantees protection against either link or node failure. Depending on the protection available at the downstream node, the downstream path provides protection against a link failure; however, it does not provide protection against a node failure, thereby preventing micro looping.

The criteria for LFA selection priority is that: the Link and Node protecting LFA is greater than the Link Protecting Downstream is greater than the Link Protecting LFA.

Configure MLDP Route-Policy for Flexible Algorithm FRR

Using the MLDP route-policy option, you can enable the FRR for selected LSPs. You can enable FRR only for flexible algorithm-based LSPs, non-flexible algorithm-based LSPs, or for both. This route policy helps when you have a large number of flows and you have enabled FRR on all of them. If you have some flows that are critical and need FRR and other flows without FRR, you can apply the customization using the route policy.

If you do not configure any route policy, then the FRR is enabled on all LSPs.

The following example shows how to configure MLDP route-policy with flexible algorithm and apply the same:

```
Router#config
Router(config)#route-policy mldp-fa-frr
Router(config-rpl)#if mldp flex-algo?
<128-255>
Algorithm number
any
Any Algorithm
```

```

Router (config-rpi)#if mldp flex-algo 128 then
Router (config-rpl-if) #pass
Router (config-rpl-if) Hendif
Router (config-rpl)#if mldp flex-algo any then
Router (config-rpl-if) #pass
Router (config-rpl-if) Hendif
Router (config-rpl) Hendif-policy

Router#config
Router (config)#mpls ldp mldp address-family ipv4
Router (config-ldp-mldp-af)#forwarding recursive route-policy mldp-fa-frr
Router (config-ldp-mldp-af)#

```

Configurations to Enable LFA FRR

Key Configurations To Enable LFA FRR

The key configurations to enable LFA FRR feature include:

- Router OSPF configuration

The various configurations available under OSPF are:

- Enabling Per-Prefix LFA
- Excluding Interface from Using Backup
- Adding Interfaces to LFA Candidate List
- Restricting LFA Candidate List
- Limiting Per-Prefix Calculation by Prefix Priority
- Disabling Load Sharing of Backup Paths

- Router ISIS configuration
- Bidirectional Forwarding Detection (BFD) configuration
- MPLS configuration

The various configurations available under MPLS are:

- MBB (MLDP) configuration
- Make Before Break (MBB) Delay X <sec> Delete Y <sec>
- Configure FRR Timer for Scale Number of MLDP LSPs

Configuring Router OSPF LFA FRR

In OSPF configuration, configure per-prefix link based LFA to enable the LFA FRR feature. The detailed configuration steps with an example follows:

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration mode.

Step 2 **router ospf 0**

Example:

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

Enters the OSPF configuration mode.

Step 3 **area 0**

Example:

```
RP/0/RP0/CPU0:router (config-ospf)# area 0
```

Enters the area submode under the OSPF configuration mode.

Step 4 **interface Bundle-Ether10**

Example:

```
RP/0/RP0/CPU0:router (config-ospf-ar)# interface Bundle-Ether10
```

Enters the interface submode configuration, under OSPF area submode.

Step 5 **fast-reroute per-prefix**

Example:

```
RP/0/RP0/CPU0:router (config-ospf-ar-if)# fast-reroute per-prefix
```

Enables the per-prefix mode of LFA calculation on the specified interface.

Step 6 **commit**

Example

Example: Configuration to Enable OSPF LFA FRR

```
!
router ospf {tag}
area {area-id}
interface {interface}
fast-reroute per-prefix enable
!
!
```

Enabling Per Prefix LFA

Lists the steps required to enable per-prefix LFA mode of LFA calculation using OSPF configuration.

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration mode.

Step 2 **router ospf 0****Example:**

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

Enters the OSPF configuration mode.

Step 3 **area 0****Example:**

```
RP/0/RP0/CPU0:router(config-ospf)# area 0
```

Enters the area sub mode under the OSPF configuration mode.

Step 4 **interface Bundle-Ether10****Example:**

```
RP/0/RP0/CPU0:router(config-ospf-ar)# interface Bundle-Ether10
```

Enters the interface sub mode configuration, under OSPF area sub mode.

Step 5 **fast-reroute per-prefix****Example:**

```
RP/0/RP0/CPU0:router(config-ospf-ar-if)# fast-reroute per-prefix
```

Enables the per-prefix mode of LFA backup path calculation on the specified interface.

Step 6 **fast-reroute per-prefix remote-lfa tunnel mpls-ldp****Example:**

```
RP/0/RP0/CPU0:router(config-ospf-ar-if)# fast-reroute per-prefix remote-lfa tunnel mpls-ldp
```

Enables the remote LFA on the specified interface.

Step 7 **commit****Adding Interfaces to LFA Candidate List**

Lists the steps required to add an interface to the LFA candidate list.

Step 1 **configure****Example:**

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration mode.

Step 2 **router ospf 0****Example:**

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

Enters the OSPF configuration mode.

Step 3 **area 0**

Example:

```
RP/0/RP0/CPU0:router(config-ospf)# area 0
```

Enters the area submode under the OSPF configuration mode.

Step 4 interface Bundle-Ether10**Example:**

```
RP/0/RP0/CPU0:router(config-ospf-ar)# interface Bundle-Ether10
```

Enters the interface submode configuration, under OSPF area submode.

Step 5 fast-reroute per-prefix lfa-candidate**Example:**

```
RP/0/RP0/CPU0:router(config-ospf-ar-if)# fast-reroute per-prefix lfa-candidate Bundle-Ether10
```

Adds the listed interface to the LFA candidate list to compute backup paths.

Note By default, no interfaces are on the LFA candidate list.

Step 6 commit

Exclude Interface from Backup

Lists the steps required to exclude an interface from using backup paths for LFA calculation using OSPF configuration.

Step 1 configure**Example:**

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration mode.

Step 2 router ospf 0**Example:**

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

Enters the OSPF configuration mode.

Step 3 area 0**Example:**

```
RP/0/RP0/CPU0:router(config-ospf)# area 0
```

Enters the area submode under the OSPF configuration mode.

Step 4 interface Bundle-Ether10**Example:**

```
RP/0/RP0/CPU0:router(config-ospf-ar)# interface Bundle-Ether10
```

Enters the interface submode configuration, under OSPF area submode.

Step 5 fast-reroute per-prefix exclude**Example:**

```
RP/0/RP0/CPU0:router(config-ospf-ar-if)# fast-reroute per-prefix exclude Bundle-Ether10
```

Excludes the specific listed interface while calculating the LFA backup paths.

Note By default, no interfaces are excluded from the LFA backup path calculation.

Step 6 commit**Restricting the Backup Interfaces to the LFA Candidate List**

Lists the steps required to restrict the backup interface to the LFA candidate list.

Step 1 configure**Example:**

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration mode.

Step 2 router ospf 0**Example:**

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

Enters the OSPF configuration mode.

Step 3 area 0**Example:**

```
RP/0/RP0/CPU0:router(config-ospf)# area 0
```

Enters the area submode under the OSPF configuration mode.

Step 4 interface Bundle-Ether10**Example:**

```
RP/0/RP0/CPU0:router(config-ospf-ar)# interface Bundle-Ether10
```

Enters the interface submode configuration, under OSPF area submode.

Step 5 fast-reroute per-prefix use-candidate-only**Example:**

```
RP/0/RP0/CPU0:router(config-ospf-ar-if)# fast-reroute per-prefix use-candidate-only
```

Restricts the calculation of the backup paths to only the interfaces listed on the LFA candidate list.

Note By default, the **fast-reroute per-prefix use-candidate-only** is disabled.

Step 6 commit

Limiting the Per-Prefix Calculation by Prefix-Priority

Lists the steps required to limit the per-prefix calculation by prefix-priority.

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration mode.

Step 2 **router ospf 0**

Example:

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

Enters the OSPF configuration mode.

Step 3 **area 0**

Example:

```
RP/0/RP0/CPU0:router (config-ospf)# area 0
```

Enters the area submode under the OSPF configuration mode.

Step 4 **interface Bundle-Ether10**

Example:

```
RP/0/RP0/CPU0:router (config-ospf-ar)# interface Bundle-Ether10
```

Enters the interface submode configuration, under OSPF area submode.

Step 5 **fast-reroute per-prefix prefix-limit {priority}**

Example:

```
RP/0/RP0/CPU0:router (config-ospf-ar-if)# fast-reroute per-prefix prefix-limit  
{priority}
```

Limits the per-prefix LFA backup path calculation by prefix-priority.

Only prefixes with the same or higher priority as specified are subjected to the per-prefix backup paths calculation.

Note By default, backup path is calculated for prefixes regardless of their priority.

Step 6 **commit**

Disabling Load Sharing of the Backup Paths

Lists the steps required to disable the load sharing of the backup paths.

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration mode.

Step 2 **router ospf 0****Example:**

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

Enters the OSPF configuration mode.

Step 3 **area 0****Example:**

```
RP/0/RP0/CPU0:router(config-ospf)# area 0
```

Enters the area submode under the OSPF configuration mode.

Step 4 **interface Bundle-Ether10****Example:**

```
RP/0/RP0/CPU0:router(config-ospf-ar)# interface Bundle-Ether10
```

Enters the interface submode configuration, under OSPF area submode.

Step 5 **fast-reroute per-prefix load-sharing disable****Example:**

```
RP/0/RP0/CPU0:router(config-ospf-ar-if)# fast-reroute per-prefix load-sharing disable
```

Disables the load sharing of the backup paths.

It is used to control the load-balancing of the backup paths on a per-prefix basis.

Note By default, load-balancing of per-prefixes across all backup paths is enabled.

Step 6 **commit**

Configuring Router ISIS LFA FRR

In ISIS configuration, configure fast-reroute per-prefix to enable the LFA FRR feature.

Step 1 **configure****Example:**

```
RP/0/RP0/CPU0:router# configure
```

Enters the global configuration mode.

Step 2 **router isis instance id****Example:**

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

Enables IS-IS routing for the specified routing instance, and places the router in router configuration mode.

Step 3 **net network-entity-title****Example:**

```
RP/0/RP0/CPU0:router(config-isis)# net 49.0001.0000.0000.0001.00
```

Configures network entity titles (NETs) for the routing instance.

- Specify a NET for each routing instance if you are configuring multi-instance IS-IS.
- This example, configures a router with area ID 49.0001.0000.0000 and system ID 0000.0001.0000.0000
- To specify more than one area address, specify additional NETs. Although the area address portion of the NET differs for all of the configured items, the system ID portion of the NET must match exactly.

Step 4 **address-family ipv4 unicast**

Example:

```
RP/0/RP0/CPU0:router(config-isis)# address-family ipv4 unicast
```

Enters the address-family submode. This is supported only on unicast topologies.

Step 5 **commit**

Step 6 **interface GigabitEthernet0/0/1/1**

Example:

```
RP/0/RP0/CPU0:router(config-isis-af)# interface GigabitEthernet0/0/1/1
```

Enters the interface submode.

Step 7 **address-family ipv4 unicast**

Example:

```
RP/0/RP0/CPU0:router(config-isis-if-af)# address-family ipv4 unicast
```

Enters the address-family submode. This is supported on unicast topologies only.

Step 8 **fast-reroute per-prefix remote-lfa tunnel mpls-ldp**

Example:

```
RP/0/RP0/CPU0:router(config-isis-if-af)# fast-reroute per-prefix remote-lfa tunnel mpls-ldp
```

Enables LFA FRR remote LFA.

Step 9 **commit**

Configuring Bidirectional Forwarding Detection

When a local interface is down, that is, due to either a fiber cut or because of interface shutdown configuration is run, it can take a long delay in the order of tens of milliseconds for the remote peer to detect the link disconnection; so, to quickly detect the remote shut on physical port or on bundle interfaces, the physical port and bundle interfaces must be running Bidirectional Forwarding Detection (BFD) to ensure faster failure detection.

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration mode.

Step 2 `router ospf instance id`

Example:

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

Enters the OSPF routing configuration mode.

Step 3 `nsr`

Example:

```
RP/0/RP0/CPU0:router(config-ospf)# nsr
```

Enables nonstop routing.

Step 4 `router-id instance id`

Example:

```
RP/0/RP0/CPU0:router(config-ospf)# router-id 21.21.21.21
```

Specifies the router ID of the particular IPv4 address.

Step 5 `nsf instance name`

Example:

```
RP/0/RP0/CPU0:router(config-ospf)# interface cisco
```

Enters the interface submode configuration, under OSPF mode.

Step 6 `address-family ipv4 unicast`

Example:

```
RP/0/RP0/CPU0:router(config-ospf)# address-family ipv4 unicast
```

Enters the address-family submode. This is supported only on unicast topologies.

Step 7 `area instance id`

Example:

```
RP/0/RP0/CPU0:router(config-ospf-af)# area 0
```

Enters the area submode under the OSPF configuration mode.

Step 8 `bfd minimum-interval value`

Example:

```
RP/0/RP0/CPU0:router(config-ospf-af)# bfd minimum-interval 3
```

Sets the bidirectional forwarding detection minimum-interval value to 3.

Step 9 `bfd fast-detect`

Example:

```
RP/0/RP0/CPU0:router(config-ospf-af)# bfd fast-detect
```

Configures bidirectional forwarding detection to fast detection.

Step 10 `bfd multiplier value`

Example:

```
RP/0/RP0/CPU0:router(config-ospf-af)# bfd multiplier 2
```

Configures bidirectional forwarding detection to fast detection.

Step 11 **fast-reroute per-prefix**

Example:

```
RP/0/RP0/CPU0:router(config-ospf-af) # fast-reroute per-prefix
```

Enables the per-prefix mode of LFA calculation on the specified interface.

Step 12 **mpls traffic-eng**

Example:

```
RP/0/RP0/CPU0:router(config-ospf-af) # mpls traffic-eng
```

Configures the MPLS TE under the OSPF area.

Step 13 **interface *instance id***

Example:

```
RP/0/RP0/CPU0:router(config-ospf-af) # interface Bundle-Ether100.1
```

Configures the specified interface.

Step 14 **bfd fast-detect**

Example:

```
RP/0/RP0/CPU0:router(config-ospf-af-if) # bfd fast-detect
```

Configures bidirectional forwarding detection to fast detection.

Step 15 **fast-reroute per-prefix**

Example:

```
RP/0/RP0/CPU0:router(config-ospf-af-if) # fast-reroute per-prefix
```

Enables the per-prefix mode of LFA calculation on the specified interface.

Step 16 **commit**

Step 17 **interface *instance id***

Example:

```
RP/0/RP0/CPU0:router(config-ospf-af) # interface Bundle-Ether100.1
```

Configures the specified interface.

Step 18 **bfd fast-detect**

Example:

```
RP/0/RP0/CPU0:router(config-ospf-af-if) # bfd fast-detect
```

Configures bidirectional forwarding detection to fast detection.

Step 19 **fast-reroute per-prefix**

Example:

```
RP/0/RP0/CPU0:router(config-ospf-af-if) # fast-reroute per-prefix
```

Enables the per-prefix mode of LFA calculation on the specified interface.

Step 20 **commit**

Step 21 interface loopback0**Example:**

```
RP/0/RP0/CPU0:router(config)# interface loopback0
```

Example

```
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
  bfd multiplier 2
  fast-reroute per-prefix
 mpls traffic-eng
 interface Bundle-Ether100.1
  bfd fast-detect
  fast-reroute per-prefix
 !
 interface Bundle-Ether100.2
  bfd fast-detect
  fast-reroute per-prefix
 !
 interface Loopback0
 !
```

In the above configuration example, **bfd minimum-interval 3** and **bfd multiplier 2** is configured; this means, that when a core-facing interface of a remote peer is down, the router detects this disconnect event in as short a time as 6 milliseconds.

Configuring MPLS LFA FRR

In MPLS configuration, configure session protection to support LFA FRR feature. The detailed configuration steps and an example follows.

Step 1 configure**Example:**

```
RP/0/RP0/CPU0:router# configure
```

Enters global configuration mode.

Step 2 router ospf 0**Example:**

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

Enters the LDP configuration mode.

Step 3 nsr**Example:**

```
RP/0/RP0/CPU0:router(config-ldp)# nsr
```

Configures non-stop routing.

Step 4 **graceful-restart**

Example:

```
RP/0/RP0/CPU0:router(config-ldp)# graceful-restart
```

Restarts the interface.

Step 5 **router-id 20.20.20.20**

Example:

```
RP/0/RP0/CPU0:router(config-ldp)# router-id 20.20.20.20
```

Configures a router-id for the LDP process.

Step 6 **session protection**

Example:

```
RP/0/RP0/CPU0:router(config-ldp)# session protection
```

Enables LFA FRR in the per-prefix mode.

Step 7 **address-family ipv4**

Example:

```
RP/0/RP0/CPU0:router(config-ldp)# address-family ipv4
```

Enters address family configuration mode.

Step 8 **forwarding recursive**

Example:

```
RP/0/RP0/CPU0:router(config-ldp)# forwarding recursive
```

Enables MLDP LFA FRR.

Step 9 **make-before-break delay 60 10**

Example:

```
RP/0/RP0/CPU0:router(config-ldp)# make-before-break delay 60 10
```

Configures make before break (MBB) delay.

Step 10 **commit**

Example

Example: Configuration to enable MLDP LFA FRR

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

```

address-family ipv4
make-before-break delay 60 10
!
!

```

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.

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters global configuration mode.
Step 2	mpls ldp Example: RP/0/RP0/CPU0:router(config)# mpls ldp	Enters the LDP configuration mode.
Step 3	log Example: RP/0/RP0/CPU0:router(config-ldp)# log	Enters the log sub mode under the LDP sub mode.
Step 4	neighbor Example: RP/0/RP0/CPU0:router(config-ldp-log)# neighbor	Configures the specified neighbor to the MLDP policy.
Step 5	nsr Example: RP/0/RP0/CPU0:router(config-ldp-log)# nsr	Configures non-stop routing.
Step 6	graceful-restart Example: RP/0/RP0/CPU0:router(config-ldp)# graceful-restart	Restarts the interface.
Step 7	commit	
Step 8	mldp Example: RP/0/RP0/CPU0:router(config-ldp)# mldp	Enters the MLDP sub mode under the LDP sub mode.
Step 9	address-family ipv4 Example:	Enters the Address Family sub mode under the MLDP sub mode.

	Command or Action	Purpose
	RP/0/RP0/CPU0:router (config-ldp-ml dp-af) # address-family ipv4	
Step 10	forwarding recursive Example: RP/0/RP0/CPU0:router (config-ldp-ml dp-af) # forwarding recursive	Enables LFA FRR.
Step 11	make-before-break delay {seconds} Example: RP/0/RP0/CPU0:router (config-ldp-ml dp-af) # make-before-break delay 60	Sets the make-before-break delay to the specified number of seconds.
Step 12	commit	

Example

Example Configuration Example of MBB for LFA FRR

```

mpls ldp
log
neighbor
nsr
graceful-restart
!
ml dp
address-family ipv4
forwarding recursive
make-before-break delay 60
!
!

```

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.

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters global configuration mode.
Step 2	mpls ldp Example: RP/0/RP0/CPU0:router (config) # mpls ldp	Enters the LDP configuration mode.

	Command or Action	Purpose
Step 3	mldp Example: RP/0/RP0/CPU0:router(config-ldp)# mldp	Enters the MLDP sub mode under the LDP sub mode.
Step 4	address-family ipv4 Example: RP/0/RP0/CPU0:router(config-ldp-mldp)# address-family ipv4	Enters the Address Family sub mode under the MLDP sub mode.
Step 5	make-before-break delay {seconds} Example: RP/0/RP0/CPU0:router(config-ldp-mldp-af)# make-before-break delay 90	Sets the Make Before Break delay to 90 seconds.
Step 6	make-before-break delay {seconds} delete {seconds} Example: RP/0/RP0/CPU0:router(config-ldp-mldp-af)# make-before-break delay 90 delete 60	Sets the Make Before Break delete delay to 60 seconds.
Step 7	commit	

Example

Example: Make Before Break Delay And Delete

```

mldp
address-family ipv4
make-before-break delay ?
<0-600> Forwarding delay in seconds
make-before-break delay 90 ?
<0-60> Delete delay in seconds
make-before-break delay 90 delete 60
!
!

```

In the above 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.



CHAPTER 3

Implementing Layer-2 Multicast with IGMP Snooping

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 on Cisco NCS 5500 Series Routers.

Table 6: Feature History for IGMP Snooping

Release	Modification
Release 6.6.25	This feature was introduced.

- [Implementing Layer-2 Multicast with IGMP Snooping, on page 73](#)
- [Prerequisites for IGMP Snooping, on page 73](#)
- [Restrictions for IGMP Snooping, on page 73](#)
- [Information About IGMP Snooping, on page 74](#)
- [How to Configure IGMP Snooping, on page 79](#)
- [Configuration Examples for IGMP Snooping, on page 86](#)
- [Additional References, on page 92](#)

Implementing Layer-2 Multicast with IGMP Snooping

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.

Prerequisites for IGMP Snooping

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

Restrictions for IGMP Snooping

- IGMP snooping is supported only under L2VPN bridge domains.

- Explicit host tracking (an IGMPv3 snooping feature) is not supported.
- IPv6 Multicast Listener Discovery (MLD) snooping is not supported.
- IGMPv1 is not supported.
- IGMP snooping over Access Pseudo-wire is not supported.



Note IPv4 multicast is supported for a multicast source that is behind the BVI interface. For example, the below configuration shows how to configure source behind BVI for IPv4 multicast:

```
l2vpn
bridge group 1
  bridge-domain 1
    multicast-source ipv4
    igmp snooping profile grp1
  !
  interface TenGigE0/0/0/3.32
  !
  routed interface BVI1
```

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.

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 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.

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
- Stateful Switch-Over (SSO)
- 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.
- 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 Cisco NCS 5500 Series Routers randomly nominates a single port from the bundle to carry the multicast traffic.

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 Host Ports

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 and host ports. [Table 7: Multicast Traffic Handling for an IGMPv2 Querier, on page 76](#) describes traffic handling for an IGMPv2 querier. [Table 8: Multicast Traffic Handling for an IGMPv3 Querier, on page 76](#) applies to an IGMPv3 querier.

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 7: Multicast Traffic Handling for an IGMPv2 Querier

Traffic Type	Received on Host Ports
IP multicast source traffic	Forwards to all mrouter ports and to host ports that indicate interest.
IGMP general queries	—
IGMP group-specific queries	Dropped
IGMPv2 joins	Examines (snoops) the reports. <ul style="list-style-type: none"> • 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
IGMPv2 leaves	Invokes last member query processing.

Table 8: Multicast Traffic Handling for an IGMPv3 Querier

Traffic Type	Received on Host Ports
IP multicast source traffic	Forwards to all mrouter ports and to host ports that indicate interest.
IGMP general queries	—
IGMP group-specific queries	—
IGMPv2 joins	Handles as IGMPv3 IS_EX{} reports.
IGMPv3 reports	<ul style="list-style-type: none"> • 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.

Information About 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. An empty profile enables the default configuration options and settings for IGMP snooping, as listed in the [Default IGMP Snooping Configuration Settings, on page 78](#).

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. Profiles can be attached to ports under the bridge, one profile per port.
- 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 9: 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 mrouter configured; dynamic discovery occurs by default.
	router guard	Disabled
	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:

$GMI = (\text{robustness-variable} * \text{query-interval}) + \text{maximum-response-time}$

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

SUMMARY STEPS

1. **configure**
2. **igmp snooping profile** *profile-name*

3. Optionally, add commands to override default configuration values.
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	igmp snooping profile <i>profile-name</i> Example: <pre>RP/0/RP0/CPU0:router(config)# igmp snooping profile default-bd-profile</pre>	<p>Enters IGMP snooping profile configuration mode and creates a named profile.</p> <p>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.</p>
Step 3	Optionally, add commands to override default configuration values.	<p>If you are creating a bridge domain profile, consider the following:</p> <ul style="list-style-type: none"> • 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. <p>If you are creating a port-specific profile, consider the following:</p> <ul style="list-style-type: none"> • 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. <p>You can detach a profile, change it, and reattach it to add commands to a profile at a later time.</p>
Step 4	commit	

Where to Go Next

You must attach a profile to a bridge domain or to a port to have it take effect. See one of the following tasks:

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.

SUMMARY STEPS

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*
4. **bridge-domain** *bridge-domain-name*
5. **multicast-source ipv4**
6. **igmp snooping profile** *profile-name*
7. **commit**
8. **show igmp snooping bridge-domain detail**
9. **show l2vpn bridge-domain detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	l2vpn Example: RP/0/RP0/CPU0:router(config)# l2vpn	Enters Layer 2 VPN configuration mode.
Step 3	bridge group <i>bridge-group-name</i> 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 <i>bridge-domain-name</i> 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	multicast-source ipv4 Example: RP/0/RP0/CPU0:router(config)# multicast-source ipv4	Configures Layer 2 multicast routes with IGMP snooping.

	Command or Action	Purpose
Step 6	igmp snooping profile <i>profile-name</i> Example: <pre>RP/0/RP0/CPU0:router(config-l2vpn-bg-bd)# igmp snooping profile default-bd-profile</pre>	Attaches the named IGMP snooping profile to the bridge domain, enabling IGMP snooping on the bridge domain.
Step 7	commit	
Step 8	show igmp snooping bridge-domain detail Example: <pre>RP/0/RP0/CPU0:router# show igmp snooping bridge-domain detail</pre>	(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: <pre>RP/0/RP0/CPU0:router# show l2vpn bridge-domain</pre>	(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.

SUMMARY STEPS

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*
4. **bridge-domain** *bridge-domain-name*
5. **no igmp snooping disable**
6. **commit**
7. **show igmp snooping bridge-domain detail**
8. **show l2vpn bridge-domain detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	

	Command or Action	Purpose
Step 2	l2vpn Example: <pre>RP/0/RP0/CPU0:router(config)# l2vpn</pre>	Enters Layer 2 VPN configuration mode.
Step 3	bridge group <i>bridge-group-name</i> Example: <pre>RP/0/RP0/CPU0:router(config-l2vpn)# bridge group GRP1</pre>	Enters Layer 2 VPN bridge group configuration mode for the named bridge group.
Step 4	bridge-domain <i>bridge-domain-name</i> Example: <pre>RP/0/RP0/CPU0:router(config-l2vpn-bg)# bridge-domain ISP1</pre>	Enters Layer 2 VPN bridge group bridge domain configuration mode for the named bridge domain.
Step 5	no igmp snooping disable Example: <pre>RP/0/RP0/CPU0:router(config-l2vpn-bg-bd)# no igmp snooping disable</pre>	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: <pre>RP/0/RP0/CPU0:router# show igmp snooping bridge-domain detail</pre>	(Optional) Verifies that IGMP snooping is disabled on a bridge domain.
Step 8	show l2vpn bridge-domain detail Example: <pre>RP/0/RP0/CPU0:router# show l2vpn bridge-domain</pre>	(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.

SUMMARY STEPS

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*
4. **bridge-domain** *bridge-domain-name*
5. **interface** *interface-type interface-number*
6. **multicast-source ipv4**
7. Do one of the following:
 - **igmp snooping profile** *profile-name*
 - **no igmp snooping**
8. **commit**
9. **show igmp snooping bridge-domain detail**
10. **show l2vpn bridge-domain detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	l2vpn Example: RP/0/RP0/CPU0:router(config)# l2vpn	Enters Layer 2 VPN configuration mode.
Step 3	bridge group <i>bridge-group-name</i> 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 <i>bridge-domain-name</i> 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 <i>interface-type interface-number</i> Example: RP/0/RP0/CPU0:router(config-l2vpn-bg-bd)# interface gig 1/1/1/1	Enters Layer 2 VPN bridge group bridge domain interface configuration mode for the named interface or PW.
Step 6	multicast-source ipv4 Example:	Configures L2 multicast routes in L2 multicast with IGMP Snooping.

	Command or Action	Purpose
	RP/0/RP0/CPU0:router (config)# multicast-source ipv4	
Step 7	Do one of the following: <ul style="list-style-type: none"> • igmp snooping profile <i>profile-name</i> • 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 8	commit	
Step 9	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 10	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

SUMMARY STEPS

1. **configure**
2. **show l2vpn forwarding bridge-domain** [*bridge-group-name:bridge-domain-name*] **mroute ipv4** [**group** *group_IPAddress*] [**hardware** {**ingress** | **egress**}] [**detail**]**location** *node-id*
3. **show l2vpn forwarding bridge-domain** [*bridge-group-name:bridge-domain-name*] **mroute ipv4** **summary** **location** *node-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	
Step 2	show l2vpn forwarding bridge-domain [<i>bridge-group-name:bridge-domain-name</i>] mroute ipv4 [group <i>group_IPAddress</i>] [hardware { ingress egress }] [detail] location <i>node-id</i>	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.

	Command or Action	Purpose
	<p>Example:</p> <pre>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</pre>	If these routes are not as expected, check the control plane configuration and correct the corresponding IGMP snooping profiles.
Step 3	<p>show l2vpn forwarding bridge-domain [<i>bridge-group-name:bridge-domain-name</i>] mroute ipv4 summary location <i>node-id</i></p> <p>Example:</p> <pre>RP/0/RP0/CPU0:router# show l2vpn forwarding bridge-domain bg1:bd1 mroute ipv4 summary location 0/3/CPU0</pre>	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 bridge domains on Cisco NCS 5500 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
!
```

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
multicast-source ipv4
igmp snooping profile port_profile

!

```

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
  mtu 1514
  !
!
interface GigabitEthernet0/8/0/8.2 l2transport
  encapsulation dot1q 1002
  mtu 1514
  !
!

```

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
    multicast-source ipv4

```

```

        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. Configure two IGMP snooping profiles.

```

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

```

2. 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
!

```

3. Configure the bundle interfaces for L2 transport.

```

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

```

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

```

l2vpn
  bridge group bg1
    bridge-domain bd1
    multicast-source ipv4
    igmp snooping profile bridge_profile
    interface bundle-Ether 1
      multicast-source ipv4
      igmp snooping profile port_profile
    interface bundle-Ether 2
  !
!
!

```

5. 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. EVPN Configuration

```

hostname peer1
!
router bgp 100
  bgp router-id 1.1.1.1
  bgp graceful-restart
  address-family l2vpn evpn
  !
  neighbor 3.3.3.3
    remote-as 100
    update-source Loopback0
    address-family l2vpn evpn
  !
!
!
evpn
  evi 2
    advertise-mac
  !

```

```

!
interface Bundle-Ether2
  ethernet-segment
    identifier type 0 02.02.02.02.02.02.02.02
    bgp route-target 0002.0002.0002
!
!
!

```

3. 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. EVPN Configuration

```

hostname peer2
!
router bgp 100
  bgp router-id 2.2.2.2
  bgp graceful-restart
  address-family l2vpn evpn
  !
  neighbor 3.3.3.3
  remote-as 100
  update-source Loopback0
  address-family l2vpn evpn
  !
!

```

```

!
evpn
 evi 2
  advertise-mac
  !
!
interface Bundle-Ether2
 ethernet-segment
  identifier type 0 02.02.02.02.02.02.02.02
  bgp route-target 0002.0002.0002
  !
!
!

```

3. 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 and EVPN Sync

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 with the EVPN sync feature.

```

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

Additional References

Related Documents

Related Topic	Document Title
Configuring MPLS VPLS bridges	Implementing Virtual Private LAN Services on Cisco IOS XR Software module in the <i>MPLS Configuration Guide for Cisco NCS 5500 Series Routers</i>
Getting started information	
Configuring EFPs and EFP bundles	<i>Interface and Hardware Component Configuration Guide for Cisco NCS 5500 Series Routers</i>

Standards

Standards ¹	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

¹ Not all supported standards are listed.

MIBs

MIBs	MIBs Link
No MIBs support IGMP snooping.	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
RFC-4541	Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport



CHAPTER 4

Implementing MLD Snooping

This module describes how to implement MLD snooping on the Cisco NCS 5500 Series Routers.

Table 10: Feature History for MLD Snooping

Release	Modification
Release 6.6.25	This feature was introduced.

- [MLD Snooping](#) , on page 95
- [Creating a MLD Snooping Profile](#), on page 100
- [Deactivating MLD Snooping on a Bridge Domain](#), on page 101
- [Configuring Static Mrouter Ports \(MLD\)](#), on page 102
- [Configuring Router Guard \(MLD\)](#), on page 102
- [Configuring Immediate-leave for MLD](#), on page 103
- [Configuring Internal Querier for MLD](#), on page 104
- [Configuring Static Groups for MLD](#), on page 105
- [Configuring MLD Snooping](#), on page 107
- [Configuring MLD Snooping on Ethernet Bundles](#), on page 108

MLD Snooping

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 NCS 5500 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.

Restrictions for MLD Snooping

Following are the restrictions (features that are not supported):

- MLD Snooping is supported only on L2VPN bridge domains.
- Explicit host tracking.
- Multicast Admission Control.
- Security filtering.
- Report rate limiting.
- Multicast router discovery.
- IPv6 multicast is not supported for a multicast source that is behind the BVI interface. For example, the below configuration is not supported:

```
l2vpn
bridge group 1
  bridge-domain 1
    multicast-source ipv6
    mld snooping profile grp1
```

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
- RP Failover
- Stateful Switch-Over (SSO)
- 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 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.
- Pseudowires (PWs) in VPLS bridge domains.
- 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 NCS 5500 Series Routers randomly nominates a single port from the bundle to carry the multicast traffic.

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 mrouter and host ports.

Table 11: 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. <ul style="list-style-type: none"> • 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. <ul style="list-style-type: none"> • 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 12: 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	<ul style="list-style-type: none"> • 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. 	<ul style="list-style-type: none"> • 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 NCS 5500. 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.

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 `t1-check` and disable `router-alert-check` on the router for mld messages.
- Static mrouter 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
RP/0/RP0/CPU0:router(config)# mld snooping profile grp1
RP/0/RP0/CPU0:router(config-mld-snooping-profile)# system-ip-address fe80::1 link-local
RP/0/RP0/CPU0:router(config-mld-snooping-profile)# internal-querier
RP/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 */
RP/0/RP0/CPU0:router# configuration
RP/0/RP0/CPU0:router(config)# l2vpn
RP/0/RP0/CPU0:router(config-l2vpn)# bridge group GRP1
RP/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. */
RP/0/RP0/CPU0:router(config-l2vpn-bg-bd)# no mld snooping profile
RP/0/RP0/CPU0:router(config-l2vpn-bg-bd)# commit
```

Running Configuration

```
RP/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 Router Guard (MLD)

To prevent multicast routing protocol messages from being received on a port and, therefore, prevent a port from being a dynamic mrouter port, follow these steps. Note that both router guard and static mrouter commands may be configured on the same port.

Prerequisite

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



Note Router guard configuration is a port-level option and should be added to profiles intended for ports. It is not recommended to add router guard configuration to a profile intended for bridge domains. To do so would prevent all mrouter, including MLD queriers, from being discovered in the bridge domain.

Configuration

```
/* Enter the global configuration mode and create the Bridge Group GRP1 and the Bridge
Domain ISP1*/
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 host-port-profile

/* Configure router guard. This protects the port from dynamic discovery.*/
RP0/0/RP0/CPU0:router(config-mld-snooping-profile)# router-guard
RP0/0/RP0/CPU0:router(config-l2vpn-bg-bd)# commit
```

Running Configuration

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

Verification

Verify that the router guard 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:
```

Router Guard:	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-ml-d-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-ml-d-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-ml-d-snooping-profile)# internal-querier
RP0/0/RP0/CPU0:router(config-ml-d-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 #SGs	Profile	Act	Ver	#Ports	#Mrtrs	#Grps
-----	-----	---	---	-----	-----	-----

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

Port                               State
----                               -
                                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)# l2transport
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)# l2transport
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
RP0/0/RP1/CPU0:Router#

```