CONTENTS

PREFACE
Preface vii
  Changes to This Document vii
  Communications, Services, and Additional Information vii

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 5
    Verification 6
  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
Multicast Configuration Guide for Cisco NCS 5500 Series Routers, IOS XR Release 6.6.x

CHAPTER 3
Implementing Layer-2 Multicast with IGMP Snooping

Implementing Layer-2 Multicast with IGMP Snooping 45
Prerequisites for IGMP Snooping 45
Restrictions for IGMP Snooping 46
Information About IGMP Snooping 46
IGMP Snooping Overview 46
Description of Basic Functions 46
High Availability Features 47
Bridge Domain Support 47
Multicast Host Ports 47
Multicast Traffic Handling within a Bridge Domain with IGMP Snooping Enabled 47
Information About IGMP Snooping Configuration Profiles 48
Creating Profiles 49
Attaching and Detaching Profiles 49
Changing Profiles 49
Default IGMP Snooping Configuration Settings 50
IGMP Snooping Configuration at the Bridge Domain Level 50
IGMP Minimum Version 50
Group Membership Interval, Robustness Variable, and Query Interval 51
Multicast over Integrated Routing Bridging Active/Active Multihome 51
How to Configure IGMP Snooping 52
Creating an IGMP Snooping Profile 52
Where to Go Next 53
Attaching a Profile and Activating IGMP Snooping on a Bridge Domain 53
Detaching a Profile and Deactivating IGMP Snooping on a Bridge Domain 54
Attaching and Detaching Profiles to Ports Under a Bridge 56
Verifying Multicast Forwarding 57
Configuration Examples for IGMP Snooping 58
Configuring IGMP Snooping on Physical Interfaces Under a Bridge: Example 58
Configuring IGMP Snooping on VLAN Interfaces Under a Bridge: Example 59
Configuring IGMP Snooping on Ethernet Bundles Under a Bridge: Example 60
Implementing MLD Snooping 69

MLD Snooping 69
  Prerequisites for MLD Snooping 70
  Restrictions for MLD Snooping 70
  Advantages of MLD Snooping 70
  High Availability (HA) features for MLD 70
  Bridge Domain Support for MLD 71
  Multicast Router and Host Ports 71
  Multicast Router Discovery for MLD 71
  Multicast Traffic Handling for MLD 71
  Multicast Listener Discovery over BVI 73
    Configuration 73
  IPv6 Multicast Listener Discovery Snooping over BVI 75
    Configuring Internal Querier for MLD Snooping 76
  Creating a MLD Snooping Profile 76
  Deactivating MLD Snooping on a Bridge Domain 77
  Configuring Static Mrouter Ports (MLD) 78
  Configuring Router Guard (MLD) 78
  Configuring Immediate-leave for MLD 79
  Configuring Internal Querier for MLD 80
  Configuring Static Groups for MLD 81
  Configuring MLD Snooping 83
  Configuring MLD Snooping on Ethernet Bundles 84
Preface

This preface contains these sections:

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

Changes to This Document

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

Table 1: Changes to This Document

<table>
<thead>
<tr>
<th>Date</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>December 2018</td>
<td>Initial release of this document.</td>
</tr>
<tr>
<td>May 2019</td>
<td>Republished for Release 6.6.2</td>
</tr>
</tbody>
</table>

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.
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

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Changed in Release</th>
<th>Where Documented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multicast Route Statistics</td>
<td>Multicast route statistic feature provides information about the IPv4 multicast routes.</td>
<td>Release 6.6.25</td>
<td>Multicast Route Statistics, on page 33</td>
</tr>
<tr>
<td>IPv6 Multicast for Multiple Sources</td>
<td>From Release 6.6.25 onwards, IPv6 multicast supports multiple sources for a single multicast group.</td>
<td>Release 6.6.25</td>
<td>IPv6 Multicast for Multiple Sources, on page 30</td>
</tr>
<tr>
<td>EVPN: Dual PIM-DR (EVPN) – IPv4 Includes PIM-SM for Uplink – IPv4</td>
<td>This feature was introduced.</td>
<td>Release 6.6.25</td>
<td>Implementing Layer-2 Multicast with IGMP Snooping, on page 45</td>
</tr>
<tr>
<td>EVPN: IPv4 MC Enhanced DF election for MCAST</td>
<td>This feature was introduced.</td>
<td>Release 6.6.25</td>
<td>Implementing Layer-2 Multicast with IGMP Snooping, on page 45</td>
</tr>
<tr>
<td>IGMP (V2 only) Snooping on Layer 2 Transport – A/A MH (Receiver only) – PIM-ASM</td>
<td>This feature was introduced.</td>
<td>Release 6.6.25</td>
<td>Implementing Layer-2 Multicast with IGMP Snooping, on page 45</td>
</tr>
</tbody>
</table>
## Multicast Features Added or Modified in IOS XR Release 6.6.x

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Changed in Release</th>
<th>Where Documented</th>
</tr>
</thead>
<tbody>
<tr>
<td>IGMP Layer 2 EVPN State Synchronization</td>
<td>This feature was introduced.</td>
<td>Release 6.6.25</td>
<td>Implementing Layer-2 Multicast with IGMP Snooping, on page 45</td>
</tr>
</tbody>
</table>
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 33
- Use Case: Video Streaming, on page 34
- Multicast Label Distribution Protocol (MLDP) for Core, on page 35
- Configuration Example for MLDP on Core, on page 36
- Label Switched Multicast (LSM) Multicast Label Distribution Protocol (mLDP) based Multicast VPN (mVPN) Support, on page 36
- Configure VRF MLDP In-Band Signaling on Edge Routers, on page 40
- Configure Global MLDP In-band Signaling on Edge Routers, on page 41
- Configuration Examples for MVPN Profiles, on page 41
- Configuration Examples for Inband mLDP Profiles on Edge Routers, on page 42
- Verification of MLDP Configuration on Edge Routers, on page 43

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. The support will be available in future releases.

**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.

## Enabling Multicast

### Configuration Example

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

```sh
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

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

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

```

### Verification

Verify that the Interfaces are enabled for multicast.

```sh
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

```
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 routers 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
Verifying Multicast

Running Configuration

```
router pim
address-family ipv4
  interface HundredGige 0/1/0/1
    bfd minimum-interval 10
    bfd fast-detect
    bfd multiplier 3

  interface TengigabitEthernet 0/0/0/4
    bfd minimum-interval 50
    bfd fast-detect
    bfd multiplier 3

  interface TengigabitEthernet 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.
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-if)# static-rpf 10.0.0.1 32 TenGigE 0/0/0/1 192.168.0.2
Router(config-ipv4-acl)# 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]
    * 10.10.10.2/32 [0/0] 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 TenGigE0/0/0/0/2
```
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

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

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.
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 (mc1):

```bash
Router#configure
Router(config)#ipv4 access-list mc1
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):

```
```
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

```bash
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

```bash
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 Turnin-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 Turnin-56:

```bash
Router#show run int TenGigE0/1/0/6/3.132
interface TenGigE0/1/0/6/3.132
  description Connected to Turin-56 ten0/0/0/0/19.132
  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, RFFID - 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, 
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, RFFID - 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, 
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-default-ipv4-if)# dr-priority 4
Router(config-pim-default-ipv4-if)# hello-interval 45
Router(config-pim-default-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.

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
</tbody>
</table>
| **Step 2** | interface *type interface-path-id*  
*Example:*  
```
RP/0/RP0/CPU0:router(config)# interface loopback 0
```  
(Optional) Enters interface configuration mode to define the IPv4 address for the interface.  
**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. |
| **Step 3** | ipv4 address *address mask*  
*Example:*  
```
RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.0.1.3 255.255.255.0
```  
(Optional) Defines the IPv4 address for the interface.  
**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. |
| **Step 4** | exit  
*Example:*  
```
RP/0/RP0/CPU0:router(config-if)# end
```  
Exits interface configuration mode. |
| **Step 5** | router msdp  
*Example:*  
```
RP/0/RP0/CPU0:router(config)# router msdp
```  
Enters MSDP protocol configuration mode. |
| **Step 6** | default-peer *ip-address [prefix-list list]*  
*Example:*  
```
RP/0/RP0/CPU0:router(config-msdp)# default-peer 172.23.16.0
```  
(Optional) Defines a default peer from which to accept all MSDP SA messages. |
| **Step 7** | originator-id *type interface-path-id*  
*Example:*  
```
RP/0/RP0/CPU0:router(config-msdp)# originator-id /1/1/0
```  
(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. |
<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>peer peer-address</td>
<td>Enters MSDP peer configuration mode and configures an MSDP peer.</td>
</tr>
</tbody>
</table>
|      | Example: RP/0/RP0/CPU0:router(config-msdp)# peer 172.31.1.2 | • 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. |
| 9    | connect-source type interface-path-id | (Optional) Configures a source address used for an MSDP connection. |
|      | Example: RP/0/RP0/CPU0:router(config-msdp-peer)# connect-source loopback 0 |
| 10   | mesh-group name | (Optional) Configures an MSDP peer to be a member of a mesh group. |
|      | Example: RP/0/RP0/CPU0:router(config-msdp-peer)# mesh-group internal |
| 11   | remote-as as-number | (Optional) Configures the remote autonomous system number of this peer. |
|      | Example: RP/0/RP0/CPU0:router(config-msdp-peer)# remote-as 250 |
| 12   | commit | |
| 13   | show msdp [ipv4] globals | Displays the MSDP global variables. |
|      | Example: RP/0/RP0/CPU0:router# show msdp globals |
|      | Example: RP/0/RP0/CPU0:router# show msdp peer 172.31.1.2 |
| 15   | show msdp [ipv4] rpf rpf-address | Displays the RPF lookup. |
|      | Example: RP/0/RP0/CPU0:router# show msdp rpf 172.16.10.13 |
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**

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>configure</td>
<td></td>
</tr>
<tr>
<td></td>
<td>router msdp</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Enters MSDP protocol configuration mode.</td>
</tr>
</tbody>
</table>

Example:

RP/0/RP0/CPU0:router(config)# router msdp
### Purpose

**Command or Action**

| Step 3 | sa-filter \{in | out\} \{ip-address | peer-name\} [list access-list-name] [rp-list access-list-name] |
|--------|----------------------------------------------------------------------------------------------------------------------------------|
| Example: | RP/0/RP0/CPU0:router(config-msdp)# sa-filter out router.cisco.com list 100 |

**Purpose**

- Configures an incoming or outgoing filter list for messages received from the specified MSDP peer.
  - 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.

<table>
<thead>
<tr>
<th>Step 4</th>
<th>cache-sa-state [list access-list-name] [rp-list access-list-name]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-msdp)# cache-sa-state 100</td>
</tr>
</tbody>
</table>

**Purpose**

- Creates and caches source/group pairs from received Source-Active (SA) messages and controls pairs through access lists.

<table>
<thead>
<tr>
<th>Step 5</th>
<th>ttl-threshold ttl-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-msdp)# ttl-threshold 8</td>
</tr>
</tbody>
</table>

**Purpose**

- (Optional) Limits which multicast data is sent in SA messages to an MSDP peer.
  - Only multicast packets with an IP header TTL greater than or equal to the ttl-value 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.

<table>
<thead>
<tr>
<th>Step 6</th>
<th>exit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>RP/0/RP0/CPU0:router(config-msdp)# exit</td>
</tr>
</tbody>
</table>

**Purpose**

- Exits the current configuration mode.

<table>
<thead>
<tr>
<th>Step 7</th>
<th>ipv4 access-list name [sequence-number] permit source [source-wildcard]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td></td>
</tr>
</tbody>
</table>

**Purpose**

- Defines an IPv4 access list to be used by SA filtering.
  - In this example, the access list 100 permits multicast group 239.1.1.1.
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>RP/0/RP0/CPU0:router(config)# ipv4 access-list 100 20 permit 239.1.1.1 0.0.0.0</td>
<td>• The <code>ipv4 access-list</code> command is required if the keyword <code>list</code> is configured for SA filtering in Step 3, on page 20.</td>
</tr>
<tr>
<td>Step 8 commit</td>
<td></td>
</tr>
</tbody>
</table>

**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.
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.

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.

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.
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 IGMP Per Interface States 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

<table>
<thead>
<tr>
<th>Interface</th>
<th>Number</th>
<th>Max # Groups</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loopback0</td>
<td>4</td>
<td>4000</td>
</tr>
<tr>
<td>TenGigE0/0/0/0</td>
<td>5</td>
<td>4000</td>
</tr>
<tr>
<td>TenGigE0/0/0/1</td>
<td>5</td>
<td>4000</td>
</tr>
<tr>
<td>TenGigE0/0/0/2</td>
<td>0</td>
<td>4000</td>
</tr>
<tr>
<td>TenGigE0/0/0/3</td>
<td>5</td>
<td>4000</td>
</tr>
<tr>
<td>TenGigE0/0/0/6</td>
<td>5</td>
<td>3000</td>
</tr>
<tr>
<td>TenGigE0/0/0/18</td>
<td>5</td>
<td>4000</td>
</tr>
<tr>
<td>TenGigE0/0/0/19</td>
<td>5</td>
<td>4000</td>
</tr>
<tr>
<td>TenGigE0/0/0/6.1</td>
<td>3</td>
<td>4000</td>
</tr>
</tbody>
</table>
```

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(config)#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 - Externally 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, RPID - RPF ID Set, X - VXLAN
Interface flags: F - Forward, A - Accept, IC - Internal Copy,
   NI - Negate Interest, DI - Don't Preserve, SP - Signal Present,
   SI - 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
```
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 black-and-white 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

Before this release IPv6 multicast support was limited to a single source for each multicast group. However, when multiple sources were involved then it resulted in duplicating multicast flows of multiple sources to all interested receivers.

From Release 6.6.1 onwards, IPv6 multicast supports multiple sources for a single multicast group.

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.

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

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>show mfib hardware route statistics location &lt;node-id&gt;</td>
<td>Displays platform-specific MFIB information for the packet and byte counters for all multicast route.</td>
</tr>
<tr>
<td>show mfib hardware route &lt;source-address&gt; location &lt;node-id&gt;</td>
<td>Displays platform-specific MFIB information for the packet and byte counters for multicast routes originating from the specified source.</td>
</tr>
<tr>
<td>Show mfib hardware route &lt;source-address&gt;&lt;group-address&gt; location &lt;node-id&gt;</td>
<td>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.</td>
</tr>
<tr>
<td>clear mfib hardware ingress route statistics location &lt;node-id&gt;</td>
<td>Resets allocated counter values regardless of the MFIB hardware statistics mode from the designated node.</td>
</tr>
<tr>
<td>clear mfib hardware route &lt;source-address&gt; location &lt;node-id&gt;</td>
<td>Resets allocated counter values regardless of the MFIB hardware statistics mode from the designated node for specified multicast route source.</td>
</tr>
<tr>
<td>clear mfib hardware route &lt;source-address&gt;&lt;group-address&gt; location &lt;node-id&gt;</td>
<td>Resets allocated counter values regardless of the MFIB hardware statistics mode from the designated node for specified multicast route source and multicast group.</td>
</tr>
</tbody>
</table>

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` 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.
Multicast Route Statistics

Multicast route statistic feature provides information about the IPv4 multicast routes. The multicast statistics information includes:

- Number of packets received.
- Rate at which packets are received.

Before enabling multicast route statistics, you must configure an IPv4 ACL to specify which of the 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.

This feature supports only:

- L3 Multicast traffic.
- Default VRFs.

Note

You can configure an IPv4 ACL to capture up to 2000 (S,G) routes statistics.

Configure Multicast Route Statistics

Configuring multicast route statistics includes these main tasks:

- Configuring IPv4 ACL
- Enabling multicast route statistics for the configured ACLs

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

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

/* Enable multicast route statistics for the configured ACL. */
RP0/RP0/CPU0:router(config)# hw-module route-stats l3mcast ipv4 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:

```
RP0/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.

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 master 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 master 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.
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 8—Global P2MP-TE
- Profile 10—VRF Static-P2MP-TE with BGP AD
- Profile 12—Default MDT - MLDP - P2MP - BGP-AD - BGP C-mcast Signaling
- Profile 14—Partitioned MDT - MLDP P2MP - BGP-AD - BGP C-mcast Signaling
Profile 17—Default MDT - MLDP - P2MP - BGP-AD - PIM C-mcast Signaling

Configuration Example for MLDP on Core

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

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

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, 7, 8 and 10.

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
Implementing Multicast

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).

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.

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
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
```
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
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 TenGigabitEthernet0/0/0/21
      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
mpls 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
```
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:
0x0001 P2MP 5.5.5.5 [vpnv6 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 : 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 12:08:39.472 MLDP GLO 0/RP0/CPU0 t6746 GEN : EDM Loc init iox success
May 31 12:08:39.472 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 12:08:39.475 MLDP GLO 0/RP0/CPU0 t6746 GEN : RUMP init iox success
May 31 12:08:39.475 MLDP GLO 0/RP0/CPU0 t6746 GEN : Chunks init iox success
May 31 12:08:39.509 MLDP ERR 0/RP0/CPU0 t6746 GEN : RIB not ready
May 31 12:08:39.509 MLDP ERR 0/RP0/CPU0 t6746 GEN : RIB not ready
May 31 12:08:39.512 MLDP GLO 0/RP0/CPU0 t6746 GEN : mldp_ens_event_ctx_chunk 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_main_evm 0x563de8fd23e8 allocated
May 31 12:08:39.512 MLDP GLO 0/RP0/CPU0 t6746 GEN : RIB Thread EVM init iox success
May 31 12:08:39.512 MLDP GLO 0/RP0/CPU0 t6746 GEN : RIB Thread Queue init iox success
May 31 12:08:39.512 MLDP GLO 0/RP0/CPU0 t6746 GEN : Bound to RIB, fd: 354
```
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 4: Feature History for IGMP Snooping

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.6.25</td>
<td>This feature was introduced.</td>
</tr>
</tbody>
</table>

- Implementing Layer-2 Multicast with IGMP Snooping, on page 45
- Prerequisites for IGMP Snooping, on page 45
- Restrictions for IGMP Snooping, on page 46
- Information About IGMP Snooping, on page 46
- Multicast over Integrated Routing Bridging Active/Active Multihome, on page 51
- How to Configure IGMP Snooping, on page 52
- Configuration Examples for IGMP Snooping, on page 58
- Additional References, on page 66

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

The following prerequisites must be satisfied before implementing IGMP snooping:

- The network must be configured with a Layer 2 VPN (L2VPN).
- 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 IGMP Snooping

- IGMP snooping is supported only under L2VPN bridge domains.
- If a L2VPN port is enabled for IGMP snooping, a BVI must be present in the bridge domain.
- 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

- Explicit host tracking (an IGMPv3 snooping feature) is not supported.
- IGMPv1 is not supported.

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.
- 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 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 5: Multicast Traffic Handling for an IGMPv2 Querier, on page 47 describes traffic handling for an IGMPv2 querier. Table 6: Multicast Traffic Handling for an IGMPv3 Querier, on page 48 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 5: Multicast Traffic Handling for an IGMPv2 Querier

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

Table 6: Multicast Traffic Handling for an IGMPv3 Querier

<table>
<thead>
<tr>
<th>Traffic Type</th>
<th>Received on Host Ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP multicast source traffic</td>
<td>Forwards to all mrouter ports and to host ports that indicate interest.</td>
</tr>
<tr>
<td>IGMP general queries</td>
<td>—</td>
</tr>
<tr>
<td>IGMP group-specific queries</td>
<td>—</td>
</tr>
<tr>
<td>IGMPv2 joins</td>
<td>Handles as IGMPv3 IS_EX{} reports.</td>
</tr>
<tr>
<td>IGMPv3 reports</td>
<td>• If proxy reporting is enabled—For state changes or source-list changes, generates a</td>
</tr>
<tr>
<td></td>
<td>state change report on all mrouter ports.</td>
</tr>
<tr>
<td></td>
<td>• If proxy reporting is disabled—Forwards on all mrouter ports.</td>
</tr>
<tr>
<td>IGMPv2 leaves</td>
<td>Handles as IGMPv3 IS_IN{} reports.</td>
</tr>
</tbody>
</table>

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 50.

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.
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 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.
Default IGMP Snooping Configuration Settings

### Table 7: IGMP Snooping Default Configuration Values

<table>
<thead>
<tr>
<th>Scope</th>
<th>Feature</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bridge Domain</td>
<td>IGMP snooping</td>
<td>Disabled on a bridge domain until an enabling IGMP profile is attached to the bridge domain.</td>
</tr>
<tr>
<td></td>
<td>internal querier</td>
<td>None configured</td>
</tr>
<tr>
<td></td>
<td>last-member-query-count</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>last-member-query-interval</td>
<td>1000 (milliseconds)</td>
</tr>
<tr>
<td></td>
<td>minimum-version</td>
<td>2 (supporting IGMPv2 and IGMPv3)</td>
</tr>
<tr>
<td></td>
<td>querier query-interval</td>
<td>60 (seconds)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note</strong> This is a nonstandard default value.</td>
</tr>
<tr>
<td></td>
<td>report-suppression</td>
<td>Enabled (enables report suppression for IGMPv2 and proxy-reporting for IGMPv3)</td>
</tr>
<tr>
<td></td>
<td>querier robustness-variable</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>router alert check</td>
<td>Enabled</td>
</tr>
<tr>
<td></td>
<td>tcn query solicit</td>
<td>Disabled</td>
</tr>
<tr>
<td></td>
<td>tcn flood</td>
<td>Enabled</td>
</tr>
<tr>
<td></td>
<td>ttl-check</td>
<td>Enabled</td>
</tr>
<tr>
<td></td>
<td>unsolicited-report-timer</td>
<td>1000 (milliseconds)</td>
</tr>
<tr>
<td>Port</td>
<td>immediate-leave</td>
<td>Disabled</td>
</tr>
<tr>
<td></td>
<td>mrouter</td>
<td>No static mrouters configured; dynamic discovery occurs by default.</td>
</tr>
<tr>
<td></td>
<td>router guard</td>
<td>Disabled</td>
</tr>
<tr>
<td></td>
<td>static group</td>
<td>None configured</td>
</tr>
</tbody>
</table>

**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:

\[
GMI = (\text{robustness-variable} \times \text{query-interval}) + \text{maximum-response-time}
\]

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.

Multicast over Integrated Routing Bridging Active/Active Multihome

Multicast over integrated routing bridging active/active multihome feature enables the routers to quickly and safely switch traffic between routers, during failure, without any traffic loss. This feature comprises of the following four sub features that work together as a solution:

• First, IGMPv2 snooping is enabled for the peer routers to know which Layer 2 interface has receiver interested in a particular group.

• After snooping, this information is synced to the peer routers with the Layer 2 EVPN sync feature.

• After both peer routers are synced, they act like a last hop router and send PIM join upstream.
Once the traffic arrives on both the peer routers, only one peer router forwards the traffic to the receiver with the designated forwarder election feature.

# 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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters IGMP snooping profile configuration mode and creates a named profile.</td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>igmp snooping profile profile-name</code> <strong>Example:</strong> <code>RP/0/RP0/CPU0:router(config)# igmp snooping profile default-bd-profile</code></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Step 3</strong> Optionally, add commands to override default configuration values.</td>
<td>If you are creating a bridge domain profile, consider the following:</td>
</tr>
<tr>
<td></td>
<td>• An empty profile is appropriate for attaching to a bridge domain. An empty profile enables IGMP snooping with default configuration values.</td>
</tr>
<tr>
<td></td>
<td>• You can optionally add more commands to the profile to override default configuration values.</td>
</tr>
<tr>
<td></td>
<td>• 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.</td>
</tr>
<tr>
<td></td>
<td>If you are creating a port-specific profile, consider the following:</td>
</tr>
</tbody>
</table>
• While an empty profile could be attached to a port, it would have no effect on the port configuration.

• When you attach a profile to a port, IGMP snooping reconfigures that port, overriding any inheritance of configuration values from the bridge-domain profile. You must repeat the commands in the port profile if you want to retain those configurations.

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

---

**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. igmp snooping profile *profile-name*
6. commit
7. show igmp snooping bridge-domain detail
8. show l2vpn bridge-domain detail

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> l2vpn</td>
<td>Enters Layer 2 VPN configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RP0/CPU0:router(config)# l2vpn</td>
</tr>
<tr>
<td><strong>Step 3</strong> bridge group <em>bridge-group-name</em></td>
<td>Enters Layer 2 VPN bridge group configuration mode for the named bridge group.</td>
</tr>
</tbody>
</table>
## 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
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
<td></td>
</tr>
</tbody>
</table>
| **Step 2** | `l2vpn`  
**Example:** RP/0/RP0/CPU0:router(config)# l2vpn | Enters Layer 2 VPN configuration mode. |
| **Step 3** | `bridge group bridge-group-name`  
**Example:** RP/0/RP0/CPU0:router(config-l2vpn)# bridge group GRP1 | Enters Layer 2 VPN bridge group configuration mode for the named bridge group. |
| **Step 4** | `bridge-domain bridge-domain-name`  
**Example:** RP/0/RP0/CPU0:router(config-l2vpn-bg)# bridge-domain ISP1 | Enters Layer 2 VPN bridge group bridge domain configuration mode for the named bridge domain. |
| **Step 5** | `no igmp snooping`  
**Example:** RP/0/RP0/CPU0:router(config-l2vpn-bg-bd)# no igmp snooping | Detaches the IGMP snooping profile from the bridge domain, disabling IGMP snooping on that bridge domain.  
**Note** Only one profile can be attached to a bridge domain at a time. If a profile is attached, IGMP snooping is enabled. If a profile is not attached, IGMP snooping is disabled. |
| **Step 6** | commit |  |
| **Step 7** | `show igmp snooping bridge-domain detail`  
**Example:** RP/0/RP0/CPU0:router# show igmp snooping bridge-domain detail | (Optional) Verifies that IGMP snooping is disabled on a bridge domain. |
| **Step 8** | `show l2vpn bridge-domain detail`  
**Example:** RP/0/RP0/CPU0:router# show l2vpn bridge-domain | (Optional) Verifies that IGMP snooping is disabled in the forwarding plane (Layer 2) on a bridge domain. |

6. commit  
7. `show igmp snooping bridge-domain detail`  
8. `show l2vpn bridge-domain detail`
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.

Bridge virtual interface (BVI) must be defined and attached to the bridge-domain for IGMP snooping to work.

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. Do one of the following:
   • igmp snooping profile profile-name
   • no igmp snooping
7. routed interface BVI bvi number
8. commit
9. show igmp snooping bridge-domain detail
10. show l2vpn bridge-domain detail

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><strong>configure</strong></td>
</tr>
<tr>
<td>Step 2</td>
<td><strong>l2vpn</strong></td>
</tr>
<tr>
<td>Example:</td>
<td><strong>Enters Layer 2 VPN configuration mode.</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td><strong>RP/0/RP0/CPU0:router(config)# l2vpn</strong></td>
</tr>
<tr>
<td>Step 3</td>
<td><strong>bridge group bridge-group-name</strong></td>
</tr>
<tr>
<td>Example:</td>
<td><strong>Enters Layer 2 VPN bridge group configuration mode for the named bridge group.</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td><strong>RP/0/RP0/CPU0:router(config-l2vpn)# bridge group GRP1</strong></td>
</tr>
<tr>
<td>Step 4</td>
<td><strong>bridge-domain bridge-domain-name</strong></td>
</tr>
<tr>
<td>Example:</td>
<td><strong>Enters Layer 2 VPN bridge group bridge domain configuration mode for the named bridge domain.</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td><strong>RP/0/RP0/CPU0:router(config-l2vpn-bg)# bridge-domain ISP1</strong></td>
</tr>
</tbody>
</table>
### Command or Action

**Step 5**

```bash
interface interface-type interface-number
```

**Example:**

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

**Purpose**

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

**Step 6**

Do one of the following:

- `igmp snooping profile profile-name`
- `no igmp snooping`

**Example:**

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

**Purpose**

Attaches the named IGMP snooping profile to the port.

**Note**

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

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

**Step 7**

`routed interface BVI bvi number`

**Example:**

```bash
RP/0/RP0/CPU0:router(config-l2vpn-bg-bd-if)# routed interface bvi 2
```

**Purpose**

Attaches the BVI to the bridge domain.

*BVI number* can be any number.

**Step 8**

`commit`

**Step 9**

`show igmp snooping bridge-domain detail`

**Example:**

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

**Purpose**

(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:**

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

**Purpose**

(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 [detail] [hardware { ingress | egress }] location node-id`
3. `show l2vpn forwarding bridge-domain [bridge-group-name:bridge-domain-name] mroute ipv4 summary location node-id`

---

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** configure | 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. Example:

```
RP/0/RP0/CPU0:router# show l2vpn forwarding bridge-domain bridgeGroup1:ABC mroute ipv4 detail location 0/3/CPU0
```

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

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

```
RP/0/RP0/CPU0:router# show l2vpn forwarding bridge-domain bridgeGroup1:ABC mroute ipv4 summary location 0/3/CPU0
```

### 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 and a BVI.

   ```
   interface GigabitEthernet0/8/0/38
   negotiation auto
   l2transport
   no shut
   
   interface GigabitEthernet0/8/0/39
   negotiation auto
   l2transport
   ```
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. Also add a BVI under bridge domain.

```
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
  routed interface BVI2
```

4. Verify the configured bridge ports.

```
show igmp snooping port
```

### Configuring IGMP Snooping on VLAN Interfaces Under a Bridge: Example

1. Configure two profiles.

```
igmp snooping profile bridge_profile
igmp snooping profile port_profile
```

2. Configure VLAN interfaces for L2 transport and a BVI.

```
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
    
    interface BVI2
    ipv4 address 100.2.0.1 255.255.255.0
```
Mac-address 1002.1111.2

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. Also add a BVI under bridge domain.

l2vpn
bridge group bg1
bridge-domain bd1
igmp snooping profile bridge_profile
interface GigabitEthernet0/8/0/8.1
igmp snooping profile port_profile
interface GigabitEthernet0/8/0/8.2
routed interface BVI2

4. Verify the configured bridge ports.

show igmp snooping port

Configuring IGMP Snooping on Ethernet Bundles Under a Bridge: Example

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

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

2. Configure two IGMP snooping profiles.

   igmp snooping profile bridge_profile
   !
   igmp snooping profile port_profile
   !

3. Configure interfaces as bundle member links.

   interface GigabitEthernet0/0/0/0

Multicast Configuration Guide for Cisco NCS 5500 Series Routers, IOS XR Release 6.6.x
4. Configure the bundle interfaces for L2 transport and a BVI.

```plaintext
interface Bundle-Ether 1
   l2transport
!
interface Bundle-Ether 2
   l2transport
!
interface BVI2
   ipv4 address 100.2.0.1 255.255.255.0
   mac-address 1002.1111.2
```

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

```plaintext
l2vpn
   bridge group bg1
   bridge-domain bd1
   igmp snooping profile bridge_profile
   interface bundle-Ether 1
      igmp snooping profile port_profile
   interface bundle-Ether 2
   routed interface BVI2
!
```

6. Verify the configured bridge ports.

```
show igmp snooping port
```

---

**Configuring Multicast over Integrated Routing Bridging Active/Active Multihome**

**Configurations performed on peer 1:**
1. Layer 2 Base Configuration

hostname peer1

interface Bundle-Ether2

interface Bundle-Ether2.2 l2transport
  encapsulation dot1q 2
  rewrite ingress tag pop 1 symmetric

interface TenGigE0/0/0/0
  bundle id 2 mode on
  no shut

interface BVI2
  ipv4 address 100.2.0.1 255.255.255.0
  mac-address 1002.1111.2

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

interface BVI2
  version 2

l2vpn
  bridge group VLAN2
  bridge-domain VLAN2
    igmp snooping profile 1
    interface Bundle-Ether2.2

  routed interface BVI2

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
! interface BVI2
   ipv4 address 100.2.0.1 255.255.255.0
   mac-address 1002.1111.2
!

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
   interface BVI2
      version 2
!
!
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.

Verifying Dual DR PIM Uplink

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

As described in the previous example, both peer1 and peer2 have BV12 as outgoing interface. However, only one of the peer should forward the traffic. Designated forwarder election elects one of them to do the forwarding. In this example, peer2 is selected as the forwarder. Peer1 has Bundle-Ether2.2 marked as NDF.

RP/0/RP0/CPU0:peer1#show l2vpn forwarding bridge-domain VLAN2:VLAN2 mroute ipv4 hardware ingress detail location 0/0/cPU0
Bridge-Domain: VLAN2:VLAN2, ID: 0

Prefix: (0.0.0.0,239.0.0.2/32)
P2MP enabled: N
IRB platform data: {0x0, 0x2d, 0x0, 0x0}, len: 32
Bridge Port:
EVPN, Xconnect id: 0x80000015 NH:2.2.2.2
Bundle-Ether2.2, Xconnect id: 0xa0000015 (NDF)

RP/0/RP0/CPU0:peer2#show l2vpn forwarding bridge-domain VLAN2:VLAN2 mroute ipv4 hardware ingress detail location 0/0/cPU0

Verifying Designated Forwarder Election

As described in the previous example, both peer1 and peer2 have BV12 as outgoing interface. However, only one of the peer should forward the traffic. Designated forwarder election elects one of them to do the forwarding. In this example, peer2 is selected as the forwarder. Peer1 has Bundle-Ether2.2 marked as NDF.
Additional References

**Related Documents**

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

**Standards**

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.</td>
</tr>
</tbody>
</table>

1 Not all supported standards are listed.

**MIBs**

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>No MIBs support IGMP snooping.</td>
<td>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: <a href="http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml">http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml</a></td>
</tr>
</tbody>
</table>

**RFCs**

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC-4541</td>
<td>Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches</td>
</tr>
</tbody>
</table>
## Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>searchable technical content, including links to products, technologies,</td>
<td></td>
</tr>
<tr>
<td>solutions, technical tips, and tools. Registered Cisco.com users can log</td>
<td></td>
</tr>
<tr>
<td>in from this page to access even more content.</td>
<td></td>
</tr>
</tbody>
</table>
Implementing MLD Snooping

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

Table 8: Feature History for MLD Snooping

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 6.6.25</td>
<td>This feature was introduced.</td>
</tr>
</tbody>
</table>

- MLD Snooping, on page 69
- Creating a MLD Snooping Profile, on page 76
- Deactivating MLD Snooping on a Bridge Domain, on page 77
- Configuring Static Mrouter Ports (MLD), on page 78
- Configuring Router Guard (MLD), on page 78
- Configuring Immediate-leave for MLD, on page 79
- Configuring Internal Querier for MLD, on page 80
- Configuring Static Groups for MLD, on page 81
- Configuring MLD Snooping, on page 83
- Configuring MLD Snooping on Ethernet Bundles, on page 84

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:

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

Advantages of MLD Snooping

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 mrouters and host ports.
### Table 9: Multicast Traffic Handling for a MLDv1 Querier

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

### Table 10: Multicast Traffic Handling for a MLDv2 Querier

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

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 through BVI interface from L3 segment to the L2 segment of the network.

- 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. MLDv2 join messages coming in or out of this broadcast domain passes through the BVI interface.

**Configuration**

This sample configuration shows how to configure BVI interface to join a multicast group and statically forward multicast traffic using MLDv2:

```
router(config)# config
router(config)# router mld
router(config-mld)# vrf BVI
router(config-mld-vrf)# interface BVI100
router(config-mld-vrf-int)# join-group fe32::1 192::4
router(config-mld-vrf-int)# static-group fe32::2 192::4
router(config-mld-vrf-int)# commit
router(config-mld-vrf-int)# exit
router(config-mld-vrf)# exit
router(config-mld)# exit
router(config)# exit
```

**Verification**

Use the command `show mld bvi stats` and `show mld group bvi < num>` to verify the MLDv2 over BVI configuration:

```
router# show mld bvi stats
Thu Nov 22 13:58:34 474 UTC
AIPC buffers received: 8365
AIPC buffer released: 8365
AIPC messages send blocked: 0
AIPC buffer release failed: 0
```
AIPC NULL buffer handles : 0
AIPC open notifications received : 0
AIPC close notifications received : 0
AIPC error notifications received : 0
AIPC LWM notifications received : 0
AIPC input waiting notifications received : 8308
AIPC send status notifications received : 2485
AIPC publish notifications received : 0
AIPC queue full notifications received : 0
AIPC output notifications received : 0
AIPC connect notifications received : 1
IGMP protocol messages received : 8365
IGMP Mrouter Add messages received : 0
IGMP Mrouter Delete messages received : 0
IGMP Mrouter Sweep messages received : 1
IGMP Mrouter Add messages transmitted : 13
IGMP Mrouter Delete messages transmitted : 22
IGMP Mrouter Sweep messages transmitted : 0
IGMP Mrouter Unknown messages received : 0
IGMP Mrouter Unknown messages transmitted : 0
AIPC transmission error : 0
AIPC buffers transmitted : 0
IGMP protocol buffers transmitted : 2482
IGMP Mrouter buffers transmitted : 3
IGMP Unknown buffers transmitted : 0
IGMP WTX Msqs received : 0
IGMP WTX Msqs sent : 0
IGMP WTX Msqs sent to protocol : 0
IGMP WTX Msqs dropped due DC : 99264
IGMP WTX Msqs dropped no memory : 0
IGMP WTX Msqs freed : 0

router# show mld group bvi 100
Thu Nov 22 13:58:52.055 UTC
MLD Connected Group Membership

BVI100
Group Address : ff02::2
Last Reporter : fe80::1a33:9dff:fe3d:b73f
  Uptime : 03:31:07
  Expires : never
Group Address : ff02::d
Last Reporter : fe80::1a33:9dff:fe3d:b73f
  Uptime : 03:31:07
show mld group bvi 100
Thu Nov 22 13:58:52.055 UTC
MLD Connected Group Membership

BVI100
Group Address : ff02::2
Last Reporter : fe80::1a33:9dff:fe3d:b73f
  Uptime : 03:31:07
  Expires : never
Group Address : ff02::d
Last Reporter : fe80::1a33:9dff:fe3d:b73f
  Uptime : 03:31:07
  Expires : never
Group Address : ff02::1:ff01:1
IPv6 Multicast Listener Discovery Snooping over BVI

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

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

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

Restrictions

• You cannot configure `ttl-check` and disable `router-alert-check` on the router for mld messages.

• Static 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
RP0/RP0/CPU0:router(config)# mld snooping profile grp1
RP0/RP0/CPU0:router(config-mld-snooping-profile)# system-ip-address fe80::1 link-local
RP0/RP0/CPU0:router(config-mld-snooping-profile)# internal-querier
RP0/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**

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

Verification

Verify that the MLD snooping profile is created:

```bash
RP0/RP0/CPU0:router# show mld snooping profile
```

<table>
<thead>
<tr>
<th>Profile</th>
<th>Bridge Domain</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>default-bd-profile</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>grp1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>grp10</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

**Deactivating MLD Snooping on a Bridge Domain**

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

**Configuration**

```bash
/* Enter the global configuration mode followed by the bridge group and the bridge domain mode */
RP0/RP0/CPU0:router# configuration
RP0/RP0/CPU0:router(config)# l2vpn
RP0/RP0/CPU0:router(config-12vpn)# bridge group GRP1
RP0/RP0/CPU0:router(config-12vpn-bg)# bridge-domain ISP1

/* Detach the MLD snooping profile from the bridge domain. This disables MLD snooping on that bridge domain */
/* Note: Only one profile can be attached to a bridge domain at a time. If a profile is attached, MLD snooping is enabled. If a profile is not attached, MLD snooping is disabled. */
RP0/RP0/CPU0:router(config-12vpn-bg-bd)# no mld snooping profile
RP0/RP0/CPU0:router(config-12vpn-bg-bd)# commit

Running Configuration

```bash
RP0/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 Snooping Profile mrouter-port-profile:

<table>
<thead>
<tr>
<th>Static Mrouter:</th>
<th>Enabled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bridge Domain References:</td>
<td>0</td>
</tr>
<tr>
<td>Port References:</td>
<td>0</td>
</tr>
</tbody>
</table>

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 mrouters, 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-12vpn-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 Snooping Profile host-port-profile:

<table>
<thead>
<tr>
<th>Router Guard:</th>
<th>Enabled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bridge Domain References:</td>
<td>0</td>
</tr>
<tr>
<td>Port References:</td>
<td>0</td>
</tr>
</tbody>
</table>

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. */
Configuring Internal Querier for MLD

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

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

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

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

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

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:

<table>
<thead>
<tr>
<th>Immediate Leave:</th>
<th>Enabled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router Guard:</td>
<td>Enabled</td>
</tr>
<tr>
<td>Bridge Domain References:</td>
<td>0</td>
</tr>
<tr>
<td>Port References:</td>
<td>0</td>
</tr>
</tbody>
</table>

Implementing MLD Snooping
Configuring Internal Querier for MLD
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

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

<table>
<thead>
<tr>
<th>Bridge Domain</th>
<th>Profile</th>
<th>Act</th>
<th>Ver</th>
<th>#Ports</th>
<th>#Mtrs</th>
<th>#Grps</th>
</tr>
</thead>
<tbody>
<tr>
<td>f1:100</td>
<td>grp1</td>
<td>y</td>
<td>v2</td>
<td>3</td>
<td>1</td>
<td>1000</td>
</tr>
</tbody>
</table>

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

Multicast Configuration Guide for Cisco NCS 5500 Series Routers, IOS XR Release 6.6.x
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)# 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-mld-snooping)# 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

<table>
<thead>
<tr>
<th>Port</th>
<th>State</th>
<th>Oper</th>
<th>STP</th>
<th>Red</th>
<th>#Grps</th>
<th>#SGs</th>
</tr>
</thead>
<tbody>
<tr>
<td>BVI1009</td>
<td>Up</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>GigabitEthernet0/8/0/38</td>
<td>Up</td>
<td>-</td>
<td>-</td>
<td>1000</td>
<td>1000</td>
<td></td>
</tr>
<tr>
<td>GigabitEthernet0/8/0/39</td>
<td>Up</td>
<td>-</td>
<td>-</td>
<td>1000</td>
<td>1000</td>
<td></td>
</tr>
</tbody>
</table>

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
RP0/0/RP0/CPU0:router(config-if)# exit
RP0/0/RP0/CPU0:router(config)# interface GigabitEthernet0/0/1
RP0/0/RP0/CPU0:router(config-if)# exit
RP0/0/RP0/CPU0:router(config)# interface GigabitEthernet0/0/2
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 GigabitEthernet0/0/3
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
/* Configure interfaces as bundle member links. */

RP0/0/RP0/CPU0:router(config)# interface GigabitEthernet0/0/0
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)# mld snooping profile bridge_profile
RP0/0/RP0/CPU0:router(config-l2vpn)# mld snooping profile port_profile
RP0/0/RP0/CPU0:router(config-l2vpn)# interface bundle-Ether 1
RP0/0/RP0/CPU0:router(config-l2vpn)# commit
RP0/0/RP0/CPU0:router(config-l2vpn)# interface bundle-Ether 2
RP0/0/RP0/CPU0:router(config-l2vpn)# 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
    mld snooping profile port_profile
      mrouter
  interface GigabitEthernet0/0/0/0/0
  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

!
12vpn
  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/RP0/CPU0:router# show mld snooping port