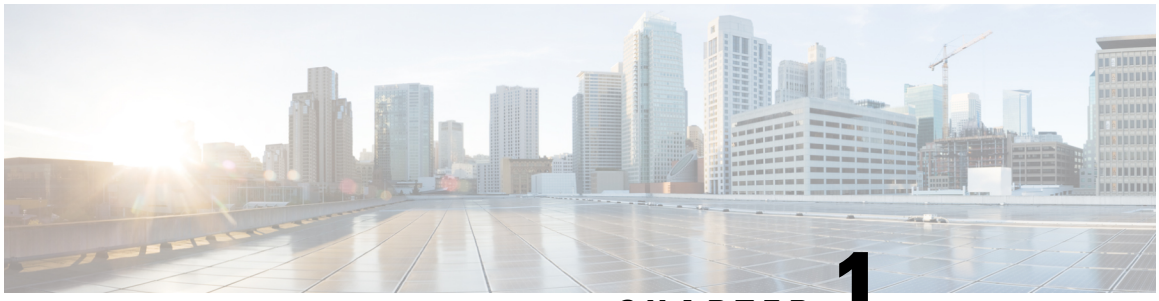
Multicast Configuration Guide for Cisco NCS 540 Series Routers, Cisco IOS XR Release 25.1.x

First Published: 2025-01-30

Last Modified: 2025-03-31

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

CONTENTS



CHAPTER 1

YANG Data Models for Multicast Features

This chapter provides information about the YANG data models for Multicast features.

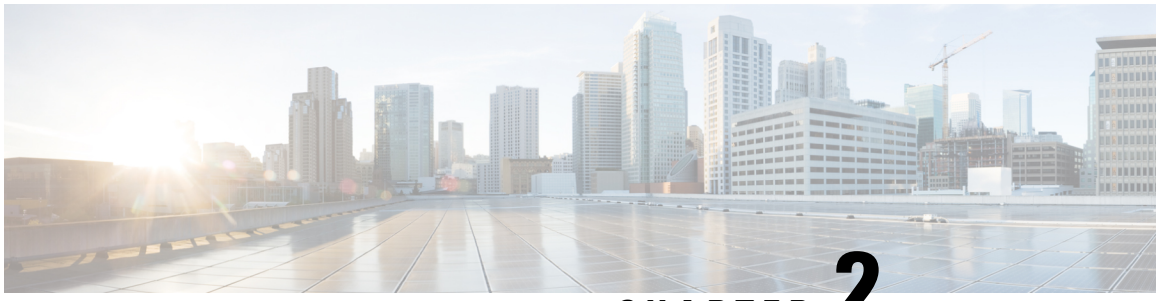
Using YANG Data Models

Cisco IOS XR supports a programmatic way of configuring and collecting operational data of a network device using YANG data models. Although configurations using CLIs are easier and human-readable, automating the configuration using model-driven programmability results in scalability.

The data models are available in the release image, and are also published in the Github repository. Navigate to the release folder of interest to view the list of supported data models and their definitions. Each data model defines a complete and cohesive model, or augments an existing data model with additional XPaths. To view a comprehensive list of the data models supported in a release, navigate to the **Available-Content.md** file in the repository.

You can also view the data model definitions using the YANG Data Models Navigator tool. This GUI-based and easy-to-use tool helps you explore the nuances of the data model and view the dependencies between various containers in the model. You can view the list of models supported across Cisco IOS XR releases and platforms, locate a specific model, view the containers and their respective lists, leaves, and leaf lists presented visually in a tree structure. This visual tree form helps you get insights into nodes that can help you automate your network.

To get started with using the data models, see the *Programmability Configuration Guide*.



CHAPTER 2

Implementing Layer 3 Multicast

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.

Prerequisites for Implementing Multicast Routing

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

Restrictions

Enabling Multicast

Configuration Example

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

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

Running Configuration

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

Verification

Verify that the Interfaces are enabled for multicast.

```
Router#show mfib interface location 0/3/CPU00/RP0/cpu0
Interface : FINT0/RP0/cpu0 (Enabled)
SW Mcast pkts in : 0, SW Mcast pkts out : 0
TTL Threshold : 0
Ref Count : 2
Interface : TenGigE0/0/0/3 (Enabled)
SW Mcast pkts in : 0, SW Mcast pkts out : 0
TTL Threshold : 0
Ref Count : 3
Interface : TenGigE0/0/0/9 (Enabled)
SW Mcast pkts in : 0, SW Mcast pkts out : 0
TTL Threshold : 0
Ref Count : 13
Interface : Bundle-Ether1 (Enabled)
SW Mcast pkts in : 0, SW Mcast pkts out : 0
TTL Threshold : 0
Ref Count : 4
Interface : Bundle-Ether1.1 (Enabled)
SW Mcast pkts in : 0, SW Mcast pkts out : 0
TTL Threshold : 0
```

Supported Multicast Features

- Hardware Offloaded BFD for PIMv4 is supported.
- IPv4 and IPV6 static groups for both IGMPv2/v3 and MLDv1/v2 are supported.
- Protocol Independent Multicast in Source-Specific Multicast (PIM-SSM) mapping is supported.
- PIMv4 SSM and PIMv6 SSM over Bundle sub-interface are supported.
- MFIB stats (source, group) or (S,G) are supported.
- Load balancing for multicast traffic for ECMP links and bundles is supported. However, an RPF check is performed on the multicast routers to ensure loop prevention in multicast trees.
- IPv6 multicast MLD joins are subjected to hop by hop LPTS punt policer. Tweaking this policer to a higher value achieves convergence at higher scale.

  Also, adjust the ICMP control traffic LPTS hardware policer to a higher value for optimal convergence at higher scale.
- Redundant sources for IPv6 PIM SSM is supported.

Note You must enable **multicast-routing** in the default VRF to use multicast in some other VRF.

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.

PIM on Bundle-Ethernet subinterface is supported.

PIM BFD Overview

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

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

Protocol Independent Multicast (PIM) uses a hello mechanism for discovering new PIM neighbors between adjacent nodes. The minimum failure detection time in PIM is 3 times the PIM Query-Interval. To enable faster failure detection, the rate at which a PIM hello message is transmitted on an interface is configurable. However, lower intervals increase the load on the protocol and can increase CPU and memory utilization and cause a system-wide negative impact on performance. Lower intervals can also cause PIM neighbors to expire frequently as the neighbor expiry can occur before the hello messages received from those neighbors are processed. When PIM BFD is enabled, BFD enables faster failure detection without waiting for hello messages from PIM.

Configure PIM BFD

This section describes how you can configure PIM BFD

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

```
Router(config-pim-ipv4-if)#  bfd fast-detect
Router(config-pim-ipv4-if)#  bfd multiplier 3
Router(config-pim-ipv4-if)#  exit
Router(config-pim-default-ipv4)# interface  Bundle-Ether 101
Router(config-pim-ipv4-if)#  bfd minimum-interval 50
Router(config-pim-ipv4-if)#  bfd fast-detect
Router(config-pim-ipv4-if)#  bfd multiplier 3
Router(config-pim-ipv4-if)#  exit
Router(config-pim-default-ipv4)#  commit
```

Running Configuration

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

  !
 !
!
```

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
```

PIM Bootstrap Router

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

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

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

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

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

Configuring PIM Bootstrap Router

Configuration Example

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

```
Router#config
Router(config)#router pim
Router(config-pim-default-ipv4)#bsr candidate-bsr 1.1.1.1 hash-mask-len 30 priority 1
Router(config-pim-default-ipv4-if)#commit
```

Configures the router to advertise itself as a candidate rendezvous point to the BSR in its PIM domain. Access list number 4 specifies the prefix associated with the candidate rendezvous point address 1.1.1.1 . This rendezvous point is responsible for the groups with the prefix 239.

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

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

Running Configuration

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

Verification

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

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

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

Reverse Path Forwarding

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

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

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

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

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

Setting the Reverse Path Forwarding Statically

Configuration Example

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

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

Running Configuration

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

Verification

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

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

RPF Vector Encoding Using IETF Standard

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

Configuring RPF Vector (IETF Standard Encoding)

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

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

Verification

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

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

Neighbor Address             Interface              Uptime    Expires  DR pri   Flags

25.25.25.1                   TenGigE0/0/0/0       1w3d      00:01:36 1       B P
25.25.25.2*                  TenGigE0/0/0/0       1w3d      00:01:41 1 (DR) B P E
32.32.32.2*                  TenGigE0/0/0/2
       1w4d      00:01:40 1      B P E
32.32.32.3                   TenGigE0/0/0/2
       1w4d      00:01:42 1 (DR) B P
```

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

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

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

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

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

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

Benefits of PIM-SSM over PIM-SM

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

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):

```
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):

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

Running Configuration

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

Multipath Option

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

Configuring Multipath Option

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

Running Configuration

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

Verification

The Bundle-Ether132 and TenGigE0/4/0/18/0.132 are two paths to reach the destination router 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:

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

Router#show run int be132
interface Bundle-Ether132
 description Bundle between Fretta-56 and Turin-56
 ipv4 address 28.0.0.1 255.255.255.240
 ipv6 address 2606::28:0:0:1/120
 load-interval 30

Router#show mrib route 50.11.30.11 detail

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

(50.11.30.11,225.255.11.1) Ver: 0x523cc294 RPF nbr: 50.11.30.11 Flags: L RPF, FGID: 11453,
 -1, -1
  Up: 4d15h
  Incoming Interface List
    HundredGigE0/4/0/10.1130 Flags: A, Up: 4d15h
  Outgoing Interface List
    FortyGigE0/1/0/5 Flags: F NS, Up: 4d15h
    TenGigE0/4/0/6/0 Flags: F NS, Up: 4d15h
    TenGigE0/1/0/6/3.132 Flags: F NS, Up: 4d15h
    TenGigE0/4/0/18/0.122 Flags: F NS, Up: 4d15h
```

```
Router#show mrib route 50.11.30.12 detail

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

(50.11.30.12,226.255.12.1) Ver: 0x5fe02e5b RPF nbr: 50.11.30.12 Flags: L RPF, FGID: 12686,
 -1, -1
  Up: 4d15h
  Incoming Interface List
    HundredGigE0/4/0/10.1130 Flags: A, Up: 4d15h
  Outgoing Interface List
    Bundle-Ether121 Flags: F NS, Up: 4d15h
    Bundle-Ether132 Flags: F NS, Up: 4d15h
    FortyGigE0/1/0/5 Flags: F NS, Up: 4d15h
    TenGigE0/4/0/6/0.117 Flags: F NS, Up: 4d15h
```

Configuring PIM-SSM

Configuration Example

Configures SSM default range 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
```

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

Running Configuration

```
Router#show running multicast-routing
multicast-routing
 address-family ipv4
  ssm range 4
```

```
  interface all enable
 !

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

Verification

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

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

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

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

Configuring PIM Parameters

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

Configuration Example

```
Router#configure
Router(config)#router pim
Router(config-pim-default)#address-family ipv4
Router(config-pim-default-ipv4)#dr-priority 2
Router(config-pim-default-ipv4)#spt-threshold infinity
Router(config-pim-default-ipv4)#interface TenGigE0/0/0/1
Router(config-pim-ipv4-if)#dr-priority 4
Router(config-pim-ipv4-if)#hello-interval 45
Router(config-pim-ipv4-if)#commit
```

Running Configuration

```
Router#show run router pim
router pim
 address-family ipv4
```

```
  dr-priority 2
  spt-threshold infinity
  interface TenGigE0/0/0/1
   dr-priority 4
   hello-interval 45
```

Verification

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

```
Router#show pim interface te0/0/0/1
PIM interfaces in VRF default
Address               Interface                PIM  Nbr   Hello  DR    DR Count Intvl
  Prior
100.1.1.1             TenGigE0/0/0/1           on   1     45     4     this system
```

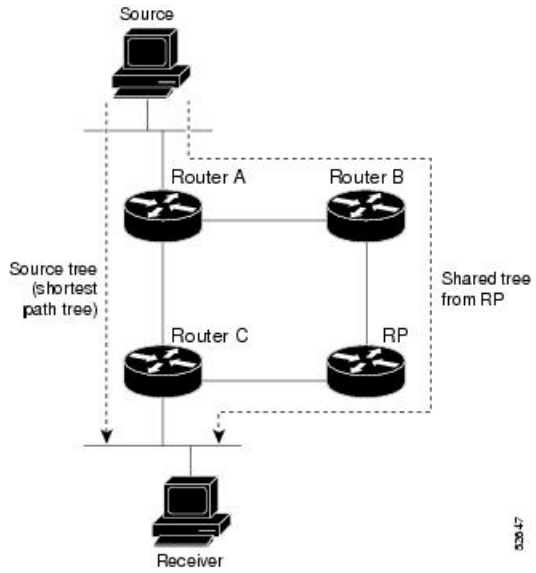
PIM-Sparse Mode

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).

Restrictions and Usage Guidelines

This section describes the restrictions and guidelines related to the PIM protocol.

The following restrictions and guidelines apply for the Protocol Independent Multicast-Sparse Mode (PIM-SM) protocol:

- IPv6 is not supported.
- Auto RP is not supported.
- BSR is not supported in profile 14.

Multicast Source Discovery Protocol

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

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

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

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

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.

Procedure

	Command or Action	Purpose
Step 1	**configure**	
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:**	(Optional) Defines a default peer from which to accept all MSDP SA messages.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-msdp)# default-peer 172.23.16.0`	
Step 7	**originator-id** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config-msdp)# originator-id GigabitEthernet0/1/1/0`	(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.
Step 8	**peer** *peer-address* **Example:** `RP/0/RP0/CPU0:router(config-msdp)# peer 172.31.1.2`	Enters MSDP peer configuration mode and configures an MSDP peer. • Configure the router as a BGP neighbor. • If you are also BGP peering with this MSDP peer, use the same IP address for MSDP and BGP. You are not required to run BGP or multiprotocol BGP with the MSDP peer, as long as there is a BGP or multiprotocol BGP path between the MSDP peers.
Step 9	**connect-source** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config-msdp-peer)# connect-source loopback 0`	(Optional) Configures a source address used for an MSDP connection.
Step 10	**mesh-group** *name* **Example:** `RP/0/RP0/CPU0:router(config-msdp-peer)# mesh-group internal`	(Optional) Configures an MSDP peer to be a member of a mesh group.
Step 11	**remote-as** *as-number* **Example:** `RP/0/RP0/CPU0:router(config-msdp-peer)# remote-as 250`	(Optional) Configures the remote autonomous system number of this peer.
Step 12	**commit**	
Step 13	**show msdp** [**ipv4**] **globals** **Example:** `RP/0/RP0/CPU0:router# show msdp globals`	Displays the MSDP global variables.

	Command or Action	Purpose
Step 14	**show msdp** [**ipv4**] **peer** [*peer-address*] **Example:** `RP/0/RP0/CPU0:router# show msdp peer 172.31.1.2`	Displays information about the MSDP peer.
Step 15	**show msdp** [**ipv4**] **rpf** *rpf-address* **Example:** `RP/0/RP0/CPU0:router# show msdp rpf 172.16.10.13`	Displays the RPF lookup.

Controlling Source Information on MSDP Peer Routers

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

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

When forwarding SA messages you can do the following:

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

When receiving SA messages you can do the following:

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

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

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

Procedure

	Command or Action	Purpose
Step 1	**configure**	

	Command or Action	Purpose
Step 2	**router msdp** **Example:** `RP/0/RP0/CPU0:router(config)# router msdp`	Enters MSDP protocol configuration mode.
Step 3	**sa-filter** {**in** | **out**} [**list** *access-list-name*] [**rp-list** *access-list-name*] **Example:** `RP/0/RP0/CPU0:router(config-msdp)# sa-filter out list 100`	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 23. • If all match criteria are true, a **permit** from the route map passes routes through the filter. A **deny** filters routes.
Step 4	**cache-sa-state** [**list** *access-list-name*] [**rp-list** *access-list-name*] **Example:** `RP/0/RP0/CPU0:router(config-msdp)# cache-sa-state list 100`	Creates and caches source/group pairs from received Source-Active (SA) messages and controls pairs through access lists.
Step 5	**ttl-threshold** *ttl-value* **Example:** `RP/0/RP0/CPU0:router(config-msdp)# ttl-threshold 8`	(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.

	Command or Action	Purpose
Step 6	**exit** **Example:** `RP/0/RP0/CPU0:router(config-msdp)# exit`	Exits the current configuration mode.
Step 7	**ipv4 access-list** *name* [*sequence-number*] **permit** *source* [*source-wildcard*] **Example:** `RP/0/RP0/CPU0:router(config)# ipv4 access-list 100 20 permit 239.1.1.1 0.0.0.0`	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. • The **ipv4 access-list** command is required if the keyword **list** is configured for SA filtering in Step 3, on page 22.
Step 8	**commit**	

Designated Routers

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

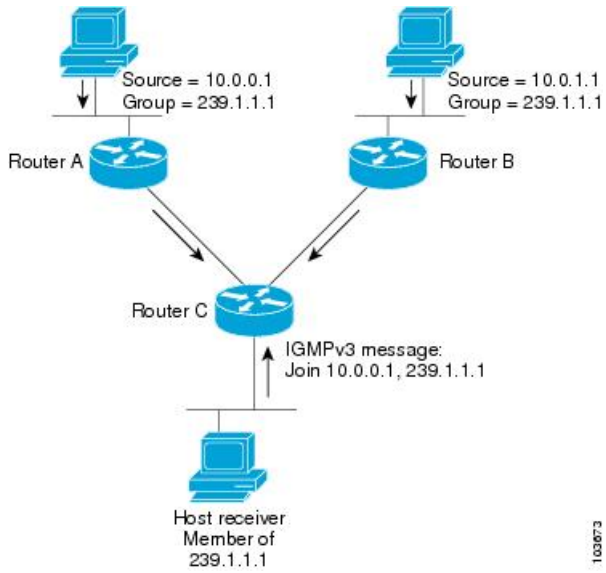
The designated router is responsible for sending PIM register and PIM join and prune messages toward the RP to inform it about host group membership.

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

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

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

Figure 2: Designated Router Election on a Multiaccess Segment



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

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

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

Configuration Example

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

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

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

Running Configuration

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

Verification

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

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

Designated Router Election Using StickyDR

Table 1: Feature History Table

Feature Name	Release Information	Feature Description
Designated Router Election Using StickyDR	Release 7.4.1	With this feature, the router sends a PIM *hello* message with a special PIM DR priority value on a multi-access LAN. The router with this special DR priority value is always elected as the designated router. The traffic now flows in the same path even when a new router is added. This feature introduces the **sticky-dr** command.

When you enable PIM on an interface or reload a router, router periodically sends the PIM Hello messages on each interface. PIM Hello messages allow a router to learn neighboring PIM routers on each interface and elects a Designated Router (DR) based on the DR Priority. The DR election avoids duplicating multicast traffic for connected hosts.

Each time the DR is reelected, the multicast control tree sets up a new path and the multicast traffic flows in different direction.

With Sticky DR feature, the designated router remains the same and doesn't allow any other router to become the designated router. The multicast control tree does not set up a new path and the multicast traffic flows in same direction, thus avoids traffic loss. DR election isn't based on DR priority.

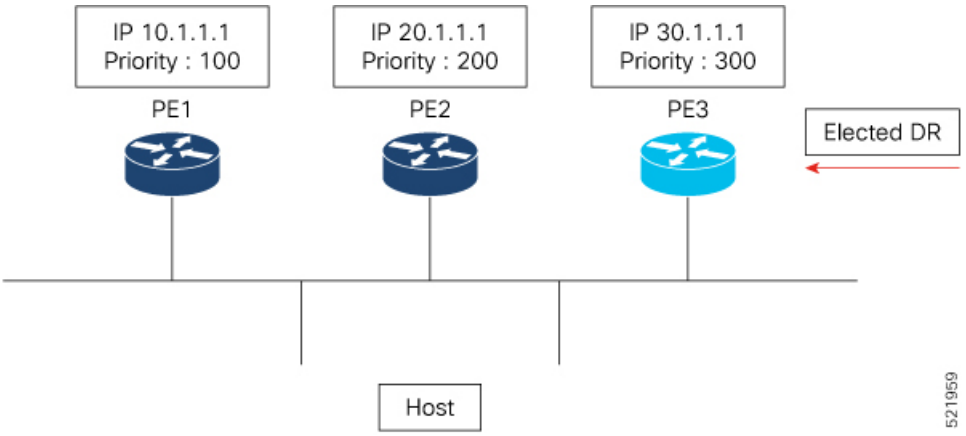
After you enable the sticky DR feature, the elected DR no longer advertises configured DR. Instead the router sends PIM Hello message with special PIM DR priority value which is reserved for Sticky PIM DR.

Restrictions

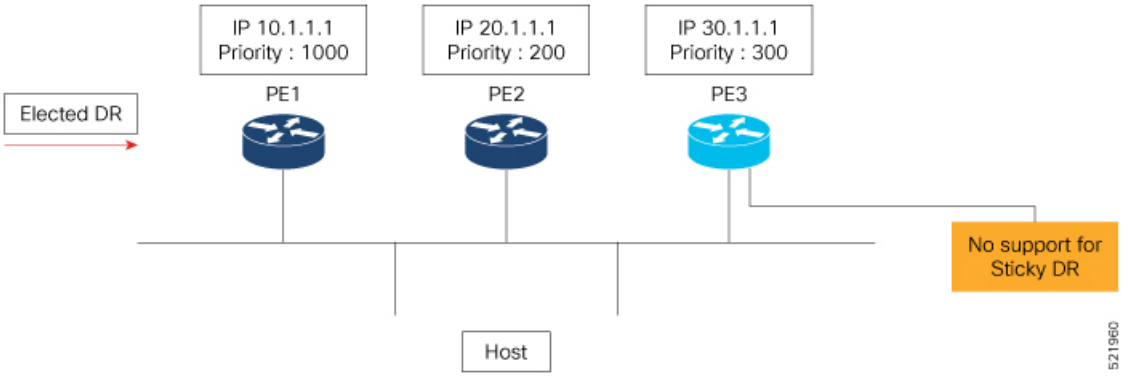
- The Sticky DR priority value is 4294967294. You must not configure DR priority with the value 4294967294 or any number greater than this value.

Topology

In this topology, PE1, PE2, PE3 are three PIM routers connected on a LAN. PE3 has the maximum priority and hence PE3 is elected as DR.

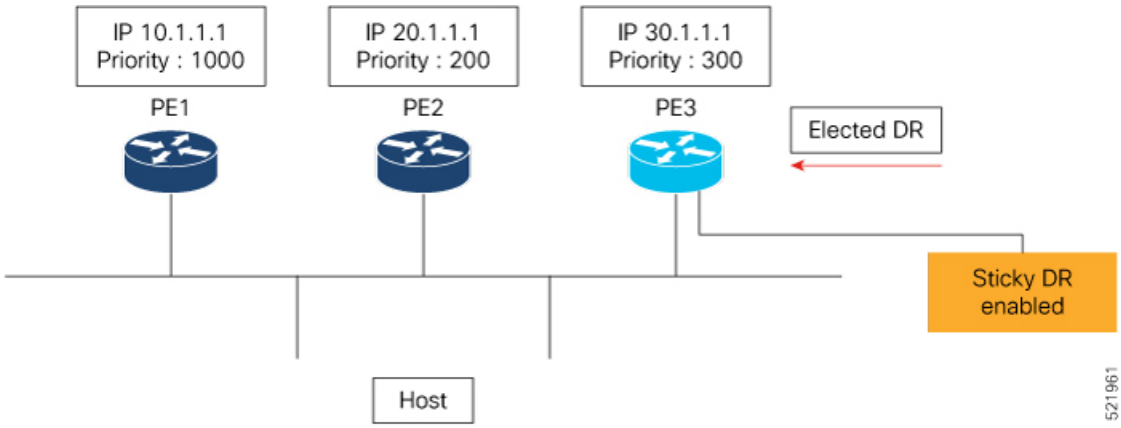


Now, when you configure PE1 with DR priority 1000, DR election process is re-initiated and PE1 becomes the new DR.



Every time a new DR is elected, the control tree computes a new path for traffic flow.

Now if you enable sticky DR on PE3, the PE3 remains the designated router irrespective of the DR priority of the PE devices.



In this example, the sticky DR is configured on PE3 and PE3 always remains as the DR.

Configuration

Let's configure sticky DR on PE3. To configure sticky DR on an interface, perform the following task:

```
Router# configure
Router(config)# router pim
Router(config-pim-default)# address-family ipv4
Router(config-pim-default-ipv4)# interface bundle-ether 72.1
Router(config-pim-ipv4-if)# sticky-dr
Router(config-ipv4-acl)# commit
```

Verification

The following output specifies that the Sticky DR is enabled on the interface and active:

```
Router# show pim interface bundle-ether 72.1 detail

PIM interfaces in VRF default
IP PIM Multicast Interface State
Flag: B - Bidir enabled, NB - Bidir disabled
      P - PIM Proxy enabled, NP - PIM Proxy disabled
      V - Virtual Interface, S - Sticky DR enabled
BFD State - State/Interval/Multiplier

Interface                  PIM  Nbr    Hello  DR
                                Count  Intvl  Prior

Bundle-Ether72.1            on   2     30     100000
    Primary Address : 200.1.72.1
              Flags : B NP S V
                BFD : On/150 ms/3
                 DR : this system
  Propagation delay : 500
  Override Interval : 2500
       Hello Timer : 00:00:24
    Neighbor Filter : -
          Sticky DR : Configured, Active since Mon Jul 26 16:53:01 2021

-----------------------
Sticky DR Event History
-----------------------
Event                     State        Time
```

```
-----                     -----        ----
Dynamic Batch             Active       (null)
```

The following output specifies that the Sticky DR is enabled on the interface and is inactive:

```
Router#  show pim interface bundle-ether 72.1 detail

PIM interfaces in VRF default
IP PIM Multicast Interface State
Flag: B - Bidir enabled, NB - Bidir disabled
      P - PIM Proxy enabled, NP - PIM Proxy disabled
      V - Virtual Interface, S - Sticky DR enabled
BFD State - State/Interval/Multiplier

Interface                  PIM  Nbr   Hello  DR
                                Count Intvl  Prior

Bundle-Ether72.1            on   2     30     1
    Primary Address : 200.1.72.1
              Flags : B NP S V
                BFD : On/150 ms/3
                 DR : 200.1.72.2
  Propagation delay : 500
  Override Interval : 2500
       Hello Timer : 00:00:18
    Neighbor Filter : -
          Sticky DR : Configured, Inactive
```

```
Router# show pim neighor detail

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

Neighbor Address             Interface              Uptime    Expires  DR pri      Flags

201.7.7.7*                   tunnel-mte1019         2d17h     00:01:36 1           (DR) B
E
  Expiry Timer: 00:01:05
201.7.7.7*                   tunnel-mte1001         2d17h     00:01:36 1           (DR) B
E
  Expiry Timer: 00:01:12
200.1.71.1*                  Bundle-Ether71.1       2d17h     00:01:31 99          (DR) B
  Expiry Timer: 00:00:02
200.1.71.2                   Bundle-Ether71.1       2d17h     00:01:19 1           B
BFD State: enabled
201.7.7.7*                   Loopback0              2d17h     00:01:41 1           (DR) B
E
  Expiry Timer: 00:01:12
201.202.7.7*                 Loopback1              2d17h     00:01:40 1           (DR) B
E
  Expiry Timer: 00:01:11
200.1.72.1*                  Bundle-Ether72.1       2d17h     00:01:15 -           (DR) B
S
  Expiry Timer: 00:01:21
```

Disable Sticky DR

To disable the sticky DR feature, perform the following task:

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

```
Router(config-pim-default-ipv4)# interface bundle-ether 72.1
Router(config-pim-ipv4-if)# no sticky-dr
Router(config-ipv4-acl)# commit
```

To clear the DR stickiness and force the DR reelection, use the following command:

```
Router# clear pim interface bundle-ether 72.1 sticky-dr
```

Multicast VPN

Multicast VPN (MVPN) provides the ability to dynamically provide multicast support over MPLS networks. MVPN introduces an additional set of protocols and procedures that help enable a provider to support multicast traffic in a VPN.

Note PIM-Bidir is not supported on MVPN.

There are two ways MCAST VPN traffic can be transported over the core network:

- Rosen GRE (native): MVPN uses GRE with unique multicast distribution tree (MDT) forwarding to enable scalability of native IP Multicast in the core network. MVPN introduces multicast routing information to the VPN routing and forwarding table (VRF), creating a Multicast VRF. In Rosen GRE, the MCAST customer packets (c-packets) are encapsulated into the provider MCAST packets (p-packets), so that the PIM protocol is enabled in the provider core, and mrib/mfib is used for forwarding p-packets in the core.
- MLDP ones (Rosen, partition): MVPN allows a service provider to configure and support multicast traffic in an MPLS VPN environment. This type supports routing and forwarding of multicast packets for each individual VPN routing and forwarding (VRF) instance, and it also provides a mechanism to transport VPN multicast packets across the service provider backbone. In the MLDP case, the regular label switch path forwarding is used, so core does not need to run PIM protocol. In this scenario, the c-packets are encapsulated in the MPLS labels and forwarding is based on the MPLS Label Switched Paths (LSPs) ,similar to the unicast case.

In both the above types, the MVPN service allows you to build a Protocol Independent Multicast (PIM) domain that has sources and receivers located in different sites.

To provide Layer 3 multicast services to customers with multiple distributed sites, service providers look for a secure and scalable mechanism to transmit customer multicast traffic across the provider network. Multicast VPN (MVPN) provides such services over a shared service provider backbone, using native multicast technology similar to BGP/MPLS VPN.

MVPN emulates MPLS VPN technology in its adoption of the multicast domain (MD) concept, in which provider edge (PE) routers establish virtual PIM neighbor connections with other PE routers that are connected to the same customer VPN. These PE routers thereby form a secure, virtual multicast domain over the provider network. Multicast traffic is then transmitted across the core network from one site to another, as if the traffic were going through a dedicated provider network.

Multi-instance BGP is supported on multicast and MVPN. Multicast-related SAFIs can be configured on multiple BGP instances.

Multicast VPN Routing and Forwarding

Dedicated multicast routing and forwarding tables are created for each VPN to separate traffic in one VPN from traffic in another.

The VPN-specific multicast routing and forwarding database is referred to as **MVRF**. On a PE router, an MVRF is created when multicast is enabled for a VRF. Protocol Independent Multicast (PIM), and Internet Group Management Protocol (IGMP) protocols run in the context of MVRF, and all routes created by an MVRF protocol instance are associated with the corresponding MVRF. In addition to VRFs, which hold VPN-specific protocol states, a PE router always has a global VRF instance, containing all routing and forwarding information for the provider network.

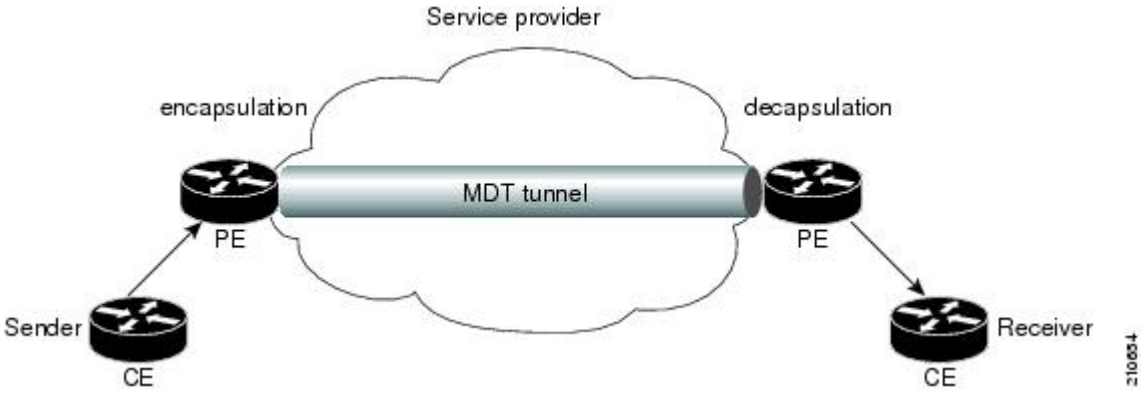
Multicast Distribution Tree Tunnels

The multicast distribution tree (MDT) can span multiple customer sites through provider networks, allowing traffic to flow from one source to multiple receivers. For MLDP, the MDT tunnel are called Labeled MDT (LMDT).

Secure data transmission of multicast packets sent from the customer edge (CE) router at the ingress PE router is achieved by encapsulating the packets in a provider header and transmitting the packets across the core. At the egress PE router, the encapsulated packets are decapsulated and then sent to the CE receiving routers.

Multicast distribution tree (MDT) tunnels are point-to-multipoint. A MDT tunnel interface is an interface that MVRF uses to access the multicast domain. It can be deemed as a passage that connects an MVRF and the global MVRF. Packets sent to an MDT tunnel interface are received by multiple receiving routers. Packets sent to an MDT tunnel interface are encapsulated, and packets received from a MDT tunnel interface are decapsulated.

Figure 3: Virtual PIM Peer Connection over an MDT Tunnel Interface



Encapsulating multicast packets in a provider header allows PE routers to be kept unaware of the packets' origin—all VPN packets passing through the provider network are viewed as native multicast packets and are routed based on the routing information in the core network. To support MVPN, PE routers only need to support native multicast routing.

MVPN also supports optimized VPN traffic forwarding for high-bandwidth applications that have sparsely distributed receivers. A dedicated multicast group can be used to encapsulate packets from a specific source, and an optimized MDT can be created to send traffic only to PE routers connected to interested receivers. This is referred to **data MDT**.

Naming Data MDTs

Table 2: Feature History Table

Feature Name	Release Information	Feature Description
Naming Data MDTs	Release 7.5.2	You can decide on the multicast flow that must be mapped to a specific MDT by naming it and assigning the flows to the named data MDT.

This feature enables you to deterministically map flows to data MDT at ingress PE by assigning the flows to a named data MDT.

You can define and create policies to map flows, which must flow to the required PEs, to specific data MDTs instead of dynamically multiplexing the flow to existing data MDT. This mapping makes the distribution of flows to data MDT more deterministic and solves the inherent inefficiencies.

Transitioning a flow to a data MDT occurs in one of the following conditions:

- The rate of the flow reaches the configured threshold limit.

  You can set the threshold limit of data MDTs by using the following command:

  ```
  mdt data <max nr of data groups> (threshold)
  ```

- Immediate switch is configured in which case the flow when detected is transitioned to a data MDT.

  For more details of immediate switch option configuration, see mdt data ingress replication command.

The multicast flow is transitioned into named data MDT if the following conditions are met:

- The criteria for a flow to be part of the data MDT is met.
- The flow satisfies the rules that are associated with the route-policy.

Named Data MDT

Based on the configured threshold, when the multicast flow is chosen to be transitioned to data MDT, the flow is compared against the configured route-policy. If it satisfies the specified conditions, the Cisco IOS XR software creates a named data MDT and transitions the flow to the new named data MDT.

The transport-specific parameters that are required to build the distribution tree based on the underlay transport (like FEC for MLDP) are autogenerated. Subsequent flows that match the policy and are mapped to the same named data MDT, are multiplexed to the same multicast transport tree in the core.

Even if you are using only named data MDTs, and the number of named data MDTs that can be configured is independent of the number of data MDTs configured, you have to configure a nonzero data MDT value for creating named data MDTs.

Flow-Mapping Rules for Named Data MDT

When you use route-policy to create named data MDTs, the following rules may be applicable:

- You can map disparate multicast flows to the same named data-mdt.

- You can specify multiple named data MDTs in one policy.
- Only the flows that are mapped using the route-policy use a named data MDT.

  Flows which do not match the route-policy but are eligible to be transitioned to data-mdt, are transitioned to regular data-mdt.
- You can map a named-data MDT to only one path (Refer Chapter: Flexible Algorithm for MLDP).

  You can have a rule for creating a new named-data-mdt, that uses the same flexible algorithm which is used by another rule.
- You can map the named data MDT to only one color (configured using the **set on-demand-color <val>** command), when Tree-SID is used as core.
- Named data MDTs with the same name in different VRFs create different named data MDTs.
- If you use the same route-policies across VRFs, it creates different data MDTs.
- Named data MDTs with same name in IPv4 and IPv6 AF of a VRF creates different named data MDTs.
- Flows that do not match the route policy are mapped to dynamic MDTs.

Note Any change to the route-policy, deletes the existing named-data-mdt and recreates it.

Restrictions and Limitation of Named Data MDT

- The total number of named data MDTs and the number of dynamic data MDTs across all VRFs and address families (AF) must not exceed the router-supported limit for a given transport.
- The number of flows that can be transitioned to a single named data MDT is restricted to 255 by default. You can change this default 255 value using the **mdt data max-aggregation <value>** command. However, the total number of named data MDTs that can be created is not limited by the total number of regular data MDTs allowed by the configuration.

  When you update the maximum aggregation (**max-aggregation**) value, it does not re-evaluate the existing flows. However, the updated value is applicable to the new flows.

Configuring Named Data MDT

You can apply the route-policy for named data MDT using the following sample configurations and verify using the show commands:

```
Router(config)#multicast-routing vrf red address-family ipv4
Router(config-mcast-red-ipv4) #mdt data mldp 10 route-policy RedGroup-1
Router(config-mcast-red-ipv4) #
Router(config-mcast-red-ipv4) #exit

Router(config) #route-policy RedGroup-1
Router(config-rpl) #if destination in (228.0.0.0/24 le 32) then
Router(config-rpl-if) #set data-mdt RedGroup-1
Router(config-rpl-if) #set flex-algo 128
Router(config-rpl-if) #pass
Router(config-rpl-if) #endif
Router(config-rpl) #end-policy
```

Verifying Named Data MDT

The following command outputs show the `Name` column which has the data MDT name.

```
Router#show pim vrt vonl mdt cache
Fri Aug 6 09:39:17.210 PDT

Core Source  Cust (Source, Group)     Core Data        Expires  Name
192.0.2.4  (31.3.233.7, 232.0.0.1)  [tree-id 524296]   never    RedGroup-1

  Leaf AD: 192.0.2.2
           192.0.2.1

192.0.2.4  (31.3.233.7, 232.0.0.2)  [tree-id 524296]    never    RedGroup-2

Leaf AD: 192.0.2.1


Router#show pim vrf vpn1 mdt sr-p2mp local
Fri Aug 6 09:39:20.435 PDT
Tree                        MDT          Cache  DIP Local  VRF Routes  On-demand Name
Identifier                  Source       Count      Entry  Using Cache Color
[tree-id 524296 (0x80008)]  192.0.2.4  2      N   Y      2           0           RedGroup-1

   Tree.SID Leaf:  192.0.2.1
                   192.0.2.2
Router#
```

```
Router#sh pim vrf vpn2 mdt cache
Tue Aug 17 04:39:58.751 PDT
Core Source   Cust (Source, Group)     Core Data      Expires Name
192.0.2.4   (31.3.234.7, 232.0.0.1)  [global-id 7]  00:02:14 n-mdt-vrf-vpn2
192.0.2.4   (31.3.234.7, 232.0.0.2)  [global-id 7]  00:02:14 n-mdt-vrf-vpn2
192.0.2.4   (31.3.234.7, 232.0.0.3)  [global-id 7]  00:02:14 n-mdt-vrf-vpn2
Router#
Router#

Router#sh pim vrf von2 mdt mldp local
Tue Aug 17 04:40:19.160 PDT
Core       MDT     Cache Max DIP Local VRF Routes    Name
Identifier Source  Count Agg     Entry Using Cache
[global-id 71      3     255 N   Y     3             n-mdt-vrf-von2
Router#
```

Internet Group Management Protocol

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

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

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

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

Restrictions

IGMP snooping under VPLS bridge domain is not supported.

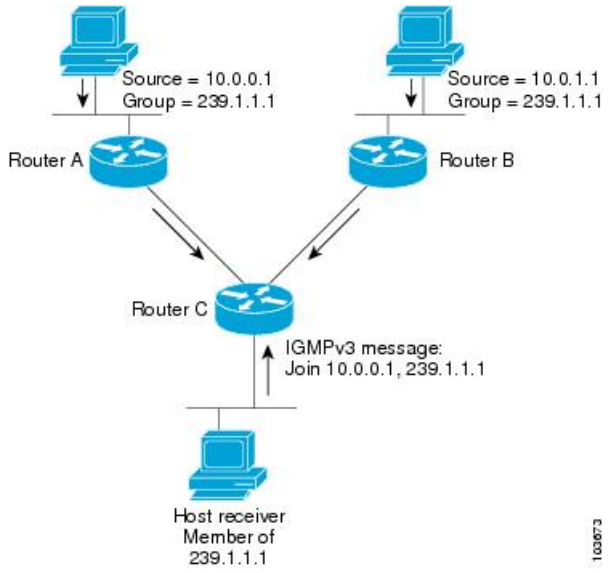
Functioning of IGMP Routing

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 4: IGMP Signaling



Configuring Maximum IGMP Per Interface Group Limit

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

- If a user has configured a maximum of 20 groups and has reached the maximum number of groups, then no more groups can be created. If the user reduces the maximum number of groups to 10, the 20 joins

will remain and a message of reaching the maximum is displayed. No more joins can be added until the number of groups has reached less than 10.

- If a user already has configured a maximum of 30 joins and add a max of 20, the configuration occurs displaying a message that the maximum has been reached. No state change occurs and also no more joins can occur until the threshold number of groups is brought down below the maximum number of groups.

Configuration Example

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

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

Running Configuration

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

Verification

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

SSM Static Source Mapping

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

Configuration Example

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

Running Configuration

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

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

Verification

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

```
Router#show mrib route 229.1.1.1 detail
IP Multicast Routing Information Base
Entry flags: L - Domain-Local Source, E - External Source to the Domain,
    C - Directly-Connected Check, S - Signal, IA - Inherit Accept,
    IF - Inherit From, D - Drop, ME - MDT Encap, EID - Encap ID,
    MD - MDT Decap, MT - MDT Threshold Crossed, MH - MDT interface handle
    CD - Conditional Decap, MPLS - MPLS Decap, EX - Extranet
    MoFE - MoFRR Enabled, MoFS - MoFRR State, MoFP - MoFRR Primary
    MoFB - MoFRR Backup, RPFID - RPF ID Set, X - VXLAN
Interface flags: F - Forward, A - Accept, IC - Internal Copy,
    NS - Negate Signal, DP - Don't Preserve, SP - Signal Present,
    II - Internal Interest, ID - Internal Disinterest, LI - Local Interest,
    LD - Local Disinterest, DI - Decapsulation Interface
    EI - Encapsulation Interface, MI - MDT Interface, LVIF - MPLS Encap,
    EX - Extranet, A2 - Secondary Accept, MT - MDT Threshold Crossed,
    MA - Data MDT Assigned, LMI - mLDP MDT Interface, TMI - P2MP-TE MDT Interface
    IRMI - IR MDT Interface
(1.1.1.1,229.1.1.1) RPF nbr: 1.1.1.1 Flags: RPF
  Up: 00:01:11
  Incoming Interface List
```

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

EVPN Active/Active Multihome Source with IGMP Snooping

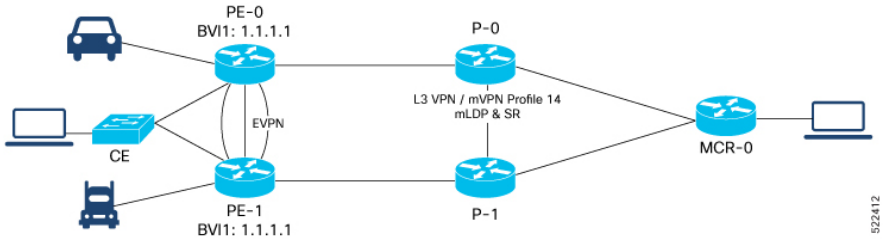
Table 3: Feature History Table

Feature Name	Release Information	Description
EVPN Active/Active Source with IGMP Snooping	Release 7.6.1	Supports multicast data packets from multihomed sources when the source is behind a BVI and the receivers at the core or on a bridge domain. This feature is not supported on the following variant: • N540-24Q8L2DD-SYS

From IOS XR 7.6.1 release, Cisco NCS 540 routers support multicast data traffic from multihomed sources. The source must be behind a BVI and the receivers at the core or on a bridge domain.

Multicast data traffic from multihome receivers is already supported on NCS 540 routers.

The following illustration shows the multicast data traffic between multihome sources and a multihome receiver.



In this illustration, the source is connected behind a CE, which is multihomed to PE-0 and PE-1. The PE-1 has a BVI with anycast IP address. The receiver that is behind the MCR-0 has a PIM connection toward the source. The following scenarios occur based on IGMP snooping:

IGMP Snooping: Activated

When the multicast data packets reach one of the PEs, the PE forwards the packets only to the interested receivers who have sent a request, including the EVPN OLE. Packets are recycled for BVI.

The recycled packets for BVI do an L3 lookup and if there is a route which exists with BVI as accept interface, the packet is forwarded to the L3 OLE.

EVPN replicated packets from PE-0 reach the other PE-1, one copy is replicated for BVI, and only the interested members and ACs receive the packets. The same packets are not returned to the CE due to SHL label filtering for the EVPN traffic. The replicated packets for the BVI are dropped due to split horizon group (SHG) filtering as EVPN and BVI belong to the same SHG.

IGMP Snooping: Dectivated

When the multicast data packets reach one of the PEs, it floods the bridge domain, recycles for BVI, and floods all the ACs, including the EVPN OLE.

The recycled packets for BVI do an L3 lookup and if there is a route which exists with BVI as accept interface, the packet is forwarded to the L3 OLE.

EVPN replicated packets from PE-0 reach the other PE-1, which flood the bridge domain. As part of this, one copy is replicated for BVI and flooded to all members and ACs. The same packets are not returned to the CE due to SHL label filtering for the EVPN traffic. The replicated packets for the BVI are dropped due to split horizon group (SHG) filtering as EVPN and BVI belong to the same SHG.

Multicast Route Statistics

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

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

Restrictions and Usage Guidelines for Multicast Route Statistics

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.
- Multicast route statistics for egress direction is supported only for phy intf, phy sub-inf, bundle intf, and bundle sub-intf. Rate is not supproted for egress stats.
- When ACL is mapped with **hw-module router-stats** configuration, you can’t modify the ACL. To modify ACLs that are mapped with router-stats, remove the existing **hw-module router-stats** configuration and update the ACL entries. Then, configure the **hw-module router-stats** again.

This feature supports:

- L3 Multicast traffic
- MVPN GRE
- Multicast over GRE
- LSM
- Default VRFs.

Configure Multicast Route Statistics

Table 4: Feature History Table

Feature Name	Release Information	Description
YANG Data Models for Multicast Interface Counters	Release 7.4.1	This feature introduces YANG data model support for multicast packets, in and out bytes per interface and sub-interface. With this feature, you can programmatically retrieve the operational details of multicast interfaces. You can access the data models from the Github repository.

Configuring multicast route statistics includes these main tasks:

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

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

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

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

Note

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

  For example:

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

- If you configure **hw-module route-stats** on both vrf default and vpn routes for either IPv4 or IPv6 ACLs, then to switch vrf default to other VRF, remove the configuration of the existing **hw-module route-stats** and commit it, and then configure the hw-module stats with the required vrf and commit it.
- If you configure **hw-module route-stats** on both vrf default and vpn routes for either IPv4 or IPv6 ACLs, then to switch vrf default to other VRF, remove the configuration of the existing **hw-module route-stats** and commit it, and then configure the hw-module stats with the required vrf and commit it.

Verification

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

Note The ingress stats are always per S, G.

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

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

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

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

IPv6 Egress Multicast Route Statistics Example

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

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

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

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

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

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

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

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

The following show command displays interface accounting stats for egress:

```
Router# show interfaces bundle-ether 100.1001 accounting rates
Mon Aug 26 15:56:41.738 IST
Bundle-Ether100.1001
                              Ingress                        Egress
  Protocol             Bits/sec         Pkts/sec     Bits/sec         Pkts/sec
  IPV4_MULTICAST              0                0     11455000              990
  IPV6_MULTICAST              0                0     11455000              990
  ARP                         0                0            0                0
  IPV6_ND                     0                0            0                0
```

Statistics for Egress Multicast Traffic Route Rate

Table 5: Feature History Table

Feature Name	Release Information	Feature Description

Statistics for Egress Multicast Traffic Route Rate	Release 7.11.1	With the ability to now view the route rates or rate of data being forwarded or transmitted per interface, you can monitor your network performance at a granular level, effectively troubleshoot network issues, and have greater control over bandwidth management.
		Previously, you could view the route rates only at the line card level.
		This feature introduces the following changes:
		• **CLI**
		  • The **rate** keyword is introduced in the **show mrib route** command.
		• **YANG Data Model**
		  • New XPaths for `Cisco-IOS-XR-mfwd-oper.yang` (see GitHub, YANG Data Models Navigator).

Multicast and interface statistics are used for accounting purpose. Route rates play a crucial role in identifying any potential traffic drops. By introducing support for multicast route rates and per outgoing interfac (OIF) statistics on compatible platforms, this feature aims to leverage per-source group (SG) telemetry traffic data. This data enables us to efficiently detect where traffic drops occur and identify the router causing them, both at the node and interface levels. As a result, you can improve network monitoring and troubleshooting by using streaming telemetry and enhanced data analysis.

Verify Multicast Traffic Route Rate Statistics

To verify the OIF egress rates per mroute, use the following example configuration:

```
RP/0/RSP0/CPU0:LA# show mrib route rate

Fri Dec 17 19:30:21.733 UTC
(11.1.1.2,232.1.1.1) RPF nbr: 11.1.1.2 Flags: RPF
  Up: 00:40:52
  Incoming Interface List
    HundredGigE0/0/0/2 Flags: A, Up: 00:40:52
    Node Rate (0/0/CPU0):    99 pps / 38407 bps
      HW Incoming count: 205444 packets
      HW Drop count:     0 packets
  Outgoing Interface List
    HundredGigE0/0/0/1 Flags: F NS, Up: 00:40:52
    Node Rate (0/0/CPU0):      99 pps / 38407 bps
      HW Forwarding count: 205444 packets
      HW Drop count:       0 packets
     Interface Rates:
        Interface: HundredGigE0/0/0/1
        Outgoing Packet Rate (PPS rate / BPS rate): 100 / 100
        HW Forwarding count: 10000 packets
        HW Drop count: 0 packets


(11.1.1.2,232.2.2.2) RPF nbr: 11.1.1.2 Flags: RPF
  Up: 00:40:52
  Incoming Interface List
    HundredGigE0/0/0/2 Flags: A, Up: 00:40:52
   Node Rate (0/0/CPU0):    74 pps / 28798 bps
      HW Incoming count: 154084 packets
      HW Drop count:     0 packets
  Outgoing Interface List
    HundredGigE0/0/0/1 Flags: F NS, Up: 00:40:52
    Node Rate (0/0/CPU0):      74 pps / 28798 bps
      HW Forwarding count: 154084 packets
      HW Drop count:       0 packets
     Interface Rates:
        Interface: HundredGigE0/0/0/1
        Outgoing Packet Rate (PPS rate / BPS rate): 100 / 100
        HW Forwarding count: 10000 packets
        HW Drop count: 0 packets
```

Bundle Member Selection

Table 6: Feature History Table

Feature Name	Release Information	Feature Description
Bundle Member Selection	Release 7.3.1	This feature enables selecting a bundle member in the control plane to steer the L2 and L3 multicast traffic traversing over bundle at the egress NP. This feature helps optimize fabric bandwidth as the member selection is performed in the control plane.

This feature enables selecting a bundle member in the control plane to steer the L2 and L3 multicast traffic traversing over bundle at the egress NP.

This feature brings following benefits:

- Optimizes fabric bandwidth as the member selection is performed in the control plane
- Reduces NP bandwidth and processing as number of OLE replications are less
- Supports bundle member change in MVPN head node with local receiver

Multicast Over IPV4 Unicast GRE Tunnels

Table 7: Feature History Table

Feature Name	Release Information	Feature Description
Support for Multicast Over IPV4 Unicast GRE Tunnels	Release 7.2.2	This feature allows encapsulation of multicast packets using GRE tunnels, thereby enabling transport of multicast packets securely between source and destination routers located in different IP clouds.

Use IPV4 unicast GRE tunnels to transport multicast traffic securely over the network.

Generic Routing Encapsulation (GRE) is a tunneling protocol that encapsulates and transport packets of one protocol over another protocol.

If you want to send multicast packets from a source to destination router configured with a different routing protocol, you can encapsulate the packets using GRE unicast tunnels. The encapsulated packets are forwarded like any other IPv4 unicast packet to the destination endpoint of the tunnel. The destination router then de-encapsulates the packet to retrieve the multicast packets.

For more information Configuring GRE Tunnels.

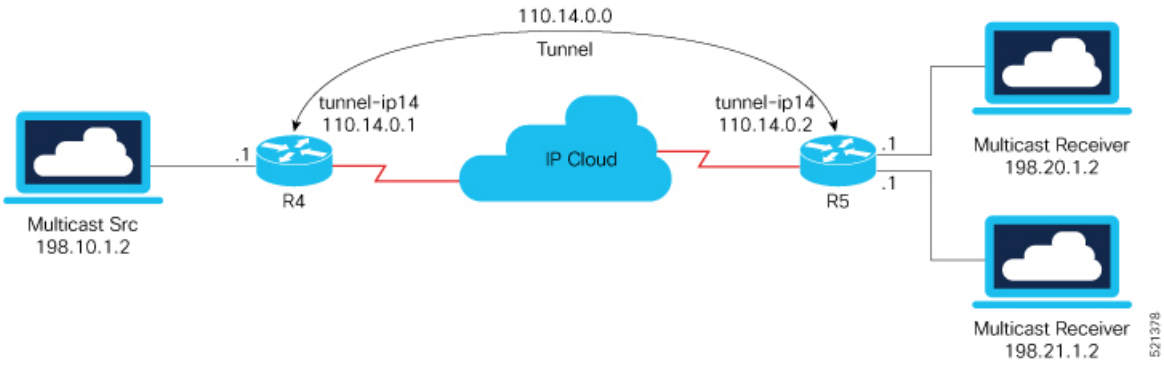
Restrictions

- ECMP and LAG hash based on GRE header is not supported on the NCS 5500 line cards line cards.
- Multicast over GRE with L3VPN is only supported on UFI and not on the NCS 5500 line cards and NCS 5700 line cards.
- Only SSM V4 and V6 address family traffic is supported.
- Supports up to 500 GRE tunnels
- Only up to 16 unique source IP addresses are supported for the tunnel source
- Multicast over GRE tunnel is supported only with 2-pass GRE tunnel configuration.
- Configurable MTU is not supported on Single-pass GRE interface, but supported on 2-pass GRE interface.
- This is a native Multicast over GRE feature and not Multicast VPN (mVPN) Profile 0 or Rosen GRE.

Configuration

In this topology, the multicast source (198.10.1.2) is connected to R4. The multicast receivers are connected to R5 and is configured to receive multicast packets. Separating the source (R4) and receiver (R5) is an IP cloud, which is not configured for multicast routing.

Multicast packets are encapsulated with GRE headers and transported via GRE tunnel (tunnel-ip14).



Configuration Example

```
R4

interface TenGigE0/0/0/32.1
 ipv4 address 198.10.1.1 255.255.255.0
 ipv6 address 2002:10:1::1/64
 encapsulation dot1q 1
!

interface Loopback14
 ipv4 address 10.10.10.14 255.255.255.255
!
interface tunnel-ip14
 ipv4 address 110.14.0.1 255.255.255.0
 ipv6 address 110:14::1/64
```

```
 tunnel mode gre ipv4
 tunnel source 10.10.10.14
 tunnel destination 20.20.20.14
!
router ospf core_native_mcast
 nsr
 router-id 10.10.10.1
 area 0.0.0.0
  interface Bundle-Ether121
  !
  interface Loopback14
  !

router ospf mogre_edge_native_mcast
 nsr
 router-id 10.10.10.11
 area 0.0.0.0
  interface tunnel-ip14
  !
  interface TenGigE0/0/0/32.1


R5

interface TenGigE0/0/0/0.1
 ipv4 address 198.20.1.1 255.255.255.0
 ipv6 address 2002:20:1::1/64
 encapsulation dot1q 1
!


interface TenGigE0/0/0/1.1
 ipv4 address 198.21.1.1 255.255.255.0
 ipv6 address 2002:21:1::1/64
 encapsulation dot1q 1
!

interface Loopback14
 ipv4 address 20.20.20.14 255.255.255.255
!

interface tunnel-ip14
 ipv4 address 110.14.0.2 255.255.255.0
 ipv6 address 110:14::2/64
 tunnel mode gre ipv4
 tunnel source 20.20.20.14
 tunnel destination 10.10.10.14
!


router ospf core_native_mcast
 nsr
 router-id 20.20.20.1
 area 0.0.0.0
  interface Bundle-Ether121
  !
  interface Loopback14


router ospf mogre_edge_native_mcast
 nsr
 router-id 20.20.20.11
 area 0.0.0.0
  interface tunnel-ip14
  !
```

```
  interface TenGigE0/0/0/0.1
  !
  interface TenGigE0/0/0/1.1
```

Verification

In this example, Router R4 receives joins for group 232.1.0.0 from source 198.10.1.2 connected to R4. PIM joins are traversed from R5 to R4, as shown in the **show PIM topology** command output. The joins are learnt on Router R4 via tunnel-ip, as it acts as the transport layer.

```
RP/0/RP0/CPU0:R4# show pim topology 232.1.0.0 198.10.1.2

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

(198.10.1.2,232.1.0.0)SPT SSM Up: 00:33:51
JP: Join(now) RPF: TenGigE0/0/0/32.1,198.10.1.2* Flags:
  tunnel-ip14                 00:29:34  fwd Join(00:03:00)
```

From the following **show mrib route** command output, you can see that TenGigE0/0/0/32.1 is the incoming interface and tunnel-ip14 is the outgoing interface for (S,G) 198.10.1.2, 232.1.0.0.

```
RP/0/RP0/CPU0:R4# show mrib route 232.1.0.0 detail
IP Multicast Routing Information Base
Entry flags: L - Domain-Local Source, E - External Source to the Domain,
    C - Directly-Connected Check, S - Signal, IA - Inherit Accept,
    IF - Inherit From, D - Drop, ME - MDT Encap, EID - Encap ID,
    MD - MDT Decap, MT - MDT Threshold Crossed, MH - MDT interface handle
    CD - Conditional Decap, MPLS - MPLS Decap, EX - Extranet
    MoFE - MoFRR Enabled, MoFS - MoFRR State, MoFP - MoFRR Primary
    MoFB - MoFRR Backup, RPFID - RPF ID Set, X - VXLAN
Interface flags: F - Forward, A - Accept, IC - Internal Copy,
    NS - Negate Signal, DP - Don't Preserve, SP - Signal Present,
    II - Internal Interest, ID - Internal Disinterest, LI - Local Interest,
    LD - Local Disinterest, DI - Decapsulation Interface
    EI - Encapsulation Interface, MI - MDT Interface, LVIF - MPLS Encap,
    EX - Extranet, A2 - Secondary Accept, MT - MDT Threshold Crossed,
    MA - Data MDT Assigned, LMI - mLDP MDT Interface, TMI - P2MP-TE MDT Interface
    IRMI - IR MDT Interface, TRMI - TREE SID MDT Interface, MH - Multihome Interface

(198.10.1.2,232.1.0.0) Ver: 0x6e42 RPF nbr: 198.10.1.2 Flags: RPF, FGID: 16915, Statistics
 enabled: 0x0, Tunnel RIF: -1
  Up: 00:32:54
 Incoming Interface List
    TenGigE0/0/0/32.1 Flags: F A LI, Up: 00:32:54
```

```
  Outgoing Interface List
    tunnel-ip14 (0/0/0) Flags: F NS, Up: 00:28:37

RP/0/RP0/CPU0:R4#sh mfib route 232.1.0.0 198.10.1.2

IP Multicast Forwarding Information Base
Entry flags: C - Directly-Connected Check, S - Signal, D - Drop,
  IA - Inherit Accept, IF - Inherit From, EID - Encap ID,
  ME - MDT Encap, MD - MDT Decap, MT - MDT Threshold Crossed,
  MH - MDT interface handle, CD - Conditional Decap,
  DT - MDT Decap True, EX - Extranet, RPFID - RPF ID Set,
  MoFE - MoFRR Enabled, MoFS - MoFRR State, X - VXLAN
Interface flags: F - Forward, A - Accept, IC - Internal Copy,
  NS - Negate Signal, DP - Don't Preserve, SP - Signal Present,
  EG - Egress, EI - Encapsulation Interface, MI - MDT Interface,
  EX - Extranet, A2 - Secondary Accept
Forwarding/Replication Counts: Packets in/Packets out/Bytes out
Failure Counts: RPF / TTL / Empty Olist / Encap RL / Other

(198.10.1.2,232.1.0.0),  Flags:
  Up: 00:35:39
  Last Used: never
  SW Forwarding Counts: 0/0/0
  SW Replication Counts: 0/0/0
  SW Failure Counts: 0/0/0/0/0
 tunnel-ip14 (0xe0) Flags:  NS, Up:00:31:16


RP/0/RP0/CPU0:R4# show mfib hardware route 232.1.0.0 198.10.1.2 location 0/0/cPU0
Route (198.10.1.2: 232.1.0.0)
   HAL PD context
       VRF ID: 0 Core MCID : 0 Core backup MCID 0

   HAL Ingress route context:
       Route FGID: 16915 RPF IF signal : not-set Local receivers: set
       Encap ID flag: not-set, Encap ID: 0
       Tunnel RIF: 0x0
       Statistics enabled: not-set

       Ingress engine context:
          local_route: set, is_accept_intf_bvi: not-set is_tun_rif_set:not-set
           VRF ID: 0 RPF ID:0 Tunnel RIF:0x0
   HAL Egress route context:
       RPF ID: 0

       Egress engine context:
          out_of_sync: not-set, local_intf: not-set
           bvi_count: 0

   DPA Route context:
      Handle: 30895ef540
      Transaction ID: 91864
      Number of OLE: 4   VRF ID: 0
      Incoming interface : Te0/0/0/32.1 A_intf_id: 0x39 Merged flag 0
      Tunnel RIF : 0x0 FGID: 16915
      FEC ID : 0x2001f888 Punt action: 0x0
      TCAM entry ID : 0x0 IPMC action: 0x4 FEC Accessed 1
      L3 Intf Refhandle : 0x308ccbd448 L3 interface ref key: 0x0
      Statistics enabled : not-set Statistics activated : not-set

      Egress Route OLEs:
        Handle: 308e669960
        Transaction ID: 267707
        NPU ID: 0 Outgoing intf: ti14
```

```
        OLE Type : Gre tunnel interface
        outgoing port : 0x0 cud: 0x13878 is_bundle: 0
        Sys_port : 0x0 mpls encap id: 0x0 LAG ID: 0
        is_pw_access: 0 pw_encap_id:0
        L3 intf refhndl : 0x308d47bee8 L3 intf refkey: 0x200040fc
        L2 Port refhandle : 0x0 L2 Port refkey: 0x0
        MPLS nh refhandle : 0x0 MPLS nh refkey: 0x0
        LAG port refhandle : 0x0 LAG port refkey: 0x0
        EFP-Visibility: not-set
        Total fwd packets : 0 Total fwd bytes: 0
```

The following command output shows that the PIM joins received from TenGigE0/0/0/0.1 and TenGigE0/0/0/1.1 for group 232.1.0.0.

```
RP/0/RP0/CPU0:R5# show pim topology 232.1.0.0 198.10.1.2

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

(198.10.1.2,232.1.0.0)SPT SSM Up: 00:44:09
JP: Join(00:00:02) RPF: tunnel-ip14,110.14.0.1 Flags:
  TenGigE0/0/0/0.1            00:44:09  fwd LI LH
  TenGigE0/0/0/1.1            00:44:06  fwd LI LH
```

The traffic received from `tunnel-ip` is then forwarded to multicast receivers TenGigE0/0/0/0.1 and TenGigE0/0/0/1.1 acting as OLE, as shown in the following output.

```
RP/0/RP0/CPU0:R5# show mrib route 232.1.0.0 detail
IP Multicast Routing Information Base
Entry flags: L - Domain-Local Source, E - External Source to the Domain,
    C - Directly-Connected Check, S - Signal, IA - Inherit Accept,
    IF - Inherit From, D - Drop, ME - MDT Encap, EID - Encap ID,
    MD - MDT Decap, MT - MDT Threshold Crossed, MH - MDT interface handle
    CD - Conditional Decap, MPLS - MPLS Decap, EX - Extranet
    MoFE - MoFRR Enabled, MoFS - MoFRR State, MoFP - MoFRR Primary
    MoFB - MoFRR Backup, RPFID - RPF ID Set, X - VXLAN
Interface flags: F - Forward, A - Accept, IC - Internal Copy,
    NS - Negate Signal, DP - Don't Preserve, SP - Signal Present,
    II - Internal Interest, ID - Internal Disinterest, LI - Local Interest,
    LD - Local Disinterest, DI - Decapsulation Interface
    EI - Encapsulation Interface, MI - MDT Interface, LVIF - MPLS Encap,
    EX - Extranet, A2 - Secondary Accept, MT - MDT Threshold Crossed,
    MA - Data MDT Assigned, LMI - mLDP MDT Interface, TMI - P2MP-TE MDT Interface
    IRMI - IR MDT Interface, TRMI - TREE SID MDT Interface, MH - Multihome Interface

(198.10.1.2,232.1.0.0) Ver: 0x8ef6 RPF nbr: 110.14.0.1 Flags: RPF, FGID: 28465, Statistics
```

```
 enabled: 0x0, Tunnel RIF: -1
  Up: 00:42:42
  Incoming Interface List
    tunnel-ip14 Flags: A, Up: 00:38:24
  Outgoing Interface List
    TenGigE0/0/0/0.1 Flags: F NS LI, Up: 00:42:42
    TenGigE0/0/0/1.1 Flags: F NS LI, Up: 00:42:40

RP/0/RP0/CPU0:R5# show mfib route 232.1.0.0 198.10.1.2

IP Multicast Forwarding Information Base
Entry flags: C - Directly-Connected Check, S - Signal, D - Drop,
  IA - Inherit Accept, IF - Inherit From, EID - Encap ID,
  ME - MDT Encap, MD - MDT Decap, MT - MDT Threshold Crossed,
  MH - MDT interface handle, CD - Conditional Decap,
  DT - MDT Decap True, EX - Extranet, RPFID - RPF ID Set,
  MoFE - MoFRR Enabled, MoFS - MoFRR State, X - VXLAN
Interface flags: F - Forward, A - Accept, IC - Internal Copy,
  NS - Negate Signal, DP - Don't Preserve, SP - Signal Present,
  EG - Egress, EI - Encapsulation Interface, MI - MDT Interface,
  EX - Extranet, A2 - Secondary Accept
Forwarding/Replication Counts: Packets in/Packets out/Bytes out
Failure Counts: RPF / TTL / Empty Olist / Encap RL / Other

(198.10.1.2,232.1.0.0),  Flags:
  Up: 00:43:30
  Last Used: never
  SW Forwarding Counts: 0/0/0
  SW Replication Counts: 0/0/0
  SW Failure Counts: 0/0/0/0/0
  tunnel-ip14 Flags:  A, Up:00:39:09
  TenGigE0/0/0/0.1 Flags:  NS, Up:00:43:30
  TenGigE0/0/0/1.1 Flags:  NS, Up:00:43:17


RP/0/RP0/CPU0:R5# show mfib hardware route 232.1.0.0 198.10.1.2 location 0/0/CPU0

Route (198.10.1.2: 232.1.0.0)
   HAL PD context
       VRF ID: 0 Core MCID : 0 Core backup MCID 0

   HAL Ingress route context:
       Route FGID: 28465 RPF IF signal : not-set Local receivers: set
       Encap ID flag: not-set, Encap ID: 0
       Tunnel RIF: 0x0
       Statistics enabled: not-set

       Ingress engine context:
          local_route: set, is_accept_intf_bvi: not-set is_tun_rif_set:not-set
           VRF ID: 0 RPF ID:0 Tunnel RIF:0x0
   HAL Egress route context:
       RPF ID: 0

       Egress engine context:
          out_of_sync: not-set, local_intf: not-set
           bvi_count: 0

   DPA Route context:
      Handle: 308852aed0
      Transaction ID: 228831
      Number of OLE: 2   VRF ID: 0
      Incoming interface : ti14 A_intf_id: 0x43 Merged flag 0
      Tunnel RIF : 0x0 FGID: 28465
      FEC ID : 0x2001fd37 Punt action: 0x0
```

```
TCAM entry ID : 0x0 IPMC action: 0x4 FEC Accessed 1
L3 Intf Refhandle : 0x308d76fee8 L3 interface ref key: 0x0
Statistics enabled : not-set Statistics activated : not-set

Egress Route OLEs:
  Handle: 308e27d930
  Transaction ID: 103691
  NPU ID: 0 Outgoing intf: Te0/0/0/1.1
  OLE Type : Main Interface
  outgoing port : 0x1d cud: 0x0 is_bundle: 0
  Sys_port : 0x0 mpls encap id: 0x0 LAG ID: 0
  is_pw_access: 0 pw_encap_id:0
  L3 intf refhndl : 0x308cf40b08 L3 intf refkey: 0x4178
  L2 Port refhandle : 0x308cf49358 L2 Port refkey: 0xe8
  MPLS nh refhandle : 0x0 MPLS nh refkey: 0x0
  LAG port refhandle : 0x0 LAG port refkey: 0x0
  EFP-Visibility: not-set
  Total fwd packets : 0 Total fwd bytes: 0

  NPU ID: 0 Outgoing intf: Te0/0/0/0.1
  OLE Type : Main Interface
  outgoing port : 0x1e cud: 0x0 is_bundle: 0
  Sys_port : 0x0 mpls encap id: 0x0 LAG ID: 0
  is_pw_access: 0 pw_encap_id:0
  L3 intf refhndl : 0x308cf3e668 L3 intf refkey: 0x4168
  L2 Port refhandle : 0x308cf470a8 L2 Port refkey: 0xf0
  MPLS nh refhandle : 0x0 MPLS nh refkey: 0x0
  LAG port refhandle : 0x0 LAG port refkey: 0x0
  EFP-Visibility: not-set
  Total fwd packets : 0 Total fwd bytes: 0
```

Associated Commands

- interface tunnel-ip
- tunnel mode
- tunnel source
- tunnel destination

Use Case: Video Streaming

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

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

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

1. Encoding devices in digital primary headends, encode one or more video channels into a Moving Pictures Expert Group (MPEG) stream which is carried in the network via IP multicast.
2. Devices at video branch office are configured by the operator to request the desired multicast content via IGMP joins.

3. The network, using PIM-SSM as its multicast routing protocol, routes the multicast stream from the digital primary headend to edge device receivers located in the video branch office. These edge devices could be edge QAM devices which modulate the MPEG stream for an RF frequency, or CMTS for DOCSIS.

Figure 5: Video Streaming

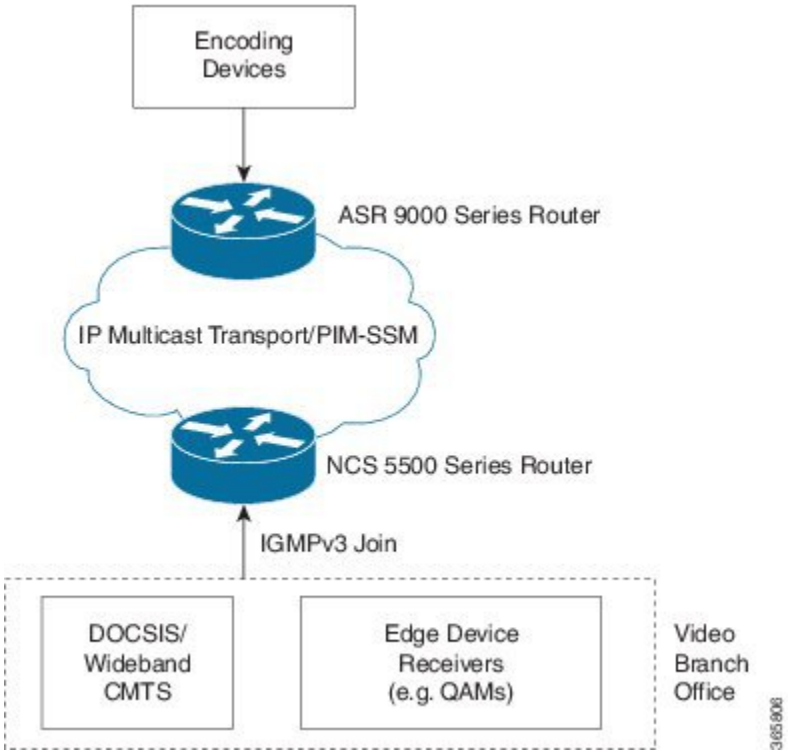
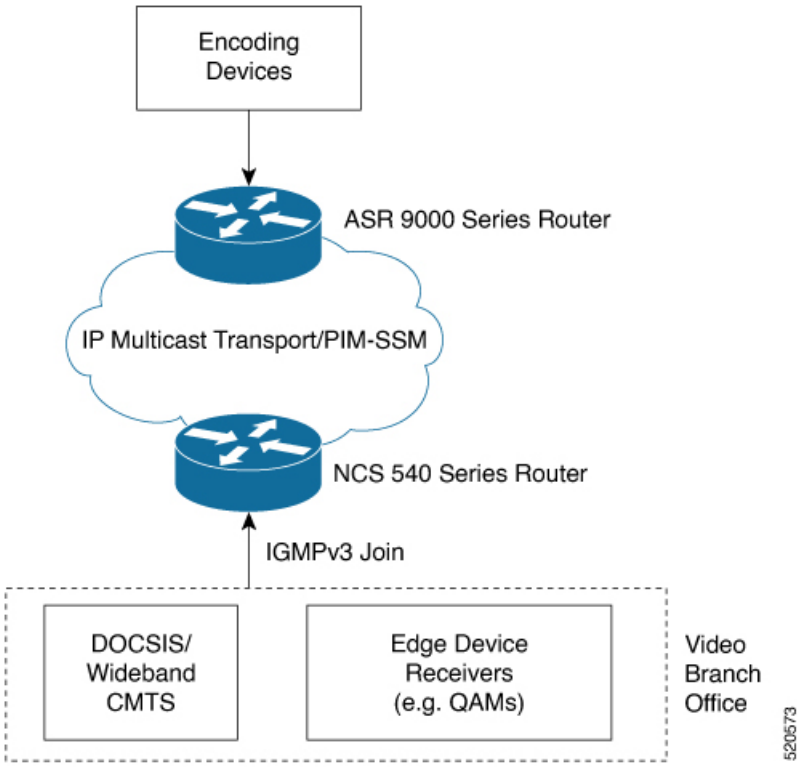


Figure 6: Video Streaming



Multicast Label Distribution Protocol

This section contains information related to Multicast Label Distribution Protocol (MLDP) and the associated features.

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

Characteristics of MLDP Profiles on Core

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

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

- Profile 14—Partitioned MDT - MLDP P2MP - BGP-AD - BGP C-mcast Signaling
- Profile 17—Default MDT - MLDP - P2MP - BGP-AD - PIM C-mcast Signaling

Point-to-Multipoint TE Profiles on Core and Edge Routers

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

- Profile 8—Global P2MP-TE
- Profile 10—VRF Static-P2MP-TE with BGP AD
- Profile 22—RSVP-TE P2MP

Label Switching Multicast for Edge Router

The following MLDP profiles are supported when the router is configured as an edge router:

- Profile 6—VRF MLDP - In-Band Signaling
- Profile 7—Global MLDP In-band Signaling
- Profile 14—MLDP Partitioned MDT P2MP with BGP AD and BGP-C Multicast Signaling

Multicast MLDP Profile 14 support on an Edge Router

Table 8: Feature History Table

Feature Name	Release Information	Feature Description
Protocol Independent Multicast (PIM) SM for Multicast VPN (MVPN) Profile 14	Release 7.10.1	With this release, MVPN profile 14 is now extended to support PIM SM mode for IPv4 and static RP.
PIM SM for mVPN Profile 14	Release 7.5.1	With this feature, MVPN profile 14 is now extended to support PIM SM mode for IPv4 and static RP. PIM SM for MVPN is not supported on Cisco NC57 line cards.
MLDP Profile 14 support on an Edge Router	Release 7.3.1	This feature is now supported on Cisco NCS 540 routers.

The MLDP Profile 14 is supported when the router is configured as an edge router.

IP based transport network is a cost efficient and convenient alternative to deliver video services combined with the delivery of other IP based services. To deliver IPTV content MLDP Profile 14 also called as the partitioned MDT, is supported when a router is configured as an edge router.

These are the characteristics of the profile 14:

- Customer traffic is SSM.
- Inter-AS Option A, B and C is supported.
- All PEs must have a unique BGP Route Distinguisher (RD) value.

Configuration Example for mLDP Profile 14 on Edge Routers

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

router pim
 vrf one
  address-family ipv4
   rpf topology route-policy rpf-for-one
   mdt c-multicast-routing bgp
   !
   interface GigabitEthernet0/1/0/0
    enable
   !
  !
 !
!

route-policy rpf-for-one
  set core-tree mldp-partitioned-p2mp
end-policy
!

multicast-routing
 vrf one
  address-family ipv4
   mdt source Loopback0
   mdt partitioned mldp ipv4 p2mp
   rate-per-route
   interface all enable
   bgp auto-discovery mldp
   !
   accounting per-prefix
  !
 !
!

mpls ldp
 mldp
  logging notifications
  address-family ipv4
  !
 !
!
router bgp 100
bgp router-id 23.23.23.23
bgp graceful-restart
address-family ipv4 unicast
  redistribute connected
```

```
!
address-family vpnv4 unicast
!
address-family ipv6 unicast
  redistribute connected
!
address-family vpnv6 unicast
!
address-family ipv4 mdt
!
address-family ipv4 mvpn
!
address-family ipv6 mvpn
!
neighbor 5.5.5.5
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
  !
  address-family vpnv4 unicast
  !
  address-family ipv6 unicast
  !
  address-family vpnv6 unicast
  !
  address-family ipv4 mdt
  !
  address-family ipv4 mvpn
  !
  address-family ipv6 mvpn
  !
neighbor 11.11.11.11
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
  !
  address-family vpnv4 unicast
  !
  address-family ipv6 unicast
  !
  address-family vpnv6 unicast
  !
  address-family ipv4 mdt
  !
  address-family ipv4 mvpn
  !
  address-family ipv6 mvpn
  !
!
neighbor 21.21.21.21
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
  !
  address-family vpnv4 unicast
  !
  address-family ipv6 unicast
  !
  address-family vpnv6 unicast
  !
  address-family ipv4 mdt
  !
  address-family ipv4 mvpn
  !
```

```
  address-family ipv6 mvpn
  !
!
 vrf vrf2
  rd 2:2
   address-family ipv4 unicast
   redistribute connected
  !
  address-family ipv6 unicast
   redistribute connected
  !
  address-family ipv4 mvpn
  !
  address-family ipv6 mvpn
  !
!
!
```

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, and 14.

For more information about the characteristics of each of the mLDP Profiles, Characteristics of mLDP Profiles.

Benefits of LSM MLDP based MVPN

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

- It leverages the MPLS infrastructure for transporting IP multicast packets, providing a common data plane for unicast and multicast.
- It applies the benefits of MPLS to IP multicast such as Fast ReRoute (FRR) and
- It eliminates the complexity associated PIM.

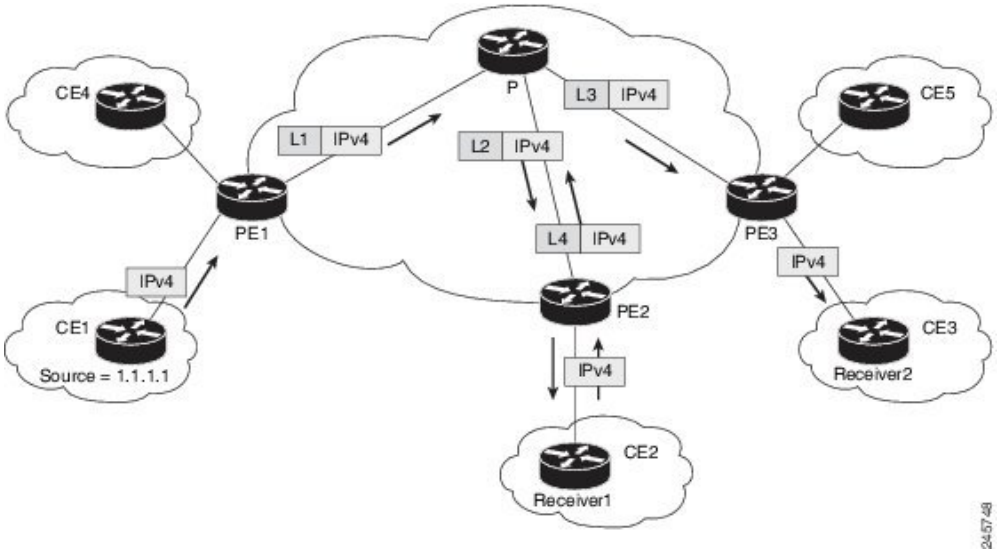
Configuring MLDP MVPN

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

the source multicast VPN Routing and Forwarding (mVRF) on the receiver Provider Edge (PE) router or configure the receiver mVRF on the source PE. MLDP MVPN is supported for both intranet and extranet.

Note If a tail-end router has multiple VRFs with the same import Route Target (RT), issues can occur. These issues arise when the intended Label Switched Multicast (LSM) is not an extranet setup. The router may perform incorrect signaling. It may also fail to build core trees.

Figure 7: MLDP based MPLS Network for Core and Edge Routers



Packet Flow in mLDP-based Multicast VPN

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

Realizing a mLDP-based Multicast VPN

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

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

- P2MP/MP2MP LSPs for MVPN based on MS-PMSI or Multidirectional Selective Provider Multicast Service Instance (Partitioned E-LAN).

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

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

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

Restrictions for mLDP on Edge Routers

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

- NETCONF/YANG on MVPN for Profile 6 and Profile 7 is not supported.
- MLDP ping traceroute is not supported.
- IPv6 BVI is not supported.
- Netflow for MPLS-encapsulated multicast packets is not supported.
- MLDP Fast-Reroute (FRR) is supported for Profile 14 only.
- The MLDP Fast-Reroute and Egress Traffic Management (ETM) features are mutually exclusive.

Configure MLDP MVPN for Intranet

Use this task to configure MLDP MVPN on an intranet.

Procedure

Step 1 Enable MPLS MLDP.

```
mpls ldp mldp
```

Step 2 Configure a VRF entry.

```
vrf vrf_name
address-family ipv4 unicast
import route-target route-target-ext-community
export route-target route-target-ext-community
```

Step 3 Configure VPN ID.

```
vrf vrf_name
vpn id vpn_id
```

Step 4 Configure MVPN routing and forwarding instance.

```
multicast-routing vrf vrf_name
address-family ipv4
mdt default mldp ipv4 root-node
```

Step 5 Configure the route distinguisher.

```
router bgp AS Number
vrf vrf_name
rd rd_value
```

Step 6 (Optional) Configure data MDTs.

```
multicast-routing vrf vrf_name
address-family ipv4
mdt data <1-255>
```

Step 7 Configure BGP MDT address family.

```
router bgp AS Number
address-family ipv4 mdt
```

Step 8 Configure BGP vpnv4 address family.

```
router bgp AS Number
address-family vpnv4 unicast
```

Step 9 Configure BGP IPv4 VRF address family.

```
router bgp AS Number
vrf vrf_name
address-family ipv4 unicast
```

Step 10 Configure PIM SM/SSM mode for the VRFs.

```
router pim
vrf vrf_name
address-family ipv4
rpf topology route-policy rosen_mvpn_mldp
```

Step 11 Configure route-policy. For each profile, a different route-policy is configured.

```
route-policy rosen_mvpn_mldp
set core-tree tree-type
pass
end-policy
```

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

Configuration Example for MLDP on Core

```
mpls ldp
 mldp
  logging notifications
```

```
  address-family ipv4
  !
 !
!
```

Flexible Algorithm for MLDP

Table 9: Feature History Table

Feature Name	Release Information	Feature Description
Flexible Algorithm for Multicast VPN profiles	Release 7.5.2	Flexible Algorithm is now available for the following profiles: • Profile 12: Default MDT - MLDP - P2MP - BGP-AD - BGP C-Mcast Signaling • Profile 14: Partitioned MDT - MLDP P2MP - BGP-AD - BGP C-Mcast Signaling
Flexible Algorithm for MLDP	Release 7.5.1	This feature gives you the flexibility to customize the metrics that IGP uses to route traffic for MLDP tunnels. With this feature, your router can generate two multicast streams for the same feed, thus ensuring low latency and high availability of multicast traffic. This feature introduces the keyword.

IGP determines the shortest path to send traffic through MLDP tunnels. However, at times, you may require to choose a path other than the shortest one. For instance, when you want to achieve low latency or want to send the traffic on a specific path to avoid a set of links or build totally two or more disjoint paths.

Multipoint LDP (mLDP) flexible algorithm allows you to customize the IGP path computation based on the business needs.

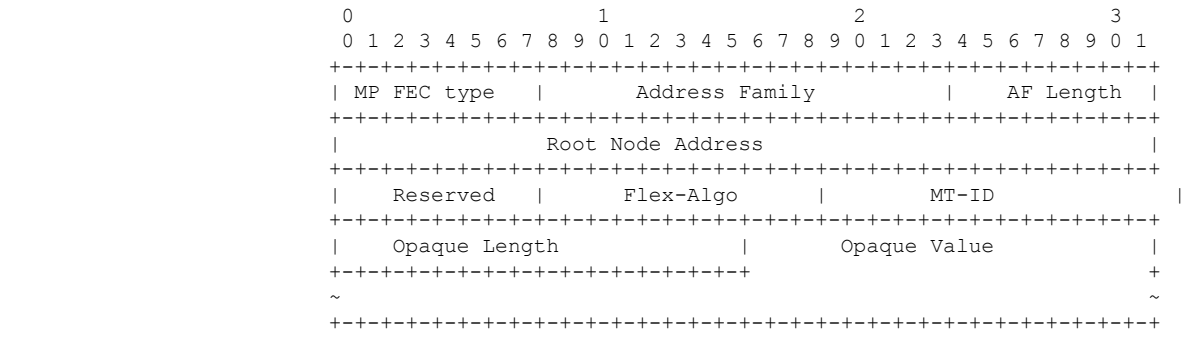
With this feature, you can get disjoint paths for two multicast streams. Each stream carries traffic within a separate network and helps reduce a connection loss or delay ensuring low latency and high availability of multicast traffic. This feature allows you to segregate multicast traffic to specific regions.

mLDP flexible algorithm is based on Segment Routing flexible algorithm that allows operators to customize IGP shortest path computation according to their own needs. For more information, see Enabling Segment Routing Flexible Algorithm.

To compute two different paths, the mLDP flexible algorithm uses a 2-tuple hash algorithm, which includes MPLS Multi-Topology Identifier (MT-ID) and IGP algorithm.

MPLS multi topology Identifier (MT-ID) is a unique identifier that is used to associate an LSP with multi topology. This identifier is part of the mLDP FEC encoding so that LDP peers are able to set up an MP LSP through their own defined policy and avoid conflicting policies for the same mLDP FEC. The MT-ID and IGP Algorithm must be part of the FEC so that different values result in unique MP-LSP FEC elements.

mLDP flexible algorithm is stored in IGP Algorithm (IPA) Registry field. A 16-bit reserved field is included and out of which 8 bits are used for flexible algorithm. The root is an IP address identifying the Root of an MLDP-P2MP tree. Only IPv4 address is supported.



For example, two flexible algorithms are required to implement a disjoint-paths scenario.

Leafs belonging to the first plane are associated with flexible algorithm 130 and leafs belonging another plane are associated with flexible algorithm 128.

Restrictions

The following features are not supported with flexible algorithm:

- PIM BiDir in the core
- Carrier supporting carrier (CSC)
- Inter-AS
- Extranet
- Default MDT
- MVPN profiles except profile-14

How to configure mLDP Flexible Algorithm?

Configure partitioned MDT with flexible algorithm MVPN profile:

```
Router #configure
Router(config)# multicast-routing
Router(config-mcast)# vrf red
Router(config-mcast-red)# address-family ipv4
Router(config-mcast-red-ipv4)# mdt partitioned mldp ipv4 p2mp flex-algo 128
```

Configure data MDT with flexible algorithm:

```
Router# configure
Router(config)# multicast-routing
Router(config-mcast)# vrf red
```

```
Router(config-mcast-red)# address-family ipv4
Router(config-mcast-red-ipv4)# mdt data mldp 10 flex-algo 130
```

Configure route policy for Data MDT with flexible algorithm:

```
Router # configure
Router(config)# multicast-routing vrf red address-family ipv4
Router(config-mcast-red-ipv4)# mdt data mldp 10 route-policy rpl-for-red flex-algo 130
```

Note We recommend having 1:1 allocation of Data MDTs with number of flows. However, you can configure less Data MDTs than the number of flows. But you may not see the expected results. You cannot modify the flexible algorithm configuration without removing the MDT configuration. You must reconfigure MDT with the new flexible algorithm. Other combinations like no flexible algorithm to flexible algorithm are not supported.

Configuration Example

The route-policy and flexible algorithm are configured on Data MDT CLI.

In this example,

- Multicast groups 232.1.1.1 and 232.1.1.2 match the route policy it uses the Data MDT created with flexible algorithm 128.
- Multicast groups 232.1.1.3 and 232.1.1.4 also match the route-policy it uses the Data MDT created with flexible algorithm 129.

  All other groups do not match the route-policy so it uses the Data MDT created with flexible algorithm 130 which is configured in Data MDT CLI.

```
route-policy c1-data-mdt
  if destination in (232.1.1.1, 232.1.1.2) then
    set flex-algo 128
  elseif destination in (232.1.1.3, 232.1.1.4) then
    set flex-algo 129
  endif
end-policy
!
multicast-routing
 vrf red
  address-family ipv4
   mdt partitioned mldp ipv4 p2mp flex-algo 131
   mdt data mldp 100 route-policy c1-data-mdt flex-algo 130
  !
 !
!
```

In the following example, route-policy is configured on Data MDT CLI but without flexible algorithm.

```
route-policy c1-data-mdt
  if destination in (232.1.1.1, 232.1.1.2) then
    set flex-algo 128
  elseif destination in (232.1.1.3, 232.1.1.4) then
    set flex-algo 129
  endif
```

```
end-policy
!
multicast-routing
 vrf red
  address-family ipv4
   mdt partitioned mldp ipv4 p2mp flex-algo 131
   mdt data mldp 100 route-policy c1-data-mdt
  !
 !
!
```

Flexible algorithm is configured on Data MDT CLI and all groups uses the Data MDT created with flexible algorithm 130.

```
multicast-routing
 vrf red
  address-family ipv4
   mdt partitioned mldp ipv4 p2mp flex-algo 131
   mdt data mldp 100 flex-algo 130
```

In the following example, flexible algorithms are configured in Partitioned MDT. All groups use the Data MDT created with Flexible Algorithm 131 which is configured in Partitioned MDT CLI.

```
multicast-routing
 vrf red
  address-family ipv4
   mdt partitioned mldp ipv4 p2mp flex-algo 131
   mdt data mldp 100
```

Verification

```
Router# show mvpn vrf red context private

MVPN context information for VRF red (0xa99dbf8)

RD: 1:1 (Valid, IID 0x1), VPN-ID: 0:0
Import Route-targets : 2
  RT:10.0.0.4:0, BGP-AD
  RT:10.0.0.4:17, BGP-AD
BGP Auto-Discovery Enabled (I-PMSI added) , MS-PMSI sent
MLDP Core-tree data:
  MDT Name: Lmdtred, Handle: 0x8041b0, idb: 0xa9b1c18
  MTU: 1376, MaxAggr: 255, SW_Int: 30, AN_Int: 60
  RPF-ID: 9/0, C:0, O:1, D:0, CP:0
  MLDP Number of Roots: 0 (Local: 0), HLI: 0x00000, Rem HLI: 0x00000
  Discovery Type,Value: Flex-Algo,131
  Data MDT Discovery Type,Value: Flex-Algo,130
  Partitioned MDT: Configured, P2MP (RD:Not added, ID:Added), HLI: 0x00005, Loc Label:
24017, Remote: None
    ID: 2 (0xa79ce90), Ctrl Trees : 0/0/0, Ctrl ID: 0 (0x0), IR Ctrl ID: 0 (0x0), Ctrl HLI:
 0x00000
  P2MP Def MDT ID: 0 (0x0), added: 0, HLI: 0x00000, Cfg: 0/0


Router# show mpls mldp database root 10.0.0.21 opaquetype global-id 5

mLDP database
LSM-ID: 0x00020  Type: P2MP  Uptime: 2d17h
  FEC Root           : 10.0.0.21
  IGP Algorithm      : Flex-Algo 129
   Opaque decoded     : [global-id 5]
  Upstream neighbor(s) :
```

```
    10.0.0.4:0 [Active] Uptime: 1d22h
    Local Label (D) : 24040
  Downstream  client(s):
    PIM MDT            Uptime: 2d17h
      Egress intf     : LmdtWHITE
      Table ID        : IPv4: 0xe0000012 IPv6: 0xe0800012
      RPF ID          : 15
      RD              : 257:18154475
```

```
Router# show pim vrf vpn1 mdt cache
Core Source      Cust (Source, Group)              Core Data        Expires
5.15.15.15       (100.0.1.1, 233.1.1.1)            [global-id 17, Flex-Algo 129]  never

5.15.15.15       (100.0.1.1, 233.1.1.2)            [global-id 18, Flex-Algo 129]  never

5.15.15.15       (100.0.1.1, 233.1.1.3)            [global-id 19, Flex-Algo 129]  never

5.15.15.15       (100.0.1.1, 233.1.1.4)            [global-id 20, Flex-Algo 129]  never

5.15.15.15       (100.0.1.1, 233.1.1.5)            [global-id 21, Flex-Algo 129]  never
```

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 Inband mLDP Profiles on Edge Routers

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

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

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

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

mpls ldp
 mldp
```

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

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

route-policy rpf-global
  set core-tree mldp-inband
end-policy

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

Verification of MLDP Configuration on Edge Routers

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

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

```
RP/0/RP0/CPU0:Head# show mpls mldp neighbors
mLDP neighbor database
MLDP peer ID      : 2.2.2.2:0, uptime 07:47:59 Up,
  Capabilities     : GR, Typed Wildcard FEC, P2MP, MP2MP
  Target Adj       : No
  Upstream count   : 1
  Branch count     : 1
  LDP GR           : Enabled
                   : Instance: 1
  Label map timer  : never
```

```
  Policy filter in :
  Path count       : 1
  Path(s)          : 12.1.1.2          TenGigE0/0/1/0/3.2000 LDP
  Adj list         : 12.1.1.2          TenGigE0/0/1/0/3.2000
  Peer addr list   : 2.25.32.2
                   : 2.2.2.2
                   : 11.1.1.1
                   : 12.1.1.2
                   : 13.10.1.1
```

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

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

LSP-ID: 0x00001 Paths: 7 Flags:
0x00001 P2MP   5.5.5.5 [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  : EDM Loc init iox success
May 31 12:08:39.475 MLDP GLO 0/RP0/CPU0 t6746 GEN  : LMRIB init iox success
May 31 12:08:39.475 MLDP GLO 0/RP0/CPU0 t18944 MRIB : MRIB connection established
May 31 12:08:39.475 MLDP GLO 0/RP0/CPU0 t6746 GEN  : Interface manager init iox success
May 31 12:08:39.475 MLDP GLO 0/RP0/CPU0 t6746 GEN  : Async init iox success
May 31 12:08:39.475 MLDP GLO 0/RP0/CPU0 t6746 GEN  : Boolean init iox success
May 31 12:08:39.475 MLDP GLO 0/RP0/CPU0 t6746 GEN  : Timers init iox success
May 31 12:08:39.479 MLDP GLO 0/RP0/CPU0 t6746 GEN  : RUMP init iox success
May 31 12:08:39.479 MLDP GLO 0/RP0/CPU0 t6746 GEN  : Chunks init iox success
May 31 12:08:39.509 MLDP ERR 0/RP0/CPU0 t6746 RIB  : RIB not ready
May 31 12:08:39.509 MLDP ERR 0/RP0/CPU0 t6746 RIB  : RIB not ready
May 31 12:08:39.512 MLDP GLO 0/RP0/CPU0 t6746 GEN  : mldp_ens_event_ctx_chunk is NULL
May 31 12:08:39.512 MLDP GLO 0/RP0/CPU0 t6746 GEN  : Context Table init iox success
May 31 12:08:39.512 MLDP GLO 0/RP0/CPU0 t6746 GEN  : mldp_rib_main_evm 0x563de8fd23e8
allocated
May 31 12:08:39.512 MLDP GLO 0/RP0/CPU0 t6746 GEN  : RIB Thread EVM init rib success
May 31 12:08:39.512 MLDP GLO 0/RP0/CPU0 t6746 GEN  : RIB Thread Chunk init rib success
May 31 12:08:39.512 MLDP GLO 0/RP0/CPU0 t6746 GEN  : RIB Thread queue init rib success
May 31 12:08:39.512 MLDP GLO 0/RP0/CPU0 t6746 RIB  : Bound to RIB, fd: 354
```

Overriding VRFs in IGMP Interfaces

Table 10: Feature History Table

Feature Name	Release Information	Feature Description
Support for IGMP VRF Override in Multicast Routers	Release 7.5.1	Using this feature, you can configure a multicast router interface to override the configuration specified in the local VRF table. When an IGMP client sends a join message to the multicast router, it performs a Reverse-path Forwarding (RPF) lookup for the IGMP join in the local VRF table. If the local VRF table does not have the information, the feature extends the lookup to the default (global) VRF table. This ensures that the interface in a specific VRF table is part of the outgoing list of interfaces in the global routing table for a multicast route.

All unicast traffic on the user-to-network interfaces of next-generation aggregation or core networks must be mapped to a specific VRF. They must then be mapped to an MPLS VPN on the network-to-network side. This requires the configuration of a physical interface in this specific VRF.

This feature allows mapping of IGMP packets entering through a user-to-user interface to the multicast routes in the global multicast routing table. This ensures that the interface in a specific VRF can be part of the outgoing list of interfaces in the table for a multicast route.

IGMP packets entering through a non-default VRF interface in the default (global) VRF are processed, with IGMP later distributing the interface-related multicast state (route/interface) to MRIB. This occurs through the default VRF rather than through the VRF to which the interface belongs. MRIB, PIM, MSDP, and MFIB then process the multicast state for this interface through the default VRF.

When an IGMP join for a specific (S, G) is received on the configured interface, IGMP stores this information in its VRF-specific databases. But, when sending an update to MRIB, IGMP sends this route through the default VRF. MRIB then programs this (S, G) along with this interface as an OLIST member in the default multicast routing table.

Similarly, when PIM requests information about IGMP routes from MRIB, MRIB sends this update to PIM in the context of the default VRF.

This feature specifically supports:

- Mapping of IGMP requests on an interface in a non-default VRF to the default VRF multicast routing table.
- Enabling and disabling of VRF override functionality at run time.

- Routing policy configuration at the global (default) VRF level, because routing policy configuration cannot be done at the granularity of an individual interface.
- Enablement and disablement of an IGMP VRF override on all Layer- 3 and Layer- 2 interface types, including physical Ethernet, VLAN sub-interface, bundles and VLANs over bundles.
- The same scale of multicast routes and OLIST interfaces currently supported by the platform even when VRF override functionality is operational.

Configuring IGMP VRF Override

This process consists of the following tasks:

Specifying VRF definition

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**vrf** *vrf-name* **Example:** `RP/0/RP0/CPU0:router(config)# vrf name1`	Enters the VRF configuration sub mode.
Step 3	**address-family ipv4 unicast** **Example:** `RP/0/RP0/CPU0:router(config-vrf)# address-family ipv4 unicast`	AFI configuration for IPv4. This is supported on unicast topologies only.
Step 4	**import route-target 1:1** **Example:** `RP/0/RP0/CPU0:router(config-vrf-af)# import route-target 1:1`	Enables VRF import.
Step 5	**export route-target 1:1** **Example:** `RP/0/RP0/CPU0:router(config-vrf-af)# export route-target 1:1`	Enables VRF export.
Step 6	**commit**	

Enabling Multicast Routing on default and non-default VRFs

This task enables multicast routing and forwarding on all new and existing interfaces. For the VRF override feature, multicast routing needs to be enabled on both, the default and the non-default VRFs.

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**multicast-routing vrf** [*vrf-name* | *default]* **Example:** `RP/0/RP0/CPU0:router(config)# multicast-routing vrf green`	Enters multicast configuration mode for the specified VRF. Note that the default configuration mode for multicast routing is default vrf (if the non-default VRF name is not specified).
Step 3	**interface** {*type interface-path-id* | **all**} **enable** **Example:** `RP/0/RP0/CPU0:router(config-mcast-green)# interface all enable`	Enables multicast routing and forwarding on one or on all new and existing interfaces.
Step 4	**commit**	

Configuring an Interface for a Non-default VRF Instance

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface tengige 0/1/0/0`	Enters PIM address-family IPv4 submode.
Step 3	**vrf** *vrf-name* **Example:** `RP/0/RP0/CPU0:router(config-if)# vrf name1`	Sets the VRF for the interface.

	Command or Action	Purpose
Step 4	**ipv4 address** *address mask* **Example:** `RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.1.1.1 255.0.0.0`	Sets the IPv4 address for the interface.
Step 5	**commit**	

Configuring route-policy

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**route-policy** *policy-name* **Example:** `RP/0/RP0/CPU0:router(config)# route-policy policy1`	Defines a route policy.
Step 3	**set rpf-topology vrf default** **Example:** `RP/0/RP0/CPU0:router(config-rpl)# set rpf-topology vrf default`	Sets the PIM RPF topology attributes for the default VRF.
Step 4	**end-policy** **Example:** `RP/0/RP0/CPU0:router(config-rpl)# end-policy`	Ends the route-policy definition configuration.
Step 5	**commit**	

Associating a route policy to PIM configuration for the VRF receiving IGMP reports

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**router pim vrf** *vrf-name* **address-family ipv4**	Enters PIM address-family IPv4 submode.
Step 3	**rpf-topology route-policy** *policy-name* **Example:** `RP/0/RP0/CPU0:router(config-rpl)# rpf-topology route-policy policy1`	Associates a previously defined route-policy with the non-default VRF that receives the IGMP reports.
Step 4	**commit**	

Configure MVPN using Draft-Rosen (Profile 0)

Table 11: Feature History Table

Feature Name	Release Information	Feature Description
Draft-Rosen Multicast VPN (Profile 0) in PIM Sparse Mode (SM)	Release 7.10.1	Draft-Rosen Multicast VPN (Profile 0) is now supported in PIM sparse mode (PIM-SM) between the PE routers that are running in VRF mode. PIM SM provides precise control in cases of large multicast traffic when there is less bandwidth available. This control is possible because it uses a temporary Rendezvous Point (RP) router to connect the multicast traffic source to the next hop router. Prior to this release, Profile 0 was supported only in PIM Source Specific Multicast (SSM) mode.

Feature Name	Release Information	Feature Description
Draft-Rosen Multicast VPN (Profile 0)	Release 7.4.1	Draft-Rosen (profile 0) is a widely used MVPN model and uses GRE tunnels to securely transmit multicast traffic between the PE routers. It also enables ease of deployment by using the Protocol-Independent Multicast (PIM) protocol between edge routers (PE) and hosts (CE), and between PE routers that are running in VRF mode.

Draft-Rosen Multicast VPN (Profile 0) uses Generic Routing Encapsulation (GRE) as an overlay protocol. All multicast packets are encapsulated inside GRE. Profile 0 has PIM as the multicast routing protocol between the edge routers (PE) and hosts (CE), and between the PE routers in the VRF mode. The PE routers directly connect using a Default Multicast Distribution Tree (MDT) formed between the PE routers. The PE routers connect to each other as PIM neighbors across the Default MDT.

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

Benefits

- Profile 0 is a widely used model and fairly easy to deploy as Profile 0 uses the native multicast in the core and does not require any additional configuration on customers routers and in the core.

Restriction

- PIM SM is not supported in core and under VRF. Only PIM SSM is supported.
- IPv6 is not supported on both VRF and core.
- BVI is not supported in the core.
- In-Service Software Upgrade (ISSU) is not supported for MVPN GRE and L2 multicast.
- Auto-RP is not supported.
- If there is an IPv4 Unicast GRE tunnel configured in your network, the Maximum Transmission Unit (MTU) size of the configured Unicast GRE tunnel impacts the MTU of the Profile-0 MDT multicast. Ensure that the Profile-0 MDT multicast packet size does not exceed the MTU value of the IPv4 unicast GRE tunnel. If the multicast packet size value exceeds the MTU value of the tunnel, then the packet is dropped.
- Use the **immediate-switch** keyword only for data MDT switchover. Switchover from the default MDT to the data MDT is not supported based on the threshold.

Configuration Example

Perform the following steps to configure Profile 0 on the PE devices:

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

Router(config)# multicast-routing
Router(config-mcast)# vrf vpn101
Router(config-mcast-vpn101)# address-family ipv4
Router(config-mcast-vpn101-ipv4)#  mdt source Loopback0
Router(config-mcast-vpn101-ipv4)#   mdt default ipv4 232.100.0.1
Router(config-mcast-vpn101-ipv4)#   mdt data 232.101.0.1/24
Router(config-mcast-vpn101-ipv4)#   interface all enable

Router(config)# router pim
Router(config-pim)# address-family ipv4
Router(config-pim-default-ipv4)# vrf vpn101
Router(config-pim-vpn101)#  address-family ipv4
Router(config-pim-vpn101-ipv4)#   rpf topology route-policy rosen-gre
Router(config-pim-vpn101-ipv4)#   exit
Router(config-pim-vpn101-ipv4)# commit
```

Running Configuration

```
/*Head Configuration*/ :

hostname PE1
logging console disable
vrf vpn101
 address-family ipv4 unicast
  import route-target
   1:1
  !
  export route-target
   1:1
  !
 !
 address-family ipv6 unicast
  import route-target
   1:1
  !
  export route-target
   1:1
  !
 !
!
line console
 exec-timeout 0 0
 absolute-timeout 0
 session-timeout 0
!
line default
 exec-timeout 0 0
 absolute-timeout 0
 session-timeout 0
!
call-home
 service active
 contact smart-licensing
```

```
 profile CiscoTAC-1
  active
  destination transport-method email disable
  destination transport-method http
 !
!
interface Bundle-Ether1
 load-interval 30
 l2transport
 !
!
interface Loopback0
 ipv4 address 4.4.4.4 255.255.255.255
 ipv6 address 4::4/124
!
interface Loopback1
 vrf vpn101
 ipv4 address 4.1.1.1 255.255.255.255
 ipv6 address 4::1:1/124
!
interface MgmtEth0/RP0/CPU0/0
 ipv4 address 7.1.10.55 255.255.0.0
!
interface TenGigE0/0/0/4
 ipv4 address 10.2.0.1 255.255.255.0
 ipv6 address 10::2:1/124
 load-interval 30
!
interface TenGigE0/0/0/18
 vrf vpn101
 ipv4 address 2.0.0.1 255.255.0.0
 ipv6 address 2::1/124
 load-interval 30
!
!
route-policy PASS
  pass
end-policy
!
route-policy rosen-gre
  set core-tree mldp-partitioned-p2mp
end-policy
!
route-policy pim-mdt-default
  set core-tree pim-default
end-policy
!
router static
 address-family ipv4 unicast
  0.0.0.0/0 7.1.0.1
 !
!
router ospf core
 router-id 4.4.4.4
 mpls ldp auto-config
 area 0
  interface Loopback0
  !
  interface TenGigE0/0/0/4
  !
 !
!
router bgp 100
 bgp router-id 4.4.4.4
```

```
 address-family ipv4 unicast
  redistribute connected
  allocate-label all
 !
 address-family vpnv4 unicast
 !
 address-family ipv6 unicast
  redistribute connected
  allocate-label all
 !
 address-family vpnv6 unicast
 !
 address-family ipv4 mdt
 !
 address-family ipv4 mvpn
 !
 address-family ipv6 mvpn
 !
 neighbor 21.21.21.21
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
  !
  address-family vpnv4 unicast
  !
  address-family ipv6 unicast
  !
  address-family vpnv6 unicast
  !
  address-family ipv4 mdt
  !
  address-family ipv4 mvpn
  !
  address-family ipv6 mvpn
  !
 !
 vrf vpn101
  rd 1:1
  address-family ipv4 unicast
   route-target download
   redistribute connected
  !
  address-family ipv6 unicast
   route-target download
   redistribute connected
  !
 !
!
mpls ldp
 mldp
  address-family ipv4
  !
 !
 router-id 4.4.4.4
 interface TenGigE0/0/0/4
 !
!
multicast-routing
 address-family ipv4
  interface all enable
 !
 address-family ipv6
  interface all enable
 !
```

```
 vrf vpn101
  address-family ipv4
   mdt source Loopback0
   rate-per-route
   interface all enable
   mdt default ipv4 232.1.0.1
   mdt data 232.100.1.1/24 immediate-switch
  !
  address-family ipv6
   mdt source Loopback0
   rate-per-route
   interface all enable
   mdt default ipv4 232.2.0.1
   mdt data 232.200.1.1/24 immediate-switch
  !
 !
!
lldp
!
router pim
 vrf vpn101
  address-family ipv4
   rpf topology route-policy pim-mdt-default
   hello-interval 1
   rp-address 21.1.1.1
  !
  address-family ipv6
   rpf topology route-policy pim-mdt-default
   hello-interval 1
  !
 !
!
mld snooping profile mldsn
 system-ip-address fe80::1 link-local
 internal-querier
!
igmp snooping profile igmpsn
 system-ip-address 4.4.4.4
 internal-querier
!
igmp snooping profile snjoin
 static group 232.1.1.1 source 40.0.0.2
!
end


/*Tail Configuration*/ :

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

```
  export route-target
   1:1
  !
 !
!
line console
 exec-timeout 0 0
 absolute-timeout 0
 session-timeout 0
!
line default
 exec-timeout 0 0
 absolute-timeout 0
 session-timeout 0
!
call-home
 service active
 contact smart-licensing
 profile CiscoTAC-1
  active
  destination transport-method email disable
  destination transport-method http
 !
!
interface Loopback0
 ipv4 address 21.21.21.21 255.255.255.255
 ipv6 address 21::21/124
!
interface Loopback1
 vrf vpn101
 ipv4 address 21.1.1.1 255.255.255.255
 ipv6 address 21::1:1/124
!
interface MgmtEth0/RP0/CPU0/0
 ipv4 address 7.1.10.57 255.255.0.0
 shutdown
!
interface TenGigE0/0/0/4
 ipv4 address 10.2.0.2 255.255.255.0
 ipv6 address 10::1:1/124
 load-interval 30
!
interface TenGigE0/0/0/6
 vrf vpn101
 ipv4 address 3.0.0.1 255.255.0.0
 ipv6 address 3::1/124
 load-interval 30
!
!
route-policy PASS
  pass
end-policy
!
route-policy rosen-gre
  set core-tree mldp-partitioned-p2mp
end-policy
!
route-policy pim-mdt-default
  set core-tree pim-default
end-policy
!
router static
 address-family ipv4 unicast
  0.0.0.0/0 7.1.0.1
```

```
 !
!
router ospf core
 router-id 21.21.21.21
 mpls ldp auto-config
 area 0
  interface Loopback0
  !
  interface TenGigE0/0/0/4
  !
 !
!
router bgp 100
 bgp router-id 21.21.21.21
 bgp graceful-restart
 address-family ipv4 unicast
  redistribute connected
 !
 address-family vpnv4 unicast
 !
 address-family ipv6 unicast
  redistribute connected
 !
 address-family vpnv6 unicast
 !
 address-family ipv4 mdt
 !
 address-family ipv4 mvpn
 !
 address-family ipv6 mvpn
 !
 neighbor 4.4.4.4
  remote-as 100
  update-source Loopback0
  address-family ipv4 unicast
  !
  address-family vpnv4 unicast
  !
  address-family ipv6 unicast
  !
  address-family vpnv6 unicast
  !
  address-family ipv4 mdt
  !
  address-family ipv4 mvpn
  !
  address-family ipv6 mvpn
  !
 !
 vrf vpn101
  rd 1:1
  address-family ipv4 unicast
   route-target download
   redistribute connected
  !
  address-family ipv6 unicast
   route-target download
   redistribute connected
  !
 !
!
multicast-routing
 address-family ipv4
  interface all enable
```

```
 !
 address-family ipv6
  interface all enable
 !
 vrf vpn101
  address-family ipv4
   mdt source Loopback0
   rate-per-route
   interface all enable
   mdt default ipv4 232.1.0.1
   mdt data 232.100.1.1/24 immediate-switch
  !
  address-family ipv6
   mdt source Loopback0
   rate-per-route
   interface all enable
   mdt default ipv4 232.2.0.1
   mdt data 232.200.1.1/24 immediate-switch
  !
 !
!
lldp
!
router pim
 vrf vpn101
  address-family ipv4
   rpf topology route-policy pim-mdt-default
   hello-interval 1
   rp-address 21.1.1.1
  !
  address-family ipv6
   rpf topology route-policy pim-mdt-default
   hello-interval 1
  !
 !
!
router igmp
 vrf vpn101
  interface  interface TenGigE0/0/0/6
   static-group 232.1.1.1 3.0.0.2
  !
 !
!
end
```

Verification

```
Router# show pim vrf vpn101 context
Wed Aug  9 14:56:28.382 UTC

PIM context information for VRF vpn101 (0x55845d7772b0)

VRF ID: 0x60000002
Table ID: 0xe0000002
Remote Table ID: 0xe0800002
MDT Default Group : 232.1.0.1
MDT Source : (4.4.4.4, Loopback0) Per-VRF
MDT Immediate Switch Not Configured
MDT handle: 0x2000806c(mdtvpn101)
Context Active, ITAL Active
Routing Enabled
Registered with MRIB
Owner of MDT Interface
Raw socket req: T, act: T, LPTS filter req: T, act: T
```

```
UDP socket req: T, act: T, UDP vbind req: T, act: T
Reg Inj socket req: T, act: T, Reg Inj LPTS filter req: T, act: T
Mhost Default Interface : TenGigE0/0/0/18 (publish pending: F)
Remote MDT Default Group : 232.2.0.1
Backup MLC virtual interface: Null
Neighbor-filter: -
MDT Neighbor-filter: -

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

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

(*,224.0.0.0/4) Ver: 0x9b0d RPF nbr: 21.1.1.1 Flags: L C RPF P, FGID: 11903, Statistics
enabled: 0x0, Tunnel RIF: 0xffffffff, Tunnel LIF: 0xffffffff
  Up: 00:03:33
  Outgoing Interface List
    Decapstunnel1 Flags: NS DI, Up: 00:03:28

(*,224.0.0.0/24) Ver: 0xec6c Flags: D P, FGID: 11901, Statistics enabled: 0x0, Tunnel RIF:
 0xffffffff, Tunnel LIF: 0xffffffff
  Up: 00:03:33

(*,224.0.1.39) Ver: 0xe7dc Flags: S P, FGID: 11899, Statistics enabled: 0x0, Tunnel RIF:
0xffffffff, Tunnel LIF: 0xffffffff
  Up: 00:03:33

(*,224.0.1.40) Ver: 0xf5fb Flags: S P, FGID: 11900, Statistics enabled: 0x0, Tunnel RIF:
0xffffffff, Tunnel LIF: 0xffffffff
  Up: 00:03:33
  Outgoing Interface List
    TenGigE0/0/0/6 Flags: II LI, Up: 00:03:33

(*,232.0.0.0/8) Ver: 0x96d1 Flags: D P, FGID: 11902, Statistics enabled: 0x0, Tunnel RIF:
0xffffffff, Tunnel LIF: 0xffffffff
  Up: 00:03:33

(2.0.0.2,232.1.1.1) Ver: 0x2c3f RPF nbr: 4.4.4.4 Flags: RPF RPFID, FGID: 11907, Statistics
 enabled: 0x0, Tunnel RIF: 0x26, Tunnel LIF: 0xa000, Flags2: 0x2
  Up: 00:00:22
  RPF-ID: 1, Encap-ID: 0
  Incoming Interface List
    mdtvpn101 Flags: A MI, Up: 00:00:22
  Outgoing Interface List
    TenGigE0/0/0/6 Flags: F NS LI, Up: 00:00:22

Router# show pim vrf vpn101 context
Wed Aug  9 15:00:03.092 UTC
```

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

(*,224.0.0.0/24) Ver: 0x7f6e Flags: D P, FGID: 9235, Statistics enabled: 0x0, Tunnel RIF:
0xffffffff, Tunnel LIF: 0xffffffff
  Up: 00:06:54

(*,224.0.1.39) Ver: 0x7af2 Flags: S P, FGID: 9233, Statistics enabled: 0x0, Tunnel RIF:
0xffffffff, Tunnel LIF: 0xffffffff
  Up: 00:06:54

(*,224.0.1.40) Ver: 0x9f9b Flags: S P, FGID: 9234, Statistics enabled: 0x0, Tunnel RIF:
0xffffffff, Tunnel LIF: 0xffffffff
  Up: 00:06:54
  Outgoing Interface List
    TenGigE0/0/0/4 Flags: II LI, Up: 00:06:54

(*,232.0.0.0/8) Ver: 0x517d Flags: D P, FGID: 9236, Statistics enabled: 0x0, Tunnel RIF:
0xffffffff, Tunnel LIF: 0xffffffff
  Up: 00:06:54

(4.4.4.4,232.1.0.1) Ver: 0xc19e RPF nbr: 4.4.4.4 Flags: RPF ME MH, FGID: 9242, Statistics
enabled: 0x0, Tunnel RIF: 0x35, Tunnel LIF: 0xc000, Flags2: 0x2
  MVPN TID: 0xe0000002
  MVPN Remote TID: 0x0
  MVPN Payload: IPv4
  MDT IFH: 0x2000806c
  Up: 00:06:49
  RPF-ID: 1, Encap-ID: 0
  Incoming Interface List
    Loopback0 Flags: F A, Up: 00:06:49
  Outgoing Interface List
    Loopback0 Flags: F A, Up: 00:06:49
    TenGigE0/0/0/4 Flags: F NS, Up: 00:04:13

(21.21.21.21,232.1.0.1) Ver: 0xa354 RPF nbr: 10.2.0.2 Flags: RPF MD MH CD, FGID: 9244,
Statistics enabled: 0x0, Tunnel RIF: 0x35, Tunnel LIF: 0xc000, Flags2: 0x2
  MVPN TID: 0xe0000002
  MVPN Remote TID: 0x0
  MVPN Payload: IPv4
  MDT IFH: 0x2000806c
  Up: 00:04:13
  RPF-ID: 1, Encap-ID: 0
  Incoming Interface List
    TenGigE0/0/0/4 Flags: A, Up: 00:04:13
  Outgoing Interface List
    Loopback0 Flags: F NS, Up: 00:04:13

(4.4.4.4,232.2.0.1) Ver: 0xbab2 RPF nbr: 4.4.4.4 Flags: RPF ME MH, FGID: 9243, Statistics
enabled: 0x0, Tunnel RIF: 0x35, Tunnel LIF: 0xc000, Flags2: 0x2
```

```
  MVPN TID: 0x0
  MVPN Remote TID: 0xe0800002
  MVPN Payload: IPv6
  MDT IFH: 0x2000806c
  Up: 00:06:49
  RPF-ID: 1, Encap-ID: 0
  Incoming Interface List
    Loopback0 Flags: F A, Up: 00:06:49
  Outgoing Interface List
    Loopback0 Flags: F A, Up: 00:06:49

(4.4.4.4,232.100.1.0) Ver: 0x86b RPF nbr: 4.4.4.4 Flags: RPF ME MH, FGID: 9246, Statistics
 enabled: 0x0, Tunnel RIF: 0x35, Tunnel LIF: 0xc000, Flags2: 0x2
  MVPN TID: 0xe0000002
  MVPN Remote TID: 0x0
  MVPN Payload: IPv4
  MDT IFH: 0x2000806c
  Up: 00:01:17
  RPF-ID: 1, Encap-ID: 0
  Incoming Interface List
    Loopback0 Flags: F A, Up: 00:01:17
  Outgoing Interface List
    Loopback0 Flags: F A, Up: 00:01:17
    TenGigE0/0/0/4 Flags: F NS, Up: 00:01:17

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

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

(*,224.0.0.0/24) Ver: 0x7f6e Flags: D P, FGID: 9235, Statistics enabled: 0x0, Tunnel RIF:
0xffffffff, Tunnel LIF: 0xffffffff
  Up: 00:06:54

(*,224.0.1.39) Ver: 0x7af2 Flags: S P, FGID: 9233, Statistics enabled: 0x0, Tunnel RIF:
0xffffffff, Tunnel LIF: 0xffffffff
  Up: 00:06:54

(*,224.0.1.40) Ver: 0x9f9b Flags: S P, FGID: 9234, Statistics enabled: 0x0, Tunnel RIF:
0xffffffff, Tunnel LIF: 0xffffffff
  Up: 00:06:54
  Outgoing Interface List
    TenGigE0/0/0/4 Flags: II LI, Up: 00:06:54

(*,232.0.0.0/8) Ver: 0x517d Flags: D P, FGID: 9236, Statistics enabled: 0x0, Tunnel RIF:
0xffffffff, Tunnel LIF: 0xffffffff
  Up: 00:06:54

(4.4.4.4,232.1.0.1) Ver: 0xc19e RPF nbr: 4.4.4.4 Flags: RPF ME MH, FGID: 9242, Statistics
```

```
enabled: 0x0, Tunnel RIF: 0x35, Tunnel LIF: 0xc000, Flags2: 0x2
  MVPN TID: 0xe0000002
  MVPN Remote TID: 0x0
  MVPN Payload: IPv4
  MDT IFH: 0x2000806c
  Up: 00:06:49
  RPF-ID: 1, Encap-ID: 0
  Incoming Interface List
    Loopback0 Flags: F A, Up: 00:06:49
  Outgoing Interface List
    Loopback0 Flags: F A, Up: 00:06:49
    TenGigE0/0/0/4 Flags: F NS, Up: 00:04:13

(21.21.21.21,232.1.0.1) Ver: 0xa354 RPF nbr: 10.2.0.2 Flags: RPF MD MH CD, FGID: 9244,
Statistics enabled: 0x0, Tunnel RIF: 0x35, Tunnel LIF: 0xc000, Flags2: 0x2
  MVPN TID: 0xe0000002
  MVPN Remote TID: 0x0
  MVPN Payload: IPv4
  MDT IFH: 0x2000806c
  Up: 00:04:13
  RPF-ID: 1, Encap-ID: 0
  Incoming Interface List
    TenGigE0/0/0/4 Flags: A, Up: 00:04:13
  Outgoing Interface List
    Loopback0 Flags: F NS, Up: 00:04:13

(4.4.4.4,232.2.0.1) Ver: 0xbab2 RPF nbr: 4.4.4.4 Flags: RPF ME MH, FGID: 9243, Statistics
enabled: 0x0, Tunnel RIF: 0x35, Tunnel LIF: 0xc000, Flags2: 0x2
  MVPN TID: 0x0
  MVPN Remote TID: 0xe0800002
  MVPN Payload: IPv6
  MDT IFH: 0x2000806c
  Up: 00:06:49
  RPF-ID: 1, Encap-ID: 0
  Incoming Interface List
    Loopback0 Flags: F A, Up: 00:06:49
  Outgoing Interface List
    Loopback0 Flags: F A, Up: 00:06:49

(4.4.4.4,232.100.1.0) Ver: 0x86b RPF nbr: 4.4.4.4 Flags: RPF ME MH, FGID: 9246, Statistics
 enabled: 0x0, Tunnel RIF: 0x35, Tunnel LIF: 0xc000, Flags2: 0x2
  MVPN TID: 0xe0000002
  MVPN Remote TID: 0x0
  MVPN Payload: IPv4
  MDT IFH: 0x2000806c
  Up: 00:01:17
  RPF-ID: 1, Encap-ID: 0
  Incoming Interface List
    Loopback0 Flags: F A, Up: 00:01:17
  Outgoing Interface List
    Loopback0 Flags: F A, Up: 00:01:17
    TenGigE0/0/0/4 Flags: F NS, Up: 00:01:17
```

Configure mVPN using RSVP-TE P2MP (Profile 22)

Table 12: Feature History Table

Feature Name	Release Information	
mVPN using RSVP-TE P2MP (Profile 22)	Release 7.5.1	This feature uses RSVP-TE to establish MPLS transport LSPs through traffic engineering and securely transmits multicast traffic between the PE routers in a MPLS network.

Profile 22 is a mVPN profile used for multicast traffic engineering using RSVP-TE P2MP. RSVP-TE is used to establish MPLS transport LSPs when there are traffic engineering requirements. The Default MDT consists of a full mesh of P2MP-TE tree. The P2MP tunnels are P2MP-TE auto-tunnels. PIM is used for C-Multicast signaling.

These are the characteristics of this profile:

- Dynamic P2MP-TE tunnels with BGP C-Multicast routing.
- All UMH options are supported.
- Default and Data MDT are supported.
- Customer traffic can be SSM.

Benefits

- More flexible designs per VPN.
- Proven functionality (Default-MDT and Data-MDT).
- Supports mLDP and P2MP-TE core tree protocols.
- Traffic engineering for multicast along with mVPN.

Configuration

Perform the following tasks on the edge routers (PE devices):

- Configure VRF
- Enable PIM with Profile 22

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

```
router pim
 vrf one
  address-family ipv4
   rpf topology route-policy rpf-vrf-one
   mdt c-multicast-routing bgp
   interface GigabitEthernet0/0/0/1.100
    enable

route-policy rpf-vrf-one
  set core-tree p2mp-te-default
end-policy

multicast-routing
 vrf one
  address-family ipv4
   mdt source Loopback0
   mdt default p2mp-te
   rate-per-route
   interface all enable
   mdt data p2mp-te 100
   bgp auto-discovery p2mp-te
   !
   accounting per-prefix

ipv4 unnumbered mpls traffic-eng Loopback0

mpls traffic-eng
 interface GigabitEthernet0/0/0/0
 !
 interface GigabitEthernet0/0/0/2
 !
  auto-tunnel p2mp
  tunnel-id min 1000 max 2000
```

Verification

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

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

(80.0.0.11,232.1.1.1) Ver: 0xb009 RPF nbr: 80.0.0.11 Flags: RPF EID,
 PD: Slotmask: 0x0
     MGID: 592
  Up: 00:07:14
  RPF-ID: 0, Encap-ID: 262145
  Incoming Interface List
    TenGigE0/1/0/1.2 Flags: A, Up: 00:07:14
```

```
  Outgoing Interface List
    Tmdtp22/ssm/v4/vrf2 Flags: F TMI, Up: 00:07:14, Head LSM-ID: 0x4000360
```

Restrictions for MVPN Profiles

The following restriction applies to the configuration of MVPN profile:

- A router being Route Reflector (RR) and Provider Edge (PE) at that same time for BGP mVPN implementation is not supported, a type 7 and type 6 IPv4 mVPN route is not advertised by a RR, which is also a PE router, if the PE router has the VRF locally configured and when there is a local receiver.

  Use full mesh for iBGP mVPN address-family or elect any core (P) router to be the RR.

Configuration Examples for MVPN Profiles

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

Configuration Examples for Inband mLDP Profiles

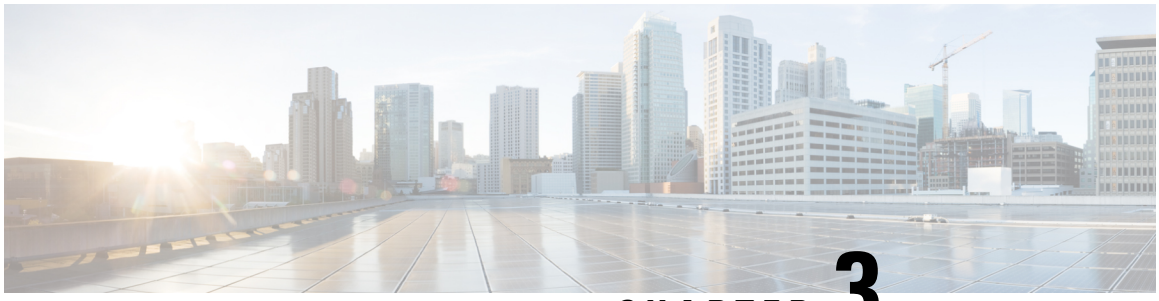
Profile-6: VRF Inband mLDP

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

Profile-7: Global Inband mLDP

```
multicast-routing
 address-family ipv4
   mdt source Loopback0
```

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



CHAPTER 3

Implementing Layer 2 Multicast

Implementing 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.

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

Prerequisites for IGMP Snooping

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

Supported Features and Restrictions for IGMP Snooping

- EVPN dual-homed Active Active (AA) IGMP State Sync using IGMP snooping profile is supported.
- BVI under bridge domain is supported.
- IGMP snooping is supported only under L2VPN bridge domains.
- Explicit host tracking (an IGMPv3 snooping feature) is not supported.
- IGMPv1 is not supported.
- IGMP snooping with VPLS on bridge domain is not supported.
- IGMP snooping over access and core Pseudo-wire is not supported.
- ISSU is not supported on Layer 2 Multicast.
- IGMPv3-exclude is not supported in EVPN multi-homing or proxy scenarios.
- For EVPN AA, IGMPv2 and IGMPv3 joins for same groups are not supported.
- **router-alert-check disable** configuration command is not supported.
- EVPN dual-home source AA is supported only on NCS 540 series router.
- EVPN configuration must have the **control-word-disable** configuration.
- PIM control packets (join and hello) processing is not supported when snooping is enabled, so a multicast router selection based on PIM packets won't occur.
- In an EVPN dual-home AA scenario:
    - If the multicast source and receiver are in the same bridge domain (BD), the receiver might receive permanent traffic duplication.
    - In an EVPN dual-home receiver AA scenario, transient traffic duplication is expected when the DH node role changes from DF to nDF and vice versa.
    - Source=ESI1=BE-X.A, Receiver=ESI1=BE-X.B under the same BD is not supported (where X.A and X.B represent two AC ports for the bundle interface BE).
    - Source=ESI1=BE-X.A (for NCS 5700 line cards), Receiver=ESI2=BE-Y.A (for NCS 5500 line cards) under the same BD is not supported (where X.A and Y.A represent two AC ports for the bundle interface BE).

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

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

IGMP snooping for bridge domains without Bridged Virtual Interface (BVI) is supported with the following design consideration: You must configure the multicast-source ipv4 command in the source switch where bridge domain and IGMP snooping are enabled.

Information About IGMP Snooping

IGMP Snooping Overview

Description of Basic Functions

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

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

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

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

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 540 Series Routers randomly nominates a single port from the bundle to carry the multicast traffic.

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

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

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

Configuration Example:

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

Multicast Router Port

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

Multicast 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 13: Multicast Traffic Handling for an IGMPv2 Querier, on page 93 describes traffic handling for an IGMPv2 querier. Table 14: Multicast Traffic Handling for an IGMPv3 Querier, on page 93 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 13: Multicast Traffic Handling for an IGMPv2 Querier

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

Table 14: Multicast Traffic Handling for an IGMPv3 Querier

Traffic Type	Received on Host Ports
IP multicast source traffic	Forwards to all mrouter ports and to host ports that indicate interest.
IGMP general queries	—
IGMP group-specific queries	—
IGMPv2 joins	Handles as IGMPv3 IS_EX{} reports.

Traffic Type	Received on Host Ports
IGMPv3 reports	• If proxy reporting is enabled—For state changes or source-list changes, generates a state change report on all mrouter ports. • If proxy reporting is disabled—Forwards on all mrouter ports.
IGMPv2 leaves	Handles as IGMPv3 IS_IN{} reports.

IGMP Snooping Configuration Profiles

To enable IGMP snooping on a bridge domain, you must attach a profile to the bridge domain. The minimum configuration is an empty profile if BVI is configured. An empty profile enables the default configuration options and settings for IGMP snooping, as listed in the Default IGMP Snooping Configuration Settings, on page 96.

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

Configuration Example:

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

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

- Any IGMP Snooping profile attached to a bridge domain, even an empty profile, enables IGMP snooping. To disable IGMP snooping, detach the profile from the bridge domain.
- An empty profile configures IGMP snooping on the bridge domain and all ports under the bridge using default configuration settings.
- A bridge domain can have only one IGMP snooping profile attached to it (at the bridge domain level) at any time.
- Port profiles are not in effect if the bridge domain does not have a profile attached to it.
- IGMP snooping must be enabled on the bridge domain for any port-specific configurations to be in effect.
- If a profile attached to a bridge domain contains port-specific configuration options, the values apply to all of the ports under the bridge, including all mrouter and host ports, unless another port-specific profile is attached to a port.
- When a profile is attached to a port, IGMP snooping reconfigures that port, disregarding any port configurations that may exist in the bridge-level profile.

Creating Profiles

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

Attaching and Detaching Profiles

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

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

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

Changing Profiles

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

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

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

Default IGMP Snooping Configuration Settings

Table 15: IGMP Snooping Default Configuration Values

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

IGMP Snooping Configuration at the Bridge Domain Level

IGMP Minimum Version

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

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

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

Group Membership Interval, Robustness Variable, and Query Interval

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

The GMI is calculated as:

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

where:

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

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

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

EVPN All-Active Multi-homed Multicast Source Behind a BVI

Table 16: Feature History Table

Feature Name	Release Information	Feature Description
EVPN All-Active Multi-homed Multicast Source Behind a BVI	Release 7.11.1	We have enhanced multicast routing efficiency, load balancing, and latency in EVPN topology by optimizing redundancy and enabling support for All-Active (AA) multicast multi-homed sources. The multi-homed multicast data sources are located behind a Bridge-Group Virtual Interface (BVI), while multicast receivers can be in either the core or a bridge domain. This feature introduces the following changes: • **CLI** • The **ole-collapse-disable** keyword is introduced in the **hw-module multicast evpn** command. • **YANG Data Model** • New leaf **evpn-ole-collapse-disable** added in `Cisco-IOS-XR-fia-hw-profile-cfg.yang` (see GitHub, YANG Data Models Navigator).

EVPN AA multi-homed refers to a specific deployment model within the EVPN technology. In the multi-homed setup, a customer site or device (CE) is connected to multiple provider edge (PE) routers or attachment circuits (ACs). Multi-homing provides redundancy and load balancing by allowing a CE to connect to multiple PE routers, enabling traffic to be distributed across different paths. In case of a link (CE to PE and local PE to remote PE) or router failure, traffic can be quickly redirected to an alternate path.

In multi-homing, an AA mode means that all the links or paths between the EVPN sites are active and forwarding traffic simultaneously. This is in contrast to other deployment models, such as Single-Active or Port-Active Load-balancing mode, where only a subset of the links is active at any given time.

Placing the CE device behind the BVI interface has the following advantages:

- It allows for a simplified configuration on the CE side. The CE only needs to be configured with a single default gateway, which is the BVI interface. The CE doesn't have to manage multiple interfaces or deal with complex routing protocols.
- The BVI interface also enables efficient replication and forwarding of multicast traffic to the appropriate multicast distribution trees within the service provider network. This eliminates the need for the CE to handle multicast replication, reducing its processing load and potentially improving overall multicast performance.
- Placing the CE behind the BVI accept interface allows for greater flexibility in multi-homing scenarios. The CE can connect to multiple provider edge (PE) routers through the BVI accept interface, enabling seamless failover and load balancing between the PE routers during link or router failures.

Prerequisites

The network must support the following topology, protocols, and features to use the EVPN AA multi-homed multicast source feature:

- EVPN Control Plane with BGP
- BVI
- IGMP Snooping and MLD Snooping
- MLDP, MPLS, and OSPF (for L3 multicast receivers at core)
- Native multicast, MVPN GRE, or mVPN Profile 14 (core)

For more information related to EVPN technology and supported protocols, refer *EVPN Features* chapter in *L2VPN and Ethernet Services Configuration Guide for Cisco NCS 5500 Series Routers*.

For more information related to IGMP Snooping and MLD Snooping features, refer *Implementing Layer 2 Multicast* chapter in *Multicast Configuration Guide for Cisco NCS 5500 Series Routers*.

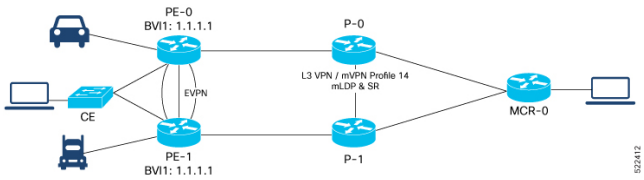
The EVPN AA multi-homed multicast source feature enables multicast data packet support for multi-homed sources in an EVPN AA (All-Active) topology.

In this setup, the multicast traffic is forwarded to the core by EVPN with BVI as the accept interface.

This deployment model combines the benefits of AA forwarding and multi-homing. It’s particularly useful in scenarios where high availability, fault tolerance, and optimized bandwidth utilization are essential requirements.

The following illustration shows the multicast data traffic route between a multi-homed source and the multi-homed receivers.

Figure 8: EVPN All-Active Multi-homed Multicast Source behind a BVI Topology



In this illustration, the multicast data sources are connected behind a CE, which is multi-homed to PE-0 and PE-1. PE-1 is configured with a BVI that has an anycast IP address. The image displays an example where

the BVI has the IP address 1.1.1.1. The receiver that is behind the MCR-0 has a PIM connection toward the multicast data source.

The data packet flow between the multicast data source and receiver occurs in the following manner:

1. The receiver, located behind Multicast Receiver (MCR-0), initiates an Internet Group Management Protocol (IGMP) join, which triggers a Protocol Independent Multicast (PIM) join towards the source.
2. The PIM join message reaches one of the PE routers (either PE-0 or PE-1) with the incoming or accept interface being the BVI and the outgoing interface leading towards the core network.
3. When the source sends traffic, it reaches one of the PE routers (PE-0 or PE-1). The next path for the traffic depends on the following IGMP snooping configurations:
   - If IGMP snooping is enabled and the multicast source is configured for both IPv4 and IPv6 traffic, the traffic is forwarded to either a route with a BVI interface or the default IGMP snoopig route.
   - If IGMP snooping is disabled, the traffic floods the multicast ID (MCID) on the bridge. As part of the flood MCID logic, the packet is recycled for the BVI and flooded to all the ACs, including the EVPN Optimized Local Egress (OLE). The recycled packets for the BVI undergo Layer 3 lookup. If there is a route with the BVI as an accepted interface, the packet is forwarded to the Olist for Layer 3 forwarding.

Note The same packet is not sent back to the CE device due to SHL (Split Horizon Label) filtering for EVPN traffic.

Usage Guidelines and Limitations

The supported scenarios for AA MH multicast are as follows:

- IPv4 SSM with BVI as accept interface is supported.
- IPv4 SM with BVI as accept interface is supported.
- IPv6 SSM with BVI as accept interface is supported.
- IPv4 SSM without BVI (only layer 2 multicast) and multicast source behind L2 is supported.
- IPv4 SM without BVI as accept interface (only layer 2 multicast) is supported.
- IPv6 SSM without BVI as accept interface (only layer 2 multicast) is supported.
- IPv6 SM without BVI as accept interface (only layer 2 multicast) is supported.

This feature has the following limitations:

- IPv6 SM with BVI as accept interface is not supported.
- Dual-homed source and Dual-homed receiver over MLDP profile on the same BD is not supported. It is recommended to disable MVPN peering between the MH nodes to prevent redundant traffic path formation in the core.
- Layer 2 IPv6 traffic is only supported on NCS 5700 fixed port routers and NCS 5500 modular routers (NCS 5700 line cards [Mode: Native]).
- In an EVPN dual-home AA scenario:

- If the multicast source and receiver are in the same BD, the receiver might receive permanent traffic duplication.
- Transient traffic duplication might occur when the DH node role changes between DF and nDF.
- In a BD, the following EVPN configuration is not supported:
  - Multicast source—ESI1=BE-X.A
  - Multicast receiver—ESI1=BE-X.B

Note ESI is the Ethernet Segment identifier, whereas X.Aand X.B represents two AC ports for the bundle interface BE.

- In a BD, the following EVPN configuration is not supported:
  - Multicast source—ESI1=BE-X.A (NCS 5700 line cards)
  - Multicast receiver—ESI1=BE-Y.A (NCS 5500 line cards)

Note ESI is the Ethernet Segment identifier, whereas X.Aand Y.A represents two AC ports for the bundle interface BE.

Configure EVPN All-Active Multi-homed Multicast Source with a BVI Interface

To configure an EVPN All-Active Multi-homed multicast source with a BVI interface, use the following example configuration:

```
Router#configure
Router(config)#l2vpn
Router(config-l2vpn)#bridge group BG1
Router(config-l2vpn-bg)#bridge-domain BD1
Router(config-bg-bd)#multicast-source ipv4-ipv6
Router(config-bg-bd)#mld snooping profile mldsn
Router(config-bg-bd)#igmp snooping profile igmpsn
Router(config-bg-bd)#interface Bundle-Ether1
Router(config-bg-bd-ac)#exit
Router(config-bg-bd)#interface TenGigE0/0/0/23.1
Router(config-bg-bd-ac)#exit
Router(config-bg-bd)#routed interface BVI1
Router(config-bg-bd)#evi 3000
Router(config-bg-bd-bvi)#commit
```

Running Configuration

This section shows the EVPN All-Active Multi-homed multicast source with BVI as accept interface running configuration.

```
l2vpn
bridge group bg1
```

```
bridge-domain bd1
multicast-source ipv4-ipv6
mld snooping profile mldsn
igmp snooping profile igmpsn
interface Bundle-Ether1
!
interface TenGigE0/0/0/23.1
!
routed interface BVI1
!
evi 3000
!
!
!
!
```

Disable EVPN Core Replications

Default behavior in EVPN involves collapsing core replications into L2 multicast routes (BD, S, G). To modify this behaviour and collapse EVPN Core to Bridge ingress multicast ID (MCID) and Snooping default routes instead of L2 multicast routes, use the following command:

```
Router(config)# hw-module multicast evpn ole-collapse-disable
```

Sample Configuration

```
Router(config)# hw-module multicast evpn ole-collapse-disable
Mon Apr 3 20:37:39.218 UTC

/*To apply the disable or re-enable EVPN OLE collapse settings, you must reload the chassis
 and all the installed line cards*/
Router# commit
Mon Apr 3 20:37:46.886 UTC
Router# end

Router# admin
Mon Apr 3 20:37:52.234 UTC
lab connected from 1.1.1.1 using ssh on sysadmin-vm:0_RP0
Reloading the RP in Order to apply the HW-cli Evpn ole collapse disable command to set

sysadmin-vm:0_RP0# hw-module location 0/RP0 reload
Mon Apr 3 20:38:15.290 UTC+00:00
Reload hardware module ? [no,yes]

/*Verification After Reload*/
Router# sh dpa objects global location 0/0/cPU0 | i evpn
Mon Apr 3 20:48:38.939 UTC
ofa_bool_t mcast_evpn_ole_collapse_disable => TRUE.

Router# sh running-config | i hw-
Mon Apr 3 20:48:43.575 UTC
hw-module multicast evpn ole-collapse-disable
```

Verification

Verify that you have configured multicast over BVI. The BVI acts as a forwarding interface for the L3 multicast packets.

```
/*PE-0*/
Router# show mrib vrf green ipv4 route 40.0.0.5
Mon May 8 12:15:44.924 UTC
(40.0.0.5,232.0.0.1) RPF nbr: 40.0.0.5 Flags: RPF
```

```
Up: 00:04:03
Incoming Interface List
BVI1 Flags: F A LI, Up: 00:04:03
Outgoing Interface List
BVI1 Flags: F A LI, Up: 00:04:03

/*Local L3 multicast join*/
TenGigE0/0/0/0.2 Flags: F NS LI, Up: 00:04:03

/*PE-1*/
Router# show mrib vrf green ipv4 route 40.0.0.5 detail
Thu May 11 09:19:07.958 UTC
(40.0.0.5,232.0.0.1) Ver: 0x1008 RPF nbr: 40.0.0.5 Flags: RPF EID, FGID: 15481, Statistics
 enabled: 0x0, Tunnel RIF: 0xffffffff, Tunnel LIF: 0xffffffff
  Up: 05:29:49
  RPF-ID: 0, Encap-ID: 262146
  Incoming Interface List
    BVI1 Flags: F A LI, Up: 05:29:49
  Outgoing Interface List
    BVI1 Flags: F A LI, Up: 05:29:49
/*Remote L3 join from multicast receiver learnt on PE-1. Multicast traffic to remote L3
multicast receiver is forwarded from PE-1*/
    Lmdtgreen Flags: F LMI TR, Up: 05:27:02, Head LSM-ID: 0x00001

/*Local L3 multicast join*/
    TenGigE0/0/0/23.2 Flags: F NS LI, Up: 05:29:48
```

How to Configure IGMP Snooping

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

Creating an IGMP Snooping Profile

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**igmp snooping profile** *profile-name* **Example:** `RP/0/RP0/CPU0:router(config)# igmp snooping profile default-bd-profile`	Enters IGMP snooping profile configuration mode and creates a named profile. The default profile enables IGMP snooping. You can commit the new profile without any additional configurations, or you can include additional configuration options to the profile. You can also return to the profile later to add configurations, as described in other tasks in this module.
Step 3	Optionally, add commands to override default configuration values.	If you are creating a bridge domain profile, consider the following: • An empty profile is appropriate for attaching to a bridge domain. An empty

	Command or Action	Purpose
		profile enables IGMP snooping with default configuration values. • You can optionally add more commands to the profile to override default configuration values. • If you include port-specific configurations in a bridge domain profile, the configurations apply to all ports under the bridge, unless another profile is attached to a port. If you are creating a port-specific profile, consider the following: • While an empty profile could be attached to a port, it would have no effect on the port configuration. • When you attach a profile to a port, IGMP snooping reconfigures that port, overriding any inheritance of configuration values from the bridge-domain profile. You must repeat the commands in the port profile if you want to retain those configurations. You can detach a profile, change it, and reattach it to add commands to a profile at a later time.
Step 4	**commit**	

Where to Go Next

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.

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**l2vpn** **Example:**	Enters Layer 2 VPN configuration mode.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config)# l2vpn`	
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	**multicast-source ipv4** **Example:** `RP/0/RP0/CPU0:router(config)# multicast-source ipv4`	Configures Layer 2 multicast routes with IGMP snooping.
Step 6	**igmp snooping profile** *profile-name* **Example:** `RP/0/RP0/CPU0:router(config-l2vpn-bg-bd)# igmp snooping profile default-bd-profile`	Attaches the named IGMP snooping profile to the bridge domain, enabling IGMP snooping on the bridge domain.
Step 7	**commit**	
Step 8	**show igmp snooping bridge-domain detail** **Example:** `RP/0/RP0/CPU0:router# show igmp snooping bridge-domain detail`	(Optional) Verifies that IGMP snooping is enabled on a bridge domain and shows the IGMP snooping profile names attached to bridge domains and ports.
Step 9	**show l2vpn bridge-domain detail** **Example:** `RP/0/RP0/CPU0:router# show l2vpn bridge-domain`	(Optional) Verifies that IGMP snooping is implemented in the forwarding plane (Layer 2) on a bridge domain.

Detaching a Profile and Deactivating IGMP Snooping on a Bridge Domain

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

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

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**l2vpn** **Example:** `RP/0/RP0/CPU0:router(config)# l2vpn`	Enters Layer 2 VPN configuration mode.
Step 3	**bridge group** *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 disable** **Example:** `RP/0/RP0/CPU0:router(config-l2vpn-bg-bd)# no igmp snooping disable`	Detaches the IGMP snooping profile from the bridge domain, disabling IGMP snooping on that bridge domain. **Note** Only one profile can be attached to a bridge domain at a time. If a profile is attached, IGMP snooping is enabled. If a profile is not attached, IGMP snooping is disabled.
Step 6	**commit**	
Step 7	**show igmp snooping bridge-domain detail** **Example:** `RP/0/RP0/CPU0:router# show igmp snooping`	(Optional) Verifies that IGMP snooping is disabled on a bridge domain.

	Command or Action	Purpose
	`bridge-domain detail`	
Step 8	**show l2vpn bridge-domain detail** **Example:** `RP/0/RP0/CPU0:router# show l2vpn bridge-domain`	(Optional) Verifies that IGMP snooping is disabled in the forwarding plane (Layer 2) on a bridge domain.

Attaching and Detaching Profiles to Ports Under a Bridge

Before you begin

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

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**l2vpn** **Example:** `RP/0/RP0/CPU0:router(config)# l2vpn`	Enters Layer 2 VPN configuration mode.
Step 3	**bridge group** *bridge-group-name* **Example:** `RP/0/RP0/CPU0:router(config-l2vpn)# bridge group GRP1`	Enters Layer 2 VPN bridge group configuration mode for the named bridge group.
Step 4	**bridge-domain** *bridge-domain-name* **Example:** `RP/0/RP0/CPU0:router(config-l2vpn-bg)# bridge-domain ISP1`	Enters Layer 2 VPN bridge group bridge domain configuration mode for the named bridge domain.
Step 5	**interface** *interface-type interface-number* **Example:** `RP/0/RP0/CPU0:router(config-l2vpn-bg-bd)# interface gig 1/1/1/1`	Enters Layer 2 VPN bridge group bridge domain interface configuration mode for the named interface or PW.

	Command or Action	Purpose
Step 6	**multicast-source ipv4** **Example:** `RP/0/RP0/CPU0:router(config)# multicast-source ipv4`	Configures L2 multicast routes in L2 multicast with IGMP Snooping.
Step 7	Do one of the following: • **igmp snooping profile** *profile-name* • **no igmp snooping** **Example:** `RP/0/RP0/CPU0:router(config-l2vpn-bg-bd-if)# igmp snooping profile mrouter-port-profile`	Attaches the named IGMP snooping profile to the port. **Note** A profile on a port has no effect unless there is also a profile attached to the bridge. The **no** form of the command detaches a profile from the port. Only one profile can be attached to a port.
Step 8	**commit**	
Step 9	**show igmp snooping bridge-domain detail** **Example:** `RP/0/RP0/CPU0:router# show igmp snooping bridge-domain detail`	(Optional) Verifies that IGMP snooping is enabled on a bridge domain and shows the IGMP snooping profile names attached to bridge domains and ports.
Step 10	**show l2vpn bridge-domain detail** **Example:** `RP/0/RP0/CPU0:router# show l2vpn bridge-domain`	(Optional) Verifies that IGMP snooping is implemented in the forwarding plane (Layer 2) on a bridge domain.

Verifying Multicast Forwarding

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**show l2vpn forwarding bridge-domain** [*bridge-group-name*:*bridge-domain-name*] **mroute ipv4** [**group** *group_IPaddress*] [**hardware** {**ingress** | **egress**}] [**detail**]**location** *node-id* **Example:**	Displays multicast routes as they are converted into the forwarding plane forwarding tables. Use optional arguments to limit the display to specific bridge groups or bridge domains. If these routes are not as expected, check the control plane configuration and correct the corresponding IGMP snooping profiles.

	Command or Action	Purpose
	RP/0/RP0/CPU0:routershow l2vpn forwarding bridge-domain bg1:bd1 mroute ipv4 group 234.192.4.1 hardware ingress detail location 0/1/cPU0	
Step 3	**show l2vpn forwarding bridge-domain** [*bridge-group-name*:*bridge-domain-name*] **mroute ipv4 summary location** *node-id* **Example:** RP/0/RP0/CPU0:router# show l2vpn forwarding bridge-domain bg1:bd1 mroute ipv4 summary location 0/3/CPU0	Displays summary-level information about multicast routes as stored in the forwarding plane forwarding tables. Use optional arguments to limit the display to specific bridge domains.

Configuration Examples for IGMP Snooping

The following examples show how to enable IGMP snooping on Layer 2 bridge domains on Cisco NCS 540 Series Routers:

Configuring IGMP Snooping on Physical Interfaces Under a Bridge: Example

1. Create two profiles.

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

!
```

2. Configure two physical interfaces for L2 transport.

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

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

```
l2vpn
```

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

!
  !
!
```

4. Verify the configured bridge ports.

```
show igmp snooping port
```

Configuring IGMP Snooping on VLAN Interfaces Under a Bridge: Example

1. Configure two profiles.

```
multicast-source ipv4
igmp snooping profile bridge_profile

igmp snooping profile port_profile

!
```

2. Configure VLAN interfaces for L2 transport.

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

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

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

```
    interface GigabitEthernet0/8/0/8.2

    !
  !
!
```

4. Verify the configured bridge ports.

```
show igmp snooping port
```

Configuring IGMP Snooping on Ethernet Bundles Under a Bridge: Example

1. Configure two IGMP snooping profiles.

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

       !
```

2. Configure interfaces as bundle member links.

```
       interface GigabitEthernet0/0/0/0
         bundle id 1 mode on
         negotiation auto
       !
       interface GigabitEthernet0/0/0/1
         bundle id 1 mode on
         negotiation auto
       !
       interface GigabitEthernet0/0/0/2
         bundle id 2 mode on
         negotiation auto
       !
       interface GigabitEthernet0/0/0/3
         bundle id 2 mode on
         negotiation auto
       !
```

3. Configure the bundle interfaces for L2 transport.

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

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

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

     !
   !
!
```

5. Verify the configured bridge ports.

```
show igmp snooping port
```

Configuring Multicast over Integrated Routing Bridging Active/Active Multihome

Configurations performed on peer 1:

1. Layer 2 Base Configuration

```
hostname peer1
!
interface Bundle-Ether2
!
interface Bundle-Ether2.2 l2transport
 encapsulation dot1q 2
 rewrite ingress tag pop 1 symmetric
!
interface TenGigE0/0/0/0
 bundle id 2 mode on
 no shut
!
```

2. EVPN Configuration

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

```
 !
 interface Bundle-Ether2
  ethernet-segment
   identifier type 0 02.02.02.02.02.02.02.02.02
   bgp route-target 0002.0002.0002
  !
 !
!
```

3. IGMPv2 Snoop Configurations

```
hostname peer1
!
router igmp

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

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

Configurations Performed on Peer 2:

1. Layer 2 Base Configuration

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

2. EVPN Configuration

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

```
!
evpn
 evi 2
  advertise-mac
  !
 !
 interface Bundle-Ether2
  ethernet-segment
   identifier type 0 02.02.02.02.02.02.02.02.02
   bgp route-target 0002.0002.0002
  !
 !
!
```

3. IGMPv2 Snoop Configurations

```
hostname peer2
!
router igmp

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

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

Verifying IGMP Snooping and EVPN Sync

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

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

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

                          Bridge Domain VLAN10:VLAN10

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


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

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

                          Bridge Domain VLAN10:VLAN10

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

Verifying Dual DR PIM Uplink

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

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

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

(126.0.0.100,239.0.0.2) RPF nbr: 30.0.0.4 Flags: RPF
  Up: 00:03:34
  Incoming Interface List
    HundredGigE0/0/0/1 Flags: A, Up: 00:03:34
  Outgoing Interface List
    BVI2 Flags: F NS, Up: 00:03:34
:
:

RP/0/RP0/CPU0:peer2#show mrib route
:
:

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

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

Verifying Designated Forwarder Election

As described in the previous example, both peer1 and peer2 have BVI2 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
:
:

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: 0x80000001 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
:
:

Bridge-Domain: VLAN2:VLAN2, ID: 0
  Prefix: (0.0.0.0,239.0.0.2/32)
  P2MP enabled: N
  IRB platform data: {0x0, 0x30, 0x0, 0x0}, len: 32
  Bridge Port:
  EVPN, Xconnect id: 0x80000001 NH:1.1.1.1
  Bundle-Ether2.2, Xconnect id: 0xa0000029
```

Additional References

Related Documents

Related Topic	Document Title
Configuring MPLS VPLS bridges	Implementing Virtual Private LAN Services on Cisco IOS XR Software module in the *MPLS Configuration Guide for Cisco NCS 540 Series Routers*
Getting started information	
Configuring EFPs and EFP bundles	*Interface and Hardware Component Configuration Guide for Cisco NCS 540 Series Routers*

Standards

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

[1] Not all supported standards are listed.

MIBs

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

RFCs

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

Technical Assistance

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

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 540 Series Routers*.

Prerequisites for MLD Snooping

- The network must be configured with a layer2 VPN.

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

Supported Features and Restrictions for MLD Snooping

- BVI under bridge domain is supported.
- Receiver behind L2 ACs in the same L2 bridge domain is supported.
- Source behind L2 ACs in the same L2 bridge domain is only supported on NCS 5700 fixed port routers and NCS 5700 line cards [Mode: Compatibility; Native].
- MLDv1 not supported over BVI.
- EVPN MLD sync is not supported.
- VPLS is not supported.
- On the NCS 5700 line cards, MLD snooping can be enabled alongside IGMP snooping only.
- The **router-alert-check disable** configuration command is not supported.
- EVPN dual-home source AA is not supported on the NCS 5500 line cards line cards.
- Both IGMP and MLD snooping configurations are necessary to enable MLD snooping on the NCS 5700 line cards.
- EVPN configuration must have the **control-word-disable** configuration.
- PIM control packets (join and hello) processing is not supported when snooping is enabled, so a multicast router selection based on PIM packets won't occur.
- Explicit host tracking.
- Multicast Admission Control.
- Security filtering.
- Report rate limiting.
- Multicast router discovery.
- IPv6 multicast is not supported for a multicast source that is behind the BVI interface. For example, the below configuration is not supported:

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

- In an EVPN dual-home AA scenario:
  - If the multicast source and receiver are in the same bridge domain (BD), the receiver might receive permanent traffic duplication.
  - In an EVPN dual-home receiver AA scenario, transient traffic duplication is expected when the DH node role changes from DF to nDF and vice versa.

- Source=ESI1=BE-X.A, Receiver=ESI1=BE-X.B under the same BD is not supported (where X.A and X.B represent two AC ports for the bundle interface BE).
- Source=ESI1=BE-X.A (for NCS 5700 line cards), Receiver=ESI2=BE-Y.A (for NCS 5500 line cards) under the same BD is not supported (where X.A and Y.A represent two AC ports for the bundle interface BE).

Advantages of MLD Snooping

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

High Availability (HA) features for MLD

MLD supports the following HA features:

- Process restarts
- 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.
- 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 540 Series Routers randomly nominates a single port from the bundle to carry the multicast traffic.

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

Multicast Router and Host Ports

MLD snooping classifies each port as one of the following:

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

Multicast Router Discovery for MLD

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

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

Multicast Traffic Handling for MLD

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

Table 17: Multicast Traffic Handling for a MLDv1 Querier

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

Table 18: Multicast Traffic Handling for a MLDv2 Querier

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

Multicast Listener Discovery over BVI

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

Note

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

MLD and BVI Overview

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

MLDv1 and MLDv2 are supported on NCS 540. However, MLDv2 is enabled when you configure MLD by default.

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

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

Multicast Traffic Over Layer 2 IPv6 Network

Table 19: Feature History Table

Feature Name	Release Information	Feature Description
Multicast Traffic over Layer 2 IPv6 Network	Release 7.9.1	This feature allows you to forward the IPv6 multicast packets only to the interested MLD-snooped Access Controllers (AC), whereas in the default case, the bridge floods the IPv6 multicast packets to all AC. Routers use Multicast Listener Discovery (MLD) protocol to discover the devices in a network and create route entries in an IPv6 multicast network. This feature introduces following CLI: • **multicast-source ipv6**

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

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

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

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

Hardware Supported

This feature is supported on routers that have the Cisco NC57 line cards installed and operate in native and compatible modes.

Limitations and Restrictions

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

Configuration Example

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

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


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

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

Verification

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

```
router# show mld snooping group

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

                 Bridge Domain bg1:bd1

Group           Ver GM  Source           PM  Port                      Exp Flg

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


                 Bridge Domain bg1:bd4

Group           Ver GM  Source           PM  Port                      Exp Flg
```

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

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

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

IPv6 Multicast Listener Discovery Snooping over BVI

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

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

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

Restrictions

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

Configuring Internal Querier for MLD Snooping

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

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

Verification

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

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

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

Creating a MLD Snooping Profile

Configuration

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

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

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

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

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

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

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

Running Configuration

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

Verification

Verify that the MLD snooping profile is created:

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

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

Deactivating MLD Snooping on a Bridge Domain

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

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

Configuration

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

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

Running Configuration

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

Configuring Static Mrouter Ports (MLD)

Prerequisite

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

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

Configuration

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

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

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

Running Configuration

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

Verification

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

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

MLD  Snoop Profile mrouter-port-profile:

  Static Mrouter:                       Enabled

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

Configuring 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-l2vpn-bg-bd)# commit
```

Running Configuration

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

Verification

Verify that the router guard config in the named profile is enabled:

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

  Router Guard:                          Enabled

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

Configuring Immediate-leave for MLD

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

Configuration

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

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

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

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

Running Configuration

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

Verification

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

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

MLD  Snoop Profile host-port-profile:

  Immediate Leave:                    Enabled
  Router Guard:                       Enabled

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

Configuring Internal Querier for MLD

Prerequisite

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

Configuration

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

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

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

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

Running Configuration

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

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

Verification

Verify that the internal querier config is enabled:

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

MLD  Snoop Profile internal-querier-profile:

  System IP Address:                  fe80::98

  Internal Querier Support:           Enabled

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

Configuring Static Groups for MLD

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

Prerequisite

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

Configuration

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

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

/* Configure a static group. */
/* Note: Repeat this step to add additional static groups. */
RP0/0/RP0/CPU0:router(config-mld-snooping-profile)# static group fe80::99 source fe80::99
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 fe80::99 source fe80::99
!
```

Verification

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

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


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

Configuring MLD Snooping

Configure

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

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

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

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

Running Configuration

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

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

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

Verification

Verify the configured bridge ports.

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


                        Bridge Domain f10:109

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

Configuring MLD Snooping on Ethernet Bundles

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

Configure

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

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

/* Configure interfaces as bundle member links. */

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

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

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

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

Running Configuration

```
RP0/0/RP0/CPU0:router# show running-config
configuration
 interface Port-channel1
    !
interface GigabitEthernet0/0/0/0
    !
interface GigabitEthernet0/0/0/1
!
    interface GigabitEthernet0/0/0/2
       channel-group 1 mode on
    !
    interface GigabitEthernet0/0/0/3
       channel-group 1 mode on
    !
mld snooping profile bridge_profile
       !
       mld snooping profile port_profile
          mrouter
       !
interface GigabitEthernet0/0/0/0
         bundle id 1 mode on
         negotiation auto
       !
       interface GigabitEthernet0/0/0/1
         bundle id 1 mode on
         negotiation auto
       !
       interface GigabitEthernet0/0/0/2
         bundle id 2 mode on
         negotiation auto
       !
       interface GigabitEthernet0/0/0/3
         bundle id 2 mode on
         negotiation auto
       !
interface Bundle-Ether 1
          l2transport
```

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

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

Verification

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

MLD Snooping Synchronization for EVPN Multi-Homing

Table 20: Feature History Table

Feature Name	Release Information	Feature Description
MLD Snooping Synchronization for EVPN Multi-Homing	Release 7.11.1	The Designated Forwarder (DF) PE router in an EVPN multi-homed network can now efficiently forward multicast traffic from the source to the interested receivers, avoiding unnecessary replication and reducing network bandwidth consumption. This is made possible by introducing support for Multicast Listener Discovery, MLDv1, and MLDv2 (IPv6) snooping state synchronization for EVPN multi-homing peers or provider edge (PE) devices, expanding the scope of the previous support for IGMP (IPv4) snooping state synchronization.

In an EVPN multi-homing network, where customer edge devices (CEs) are multi-homed to more than one PE device, the MLD snooping synchronization feature enables routers to accurately track multicast group membership information and forward multicast traffic only to the interested receivers.

In an All-Active redundancy mode, the CEs can send an MLD message to any one of the multi-homed PEs, either DF or non-DF. Only the EVPN DF forwards traffic for the bridge domain (BD) for any group. Therefore, all PEs attached to a given EVPN Segment (ES) must coordinate MLD Join and Leave Group (x, G) state, where x may be either '*' (unspecified multicast source address) or a particular source address 'S', and G represents the multicast group address, for each [EVI, broadcast domain (BD)] on that ES. This allows the DF for that ES, EVI, or BD to correctly advertise or withdraw a Selective Multicast Ethernet Tag route for that (x, G) group in that EVI or BD when needed.

In Single-Active redundancy mode, the PEs attached to a multi-homed ES coordinate the MLD Join (x, G) state. MLD join messages (MLD host membership reports) are received by the DF PE and distributed to the non-DF PEs for faster convergence. The non-DF PE also receives traffic by building the distribution tree toward the Rendezvous Point (RP) or multicast source, but doesn't forward it to the receivers in a multicast group. When a non-DF PE becomes the DF PE, it starts forwarding traffic to the CE.

Some benefits of the MLD state synchronization feature are as follows:

- **Seamless Mobility Support**—It ensures smooth mobility support for multicast listeners. When listeners move between different network devices or ports, the synchronized MLD snooping state helps maintain consistent multicast group membership information. The DF intelligently updates the forwarding information, ensuring uninterrupted multicast service delivery to mobile listeners.
- **Reduced Control Plane Overhead**—By synchronizing the MLD snooping state, we have reduced signaling messages overhead in the control plane for routing. The DF processes and propagates multicast control messages, such as MLD join and leave messages, only to the relevant ports based on the synchronized group membership information. This minimizes unnecessary control plane processing and improves network scalability.
- **Enhanced Network Stability**:—It contributes to network stability by maintaining consistent multicast group membership information across PE devices. This ensures reliable multicast service delivery and prevents disruptions or inconsistencies that could impact the network's overall performance.
- **Efficient Resource Utilization**—It uses a DF to optimize resource utilization by forwarding multicast traffic only to the ports where receivers are present. This prevents unnecessary multicast data replication and conserves network bandwidth, improving overall network efficiency.

MLD Snooping Synchronization with Proxy Querier

Each subnet has one of the two roles:

- Querier—the router with the lowest IP address in a subnet. Querier is responsible for sending the MLD or IGMP queries to know which multicast groups are active on the subnet.
- Non-Querier—the router that listens for MLD or IGMP queries and forwards them to the entire VLAN.

Initially, all multicast routers start up as a Querier on each attached network. If a router hears a Query message from a lower IP address, it becomes a Non-Querier. If a router doesn't hear a Query message for a certain period, it becomes the Querier again. The Querier router regularly sends a General Query on each attached network to gather multicast group membership information.

In this feature, two peer PEs in EVPN can both act as Queriers for the same BD. The first PE receiving the MLD join from CE sends an EVPN Join sync message to the second peer PE, which, upon receipt, sets the

"learnt via EVPN " flag on the group. The group is not expired for lack of a direct MLD Join response to the initiated query (by second PE) as long as the 'learnt via EVPN' flag is set.

Note The MLD queries are not sent over the MPLS core. ACL filter is applied on the core facing interface to drop all the MLD queries.

Usage Guidelines and Limitations

A BD can have a combination of MLDv1 and MLDv2 receivers, sending corresponding Join messages to the PE router. Additionally, MLDv2 Join messages could either be in the include or exclude mode, where a multicast receiver can specify to either listen only for packets from some list of source addresses (include) or only for packets that don't come from some list of source addresses (exclude). Handling such mixed MLD Joins involves ensuring that the PE routers can properly interpret and forward multicast traffic to hosts accordingly.

In the following table, use the supported and unsupported scenarios for the MLDv1 and MLDv2 Joins at PE as guidelines for using the MLD snooping synchronization feature:

Table 21: MLDv1 and MLDv2 Combination Joins at PE—Supported and Unsupported Scenarios

Current MLD Join State	**MLD Join Update Received**			
	v1 (*, G)	**v2 Include (S, G)**	**v2 Exclude (*, G)**	**v2 Exclude (S, G)**
No state	Accepted	Accepted	Accepted	Drop
v1 (*, G)	Accepted	Drop	Accepted	Drop
v2 Include (S, G)	Drop	Accepted	Drop	Drop
v2 Exclude (*, G)	Accepted	Drop	Accepted	Drop

This feature has the following limitations:

- If the source is directly connected to the PE where the MLD Join is received, no MLD sync route is generated.
- Any router behind an All-Active multi-homed network is not supported.
- Configuring different MLD snooping profiles on peer PEs in an All-Active multi-homed network is not supported.
- An mrouter port behind CE is not supported.
- To prevent convergence issues, per multicast route DF election is not supported.
- The IGMP and MLD snooping profiles must be enabled together.

Configure MLD Snooping Synchronization for EVPN Multi-Homing

To configure MLD Snooping Synchronization for EVPN Multi-Homing, use the following example configuration:

/* Configure the EVPN EVI */

```
Router(config)# evpn
Router(config-evpn)# interface Bundle-Ether34
Router(config-evpn-ac)# ethernet-segment
Router(config-evpn-ac-es)# identifier type 0 23.23.23.11.FF.11.11.11.11
Router(config-evpn-ac-es)# exit
Router(config-evpn-ac)# exit
Router(config-evpn)# evi 5
Router(config-evpn-instance)# advertise-mac
Router(config-evpn-instance)# exit
Router(config-evpn)# exit
```

/* Configure the L2VPN BD with MLD snooping profile and EVI */

```
Router(config)# l2vpn
Router(config-l2vpn)# bridge group bg1
Router(config-l2vpn-bg)# bridge-domain bd5
Router(config-l2vpn-bg-bd)# mld snooping profile prof1
Router(config-l2vpn-bg-bd-mld-snooping-profile)# exit
Router(config-l2vpn-bg-bd)# igmp snooping profile prof2
Router(config-l2vpn-bg-bd-igmp-snooping-profile)# exit
Router(config-l2vpn-bg-bd)# interface Bundle-Ether34.5
Router(config-l2vpn-bg-bd-ac)# exit
Router(config-l2vpn-bg-bd)# routed interface BVI5
Router(config-l2vpn-bg-bd-bvi)# exit
Router(config-l2vpn-bg-bd)# evi 5
Router(config-l2vpn-bg-bd-evpn-instance)# exit
```

/* Configure the MLD snooping profile

```
Router(config)# mld snooping profile prof1
Router(config-mld-snooping-profile)# internal-querier
Router(config-mld-snooping-profile)# internal-querier query-interval 5
Router(config-mld-snooping-profile)# commit
```

Running Configuration

```
/*EVPN EVI*/
evpn
    interface Bundle-Ether34
      ethernet-segment
        identifier type 0 23.23.23.11.FF.11.11.11.11
    !
    evi 5
      advertise-mac
      !
    !
```

```
/* Configure the L2VPN BD with MLD snooping profile and EVI */
l2vpn
 bridge group bg1
   bridge-domain bd5
     mld snooping profile prof1
     igmp snooping profile prof2
     interface Bundle-Ether34.5
     !
     routed interface BVI5
     !
     evi 5
     !

/*MLD Snooping Profile*/
mld snooping profile prof1
     internal-querier
     internal-querier query-interval 5
```

Verify MLD Snooping Synchronization for EVPN Multi-Homing

To verify the configuration for this feature, use the following example commands.

/*Verify MLD Snooping Synchronization*/

```
RP/0/RP1/CPU0:tb11-r8#show mld  snooping  group
Fri Oct  6 17:53:42.640 UTC

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

                            Bridge Domain bg-1:bd-1001

Ver GM PM Port                          Exp   Flgs Group,Source
--- -- -- ----                          ---   ---- ------------
*/B indicates MLD snooping sync through BGP*/
V2  IN IN BE1.1001                      never B    ff03::1,1108:101::100

*/D indicates MLD snooping state is locally learned through EVPN*/
V2  IN IN BE2.1001                      223   D    ff03::1,1108:101::100
```

/*Verify DF Election*/

```
RP/0/RP1/CPU0:tb11-r8#show evpn ethernet-segment carving detail

Tue Oct 17 18:14:56.607 UTC
Legend:
  B   - No Forwarders EVPN-enabled,
  C   - MAC missing (Backbone S-MAC PBB-EVPN / Grouping ES-MAC vES),
  RT  - ES-Import Route Target missing,
  E   - ESI missing,
  H   - Interface handle missing,
  I   - Name (Interface or Virtual Access) missing,
  M   - Interface in Down state,
  O   - BGP End of Download missing,
  P   - Interface already Access Protected,
  Pf  - Interface forced single-homed,
  R   - BGP RID not received,
  S   - Interface in redundancy standby state,
  X   - ESI-extracted MAC Conflict
  SHG - No local split-horizon-group label allocated
  Hp  - Interface blocked on peering complete during HA event
```

```
  Rc  - Recovery timer running during peering sequence

Ethernet Segment Id      Interface                          Nexthops
------------------------ ---------------------------------- --------------------
0000.0100.ac00.0001.0a00 BE1                                7.7.7.7
                                                            8.8.8.8
  ES to BGP Gates   : Ready
  ES to L2FIB Gates : Ready
  Main port         :
     Interface name : Bundle-Ether1
     Interface MAC  : b402.1657.e485
     IfHandle       : 0x2000a164
     State          : Up
     Redundancy     : Not Defined
  ESI ID            : 1
  ESI type          : 0
     Value          : 0000.0100.ac00.0001.0a00
  ES Import RT      : 0001.00ac.0000 (from ESI)
  Source MAC        : 0000.0000.0000 (N/A)
  Topology          :
     Operational    : MH, All-active
     Configured     : All-active (AApF) (default)
  Service Carving   : Auto-selection
     Multicast      : Disabled
  Convergence       :
  Peering Details   : 2 Nexthops
     7.7.7.7 [MOD:P:00:T]
     8.8.8.8 [MOD:P:00:T]
  Service Carving Synchronization:
     Mode           : NONE
     Peer Updates   :
                 7.7.7.7 [SCT: N/A]
                 8.8.8.8 [SCT: N/A]
  Service Carving Results:
     Forwarders     : 999
     Elected        : 500
            EVI E   :     1001,    1003,    1005,    1007,    1009,    1011
            EVI E   :     1013,    1015,    1017,    1019,    1021,    1023,
            …..
            EVI E   :     1999,    2001
     Not Elected    : 499
            EVI NE  :     1002,    1004,    1006,    1008,    1010,    1012
           …
            EVI NE  :     1990,    1992,    1994,    1996,    1998,    2000,
            EVI NE  :     2002
  …..
     Main port         :
     Interface name : Bundle-Ether2
     Interface MAC  : b402.1657.e484
     IfHandle       : 0x2000a16c
     State          : Up
     Redundancy     : Not Defined
  ESI ID            : 1
  ESI type          : 0
     Value          : 0011.0200.ac00.0001.0a00
  ES Import RT      : 1102.00ac.0000 (from ESI)
  Source MAC        : 0000.0000.0000 (N/A)
  Topology          :
     Operational    : MH, All-active
     Configured     : All-active (AApF) (default)
  Service Carving   : Auto-selection
     Multicast      : Disabled
  Convergence       :
  Peering Details   : 2 Nexthops
```

```
      7.7.7.7 [MOD:P:00:T]
      8.8.8.8 [MOD:P:00:T]
   Service Carving Synchronization:
      Mode           : NONE
      Peer Updates   :
                  7.7.7.7 [SCT: N/A]
                  8.8.8.8 [SCT: N/A]
   Service Carving Results:
      Forwarders     : 998
      Elected        : 500
             EVI E   :     1001,    1003,    1005,    1007,    1009,    1011
            …
             EVI E   :     1987,    1989,    1991,    1993,    1995,    1997,
             EVI E   :     1999,    2001
      Not Elected    : 498
             EVI NE  :     1002,    1004,    1006,    1008,    1010,    1012
             EVI NE  :     1980,    1982,    1984,    1986,    1988,    1990,
             EVI NE  :     1992,    1994,    1996,    1998,    2000,    2002
   EVPN-VPWS Service Carving Results:
      Primary        : 0
      Backup         : 0
      Non-DF         : 0
   MAC Flushing mode : STP-TCN
   Peering timer     : 3 sec [not running]
   Recovery timer    : 30 sec [not running]
   Carving timer     : 0 sec [not running]
   Revert timer      : 0 sec [not running]
   HRW Reset timer   : 5 sec [not running]
   Local SHG label   : 27051
   Remote SHG labels : 1
               27051 : nexthop 7.7.7.7
   Access signal mode: Bundle OOS

N/A                       Te0/1/0/4/0                          8.8.8.8
   ES to BGP Gates   : Ready
   ES to L2FIB Gates : Ready
   Main port         :
      Interface name : TenGigE0/1/0/4/0
      Interface MAC  : b402.1657.e0a0
      IfHandle       : 0x020040c8
      State          : Up
      Redundancy     : Not Defined
   ESI ID            : 0
   ESI type          : Invalid
   ES Import RT      : 0000.0000.0000 (Incomplete Configuration)
   Source MAC        : b402.1657.e480 (PBB BSA, no ESI)
   Topology          :
      Operational    : SH
      Configured     : Single-active (AApS) (default)
   Service Carving   : Auto-selection
      Multicast      : Disabled
   Convergence       :
   Peering Details   : 1 Nexthops
      8.8.8.8 [MOD:P:00]
   Service Carving Synchronization:
      Mode           : NONE
      Peer Updates   :
                  8.8.8.8 [SCT: N/A]
   Service Carving Results:
      Forwarders     : 10
      Elected        : 10
             EVI E   :     1001,    1002,    1003,    1004,    1005,    1006
             EVI E   :     1007,    1008,    1009,    1010
      Not Elected    : 0
```

```
  EVPN-VPWS Service Carving Results:
     Primary        : 0
     Backup         : 0
     Non-DF         : 0
  MAC Flushing mode : STP-TCN
  Peering timer     : 0 sec [not running]
  Recovery timer    : 0 sec [not running]
  Carving timer     : 0 sec [not running]
  Revert timer      : 0 sec [not running]
  HRW Reset timer   : 5 sec [not running]
  Local SHG label   : None
  Remote SHG labels : 0
  Access signal mode: Unsupported
```

/*Verify EVPN IGMP Snooping*/

```
RP/0/RSP0/CPU0:tb8-r3-AVA2#show evpn igmp
Mon Nov  6 11:18:19.497 UTC

EVI   Ethernet Segment         (S,G)                                      Source
                Type
----- ---------------------- ------------------------------------------
------------------------------ ------
1001  0000.0100.ac00.0001.0a00 (1108:101::100,ff03::1)                 Bundle-Ether1.1001
            JOIN
1001  0011.0200.ac00.0001.0a00 (1108:101::100,ff03::1)                 Bundle-Ether2.1001
            JOIN
1001  0000.0100.ac00.0001.0a00 (1108:101::100,ff03::1:2)               Bundle-Ether1.1001
            JOIN
1001  0011.0200.ac00.0001.0a00 (1108:101::100,ff03::1:2)               Bundle-Ether2.1001
            JOIN
1001  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:16:1::1)            Bundle-Ether2.1001
            JOIN
1001  0000.0100.ac00.0001.0a00 (1108:101::100,ff03:123:1::1)           Bundle-Ether1.1001
            JOIN
1001  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:123:1::1)           Bundle-Ether2.1001
            JOIN
1001  0000.0100.ac00.0001.0a00 (2205:101::23,ff03:13:1::1)             Bundle-Ether1.1001
            JOIN
1001  0011.0200.ac00.0001.0a00 (2205:101::23,ff03:13:1::1)             Bundle-Ether2.1001
            JOIN
1002  0000.0100.ac00.0001.0a00 (1108:101::100,ff03::2)                 Bundle-Ether1.1002
            JOIN
1002  0011.0200.ac00.0001.0a00 (1108:101::100,ff03::2)                 Bundle-Ether2.1002
            JOIN
1002  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:16:2::1)            Bundle-Ether2.1002
            JOIN
1002  0000.0100.ac00.0001.0a00 (1108:101::100,ff03:123:2::1)           Bundle-Ether1.1002
            JOIN
1002  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:123:2::1)           Bundle-Ether2.1002
            JOIN
1002  0000.0100.ac00.0001.0a00 (2205:101::23,ff03:14:1::1)             Bundle-Ether1.1002
            JOIN
1002  0011.0200.ac00.0001.0a00 (2205:102::441,ff03:14:1::1)            Bundle-Ether2.1002
            JOIN
1003  0000.0100.ac00.0001.0a00 (1108:101::100,ff03::3)                 Bundle-Ether1.1003
            JOIN
1003  0011.0200.ac00.0001.0a00 (1108:101::100,ff03::3)                 Bundle-Ether2.1003
            JOIN
1003  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:16:3::1)            Bundle-Ether2.1003
            JOIN
1003  0000.0100.ac00.0001.0a00 (1108:101::100,ff03:123:3::1)           Bundle-Ether1.1003
            JOIN
```

```
1003  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:123:3::1)             Bundle-Ether2.1003
          JOIN
1004  0000.0100.ac00.0001.0a00 (1108:101::100,ff03::4)                   Bundle-Ether1.1004
          JOIN
1004  0011.0200.ac00.0001.0a00 (1108:101::100,ff03::4)                   Bundle-Ether2.1004
          JOIN
1004  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:16:4::1)             Bundle-Ether2.1004
          JOIN
1004  0000.0100.ac00.0001.0a00 (1108:101::100,ff03:123:4::1)            Bundle-Ether1.1004
          JOIN
1004  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:123:4::1)            Bundle-Ether2.1004
          JOIN
1005  0000.0100.ac00.0001.0a00 (1108:101::100,ff03::5)                   Bundle-Ether1.1005
          JOIN
1005   0011.0200.ac00.0001.0a00 (1108:101::100,ff03::5)                     7.7.7.7
             JOIN
1005  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:16:5::1)             Bundle-Ether2.1005
          JOIN
1005  0000.0100.ac00.0001.0a00 (1108:101::100,ff03:123:5::1)            Bundle-Ether1.1005
          JOIN
1005  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:123:5::1)            Bundle-Ether2.1005
          JOIN
1006  0000.0100.ac00.0001.0a00 (1108:101::100,ff03::6)                   Bundle-Ether1.1006
          JOIN
1006  0011.0200.ac00.0001.0a00 (1108:101::100,ff03::6)                   Bundle-Ether2.1006
          JOIN
1006  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:16:6::1)             Bundle-Ether2.1006
          JOIN
1006  0000.0100.ac00.0001.0a00 (1108:101::100,ff03:123:6::1)            Bundle-Ether1.1006
          JOIN
1006   0011.0200.ac00.0001.0a00 (1108:101::100,ff03:123:6::1)               7.7.7.7
             JOIN
1007  0000.0100.ac00.0001.0a00 (1108:101::100,ff03::7)                   Bundle-Ether1.1007
          JOIN
1007  0011.0200.ac00.0001.0a00 (1108:101::100,ff03::7)                   Bundle-Ether2.1007
          JOIN
1007   0011.0200.ac00.0001.0a00 (1108:101::100,ff03:16:7::1)               7.7.7.7
             JOIN
1007  0000.0100.ac00.0001.0a00 (1108:101::100,ff03:123:7::1)            Bundle-Ether1.1007
          JOIN
1007  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:123:7::1)            Bundle-Ether2.1007
          JOIN
1008  0000.0100.ac00.0001.0a00 (1108:101::100,ff03::8)                   Bundle-Ether1.1008
          JOIN
1008   0011.0200.ac00.0001.0a00 (1108:101::100,ff03::8)                     7.7.7.7
             JOIN
1008  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:16:8::1)             Bundle-Ether2.1008
          JOIN
1008  0000.0100.ac00.0001.0a00 (1108:101::100,ff03:123:8::1)            Bundle-Ether1.1008
          JOIN
1008  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:123:8::1)            Bundle-Ether2.1008
          JOIN
1009  0000.0100.ac00.0001.0a00 (1108:101::100,ff03::9)                   Bundle-Ether1.1009
          JOIN
1009  0011.0200.ac00.0001.0a00 (1108:101::100,ff03::9)                   Bundle-Ether2.1009
          JOIN
1009  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:16:9::1)             Bundle-Ether2.1009
          JOIN
1009  0000.0100.ac00.0001.0a00 (1108:101::100,ff03:123:9::1)            Bundle-Ether1.1009
          JOIN
1009  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:123:9::1)            Bundle-Ether2.1009
          JOIN
```

```
1010  0000.0100.ac00.0001.0a00 (1108:101::100,ff03::a)                  Bundle-Ether1.1010
            JOIN
1010  0011.0200.ac00.0001.0a00 (1108:101::100,ff03::a)                  Bundle-Ether2.1010
            JOIN
1010  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:16:a::1)               7.7.7.7
              JOIN
1010  0000.0100.ac00.0001.0a00 (1108:101::100,ff03:123:a::1)            Bundle-Ether1.1010
            JOIN
1010  0011.0200.ac00.0001.0a00 (1108:101::100,ff03:123:a::1)            Bundle-Ether2.1010
            JOIN
```

Multicast IRB

Multicast IRB provides the ability to route multicast packets between a bridge group and a routed interface using a bridge-group virtual interface (BVI). It can be enabled with multicast-routing. THE BVI is a virtual interface within the router that acts like a normal routed interface. For details about BVI, refer *Interface and Hardware Component Configuration Guide for Cisco NCS 540 Series Routers*

BV interfaces are added to the existing VRF routes and integrated with the replication slot mask. After this integration, the traffic coming from a VRF BVI is forwarded to the VPN.

Supported Bridge Port Types

- Bundles
- Satellites
- EFPs (physical, vlans, etc)

Restrictions

- Supported only on Ethernet line cards and enhanced ethernet line cards.
- Support only for IPv4

Example

The CE-PE is collapsed into 1 router (IRB) and IGMP snooping is enabled on the BVIs.

BVI type is included in a multicast VRF. After the BVI slot mask is included in the VRF route slot mask, the traffic from the VRF BVI is forwarded to the VPN/ core.

Access Pseudowire in VPLS Bridge Domains

Table 22: Feature History Table

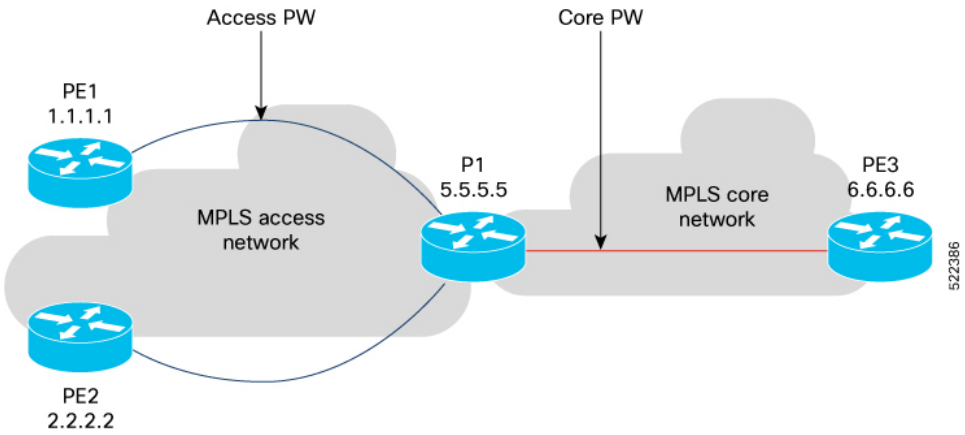
Feature Name	Release Information	Description
Access Pseudowire in VPLS Bridge Domains	Release 7.6.1	You can configure EVPN in the access node under the same bridge domain as EVPN in the core and create a pseudowire (PW) to the nearest PE that binds the access circuits using EVPN. This PW between the access PE and the single-homed PE ensures that the access nodes can leverage the benefits of EVPN.

You can enable VPLS Access Pseudowire in a Bridge Domain (BD) where flooding is enabled.

VPLS is a multipoint Layer 2 VPN technology that connects two or more customer devices using bridging techniques. In scenarios where an L3 multicast route has invalid or incorrect OLEs (Output List Element: a hardware instance of a multicast outgoing interface in a multicast route), instead of dropping the packets, they are sent again to the receiver. If the L3 multicast route already has valid OLE entries apart from the invalid ones, at the receiver end, you can see duplicate packets.

To ensure an uninterrupted flow of packets, the egress traffic management model employs a two-pass model. When you enable access pseudowire, in the two-pass model, at egress, the duplicate IP packet is recycled and gets embedded and egresses from the bundle-ether as OLE.

Following figure shows the interconnection between the provider edge (PE) routers over IP/MPLS networks. The VPLS network requires a bridge domain (Layer 2 broadcast domain) on each PE router. It is responsible for all flooding broadcast frames and multicast replications. The PEs are connected with Pseudowires (PWs).



Limitations

This feature is not supported when IGMP snooping is enabled.

The multicast L3 to L2 traffic is supported only in flood BD configuration.

Configure Access Pseudowire

To enable Access Pseudowire in a VPLS BD, use the following command:

```
Router#configure terminal
Router(config)#hw-module multicast access-pw-enable
```