



# Implementing Layer-3 Multicast Routing

Multicast routing allows a host to send packets to a subset of all hosts as a group transmission rather than to a single host, as in unicast transmission, or to all hosts, as in broadcast transmission. The subset of hosts is known as group members and are identified by a single multicast group address that falls under the IP Class D address range from 224.0.0.0 through 239.255.255.255.

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

The following protocols are supported to implement multicast routing:

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

## Prerequisites for Implementing Multicast Routing

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

## Restrictions

- Multicast over access or core pseudowire is not supported.
- [Restrictions for Implementing Multicast Routing, on page 2](#)
- [Enabling Multicast, on page 2](#)
- [Protocol Independent Multicast, on page 3](#)
- [PIM BFD Overview, on page 3](#)
- [Reverse Path Forwarding , on page 5](#)
- [RPF Vector Encoding Using IETF Standard , on page 6](#)
- [PIM-Source Specific Multicast, on page 6](#)
- [Multicast Source Discovery Protocol, on page 11](#)
- [PIM-Sparse Mode, on page 16](#)

- PIM Bootstrap Router, on page 18
- Designated Routers, on page 19
- Internet Group Management Protocol, on page 21
- IPv6 Multicast for Multiple Sources, on page 24
- Use Case: Video Streaming, on page 24
- Label Switched Multicast (LSM) Multicast Label Distribution Protocol (mLDP) based Multicast VPN (mVPN) Support, on page 26
- Multicast Label Distribution Protocol (MLDP) for Core, on page 28
- Multicast MLDP for Edge Router, on page 29
- Configuration Example for MLDP on Core, on page 30
- Configure VRF MLDP In-Band Signaling on Edge Routers , on page 30
- Configure Global MLDP In-band Signaling on Edge Routers, on page 31
- Configuration Examples for MVPN Profiles, on page 31
- Configuration Examples for Inband mLDP Profiles on Edge Routers, on page 34
- Verification of MLDP Configuration on Edge Routers, on page 35

## Restrictions for Implementing Multicast Routing

- MLD Snooping is not supported.
- PIM SM is not supported for any of the supported MLDP profiles.
- DATA MDT with rate based or policy based switchover is not supported.
- Auto RP is not supported.

## Enabling Multicast

### Configuration Example

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

```
Router#config
Router(config)#multicast-routing
Router(config-mcast)#address-family ipv4
Router(config-mcast-default-ipv4)#interface all enable
*/In the above command, you can also indicate a specific interface (For example, interface
TenGigE0/11/0/0)
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/RP0/cpu0
Interface : FINT0/RP0/cpu0 (Enabled)
SW Mcast pkts in : 0, SW Mcast pkts out : 0
TTL Threshold : 0
Ref Count : 2
Interface : TenGigE0/11/0/0 (Enabled)
SW Mcast pkts in : 0, SW Mcast pkts out : 0
TTL Threshold : 0
Ref Count : 3
Interface : TenGigE0/11/0/1 (Enabled)
SW Mcast pkts in : 0, SW Mcast pkts out : 0
TTL Threshold : 0
Ref Count : 13
Interface : Bundle-Ether1 (Enabled)
SW Mcast pkts in : 0, SW Mcast pkts out : 0
TTL Threshold : 0
Ref Count : 4
Interface : Bundle-Ether1.1 (Enabled)
SW Mcast pkts in : 0, SW Mcast pkts out : 0
TTL Threshold : 0
```

# Protocol Independent Multicast

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

Proper operation of multicast depends on knowing the unicast paths towards a source or an RP. PIM relies on unicast routing protocols to derive this reverse-path forwarding (RPF) information. As the name PIM implies, it functions independently of the unicast protocols being used. PIM relies on the Routing Information Base (RIB) for RPF information. Protocol Independent Multicast (PIM) is designed to send and receive multicast routing updates.

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



**Note** PIM BFD for IPv6 is not supported.

This section describes how you can configure PIM BFD

```
Router# configure
Router(config)# router pim address-family ipv4
Router(config-pim-default-ipv4)# interface HundredGigE0/9/0/0
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 TenGigE0/11/0/0
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
```

### Running Configuration

```
router pim
  address-family ipv4
    interface HundredGigE 0/9/0/0
      bfd minimum-interval 10
      bfd fast-detect
      bfd multiplier 3
    !
  interface TenGigE 0/11/0/0
    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
-----
Hu0/9/0/0          10.12.12.2      0s (0s*0)  90ms (30ms*3)    UP
0/RP0/CPU0
Te0/11/0/0         10.112.112.2    0s (0s*0)  90ms (30ms*3)    UP          Yes    0/RP0/CPU0

```

```
Router# show bfd client
```

```

Name              Node              Num sessions
-----
L2VPN_ATOM        0/RP0/CPU0 0
MPLS-TR           0/RP0/CPU0 0
bgp-default       0/RP0/CPU0 0
bundlemgr_distrib 0/RP0/CPU0 14
isis-1            0/RP0/CPU0 0
object_tracking   0/RP0/CPU0 0
pim6              0/RP0/CPU0 0
pim              0/RP0/CPU0 0
service-layer     0/RP0/CPU0 0

```

## Reverse Path Forwarding

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

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

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

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

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

## 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 TenGigE 0/11/0/0
(config-mcast)# interface TenGigE 0/11/0/1
```

#### 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	TenGigE 0/11/0/0	1w3d	00:01:36	1	B P
<b>25.25.25.2*</b>	<b>TenGigE 0/11/0/0</b>	<b>1w3d</b>	<b>00:01:41</b>	<b>1 (DR)</b>	<b>B P E</b>
32.32.32.2*	TenGigE 0/11/0/1	1w4d	00:01:40	1	B P E
<b>32.32.32.3</b>	<b>TenGigE 0/11/0/1</b>	<b>1w4d</b>	<b>00:01:42</b>	<b>1 (DR)</b>	<b>B P</b>

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

## IGMPv2

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

### Configuring Example

Configures the access-list (mc1):

```
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 mcl
Router(config-igmp)#commit
```

### Running Configuration

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

## 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 TenGigE0/11/0/1.132 are two paths to reach the destination router Turin-56. Since we have enabled multipath option, the source has two IP addresses 50.11.30.12 and 50.11.30.11. The Multicast traffic from two sources take two different paths Bundle-Ether132 and TenGigE0/11/0/1.132 to reach same destination.

This show run output shows that Bundle-Ether132 and TenGigE0/11/0/1.132 are connected to same destination router Turin-56:

```
Router#show run int TenGigE0/11/0/2.132
interface TenGigE0/1/11/2/3.132
description Connected to Turin-56 ten0/11/0/1.132
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
HundredGigE0/9/0/3.1130 Flags: A, Up: 4d15h
Outgoing Interface List
TenGigE0/11/0/6 Flags: F NS, Up: 4d15h
TenGigE0/11/0/6/3.132 Flags: F NS, Up: 4d15h
TenGigE0/11/0/1.122 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
HundredGigE0/9/0/1.1130 Flags: A, Up: 4d15h
Outgoing Interface List
Bundle-Ether121 Flags: F NS, Up: 4d15h
Bundle-Ether132 Flags: F NS, Up: 4d15h
TenGigE0/11/0/6.117 Flags: F NS, Up: 4d15h

```

## Configuring PIM-SSM

### Configuration Example

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

```

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

```

### Running Configuration

```

Router#show running multicast-routing
multicast-routing
  address-family ipv4
    ssm range 4
  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 TenGigE0/11/0/1

```

```
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 TenGigE0/11/0/1
    dr-priority 4
    hello-interval 45
```

### Verification

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

```
Router#show pim interface te0/11/0/1
PIM interfaces in VRF default
Address          Interface          PIM  Nbr  Hello  DR    DR Count Intvl
  Prior
100.1.1.1        TenGigE0/11/0/1  on   1    45    4     this system
```

## Multicast Source Discovery Protocol

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

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

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




---

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

---

## Interconnecting PIM-SM Domains with MSDP

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

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

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

### Before you begin

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

## SUMMARY STEPS

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

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>configure</b>	
<b>Step 2</b>	<b>interface</b> <i>type interface-path-id</i> <b>Example:</b> <pre>RP/0/RP0/CPU0:router(config)# interface loopback 0</pre>	(Optional) Enters interface configuration mode to define the IPv4 address for the interface. <b>Note</b> This step is required if you specify an interface type and number whose primary address becomes the source IP address for the TCP connection.
<b>Step 3</b>	<b>ipv4 address</b> <i>address mask</i> <b>Example:</b> <pre>RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.0.1.3 255.255.255.0</pre>	(Optional) Defines the IPv4 address for the interface. <b>Note</b> 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 <b>connect-source</b> command.

	Command or Action	Purpose
<b>Step 4</b>	<b>exit</b> <b>Example:</b> RP/0/RP0/CPU0:router(config-if)# end	Exits interface configuration mode.
<b>Step 5</b>	<b>router msdp</b> <b>Example:</b> RP/0/RP0/CPU0:router(config)# router msdp	Enters MSDP protocol configuration mode.
<b>Step 6</b>	<b>default-peer ip-address [prefix-list list]</b> <b>Example:</b> RP/0/RP0/CPU0:router(config-msdp)# default-peer 172.23.16.0	(Optional) Defines a default peer from which to accept all MSDP SA messages.
<b>Step 7</b>	<b>originator-id type interface-path-id</b> <b>Example:</b> 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.
<b>Step 8</b>	<b>peer peer-address</b> <b>Example:</b> RP/0/RP0/CPU0:router(config-msdp)# peer 172.31.1.2	Enters MSDP peer configuration mode and configures an MSDP peer. <ul style="list-style-type: none"> <li>• Configure the router as a BGP neighbor.</li> <li>• 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.</li> </ul>
<b>Step 9</b>	<b>connect-source type interface-path-id</b> <b>Example:</b> RP/0/RP0/CPU0:router(config-msdp-peer)# connect-source loopback 0	(Optional) Configures a source address used for an MSDP connection.
<b>Step 10</b>	<b>mesh-group name</b> <b>Example:</b> RP/0/RP0/CPU0:router(config-msdp-peer)# mesh-group internal	(Optional) Configures an MSDP peer to be a member of a mesh group.

	Command or Action	Purpose
<b>Step 11</b>	<b>remote-as</b> <i>as-number</i> <b>Example:</b> <pre>RP/0/RP0/CPU0:router(config-msdp-peer)# remote-as 250</pre>	(Optional) Configures the remote autonomous system number of this peer.
<b>Step 12</b>	<b>commit</b>	
<b>Step 13</b>	<b>show msdp [ipv4] globals</b> <b>Example:</b> <pre>RP/0/RP0/CPU0:router# show msdp globals</pre>	Displays the MSDP global variables.
<b>Step 14</b>	<b>show msdp [ipv4] peer [peer-address]</b> <b>Example:</b> <pre>RP/0/RP0/CPU0:router# show msdp peer 172.31.1.2</pre>	Displays information about the MSDP peer.
<b>Step 15</b>	<b>show msdp [ipv4] rpf rpf-address</b> <b>Example:</b> <pre>RP/0/RP0/CPU0:router# show msdp rpf 172.16.10.13</pre>	Displays the RPF lookup.

## Controlling Source Information on MSDP Peer Routers

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

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

When forwarding SA messages you can do the following:

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

When receiving SA messages you can do the following:

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

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

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

## SUMMARY STEPS

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

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>configure</b>	
Step 2	<b>router msdp</b>  <b>Example:</b>  RP/0/RP0/CPU0:router(config)# router msdp	Enters MSDP protocol configuration mode.
Step 3	<b>sa-filter</b> {in   out} [ <b>list</b> <i>access-list-name</i> ] [ <b>rp-list</b> <i>access-list-name</i> ]  <b>Example:</b>  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. <ul style="list-style-type: none"> <li>• If you specify both the <b>list</b> and <b>rp-list</b> keywords, all conditions must be true to pass any source, group (S, G) pairs in outgoing Source-Active (SA) messages.</li> <li>• You must configure the <b>ipv4 access-list</b> command in <a href="#">Step 7, on page 16</a>.</li> <li>• If all match criteria are true, a <b>permit</b> from the route map passes routes through the filter. A <b>deny</b> filters routes.</li> </ul>
Step 4	<b>cache-sa-state</b> [ <b>list</b> <i>access-list-name</i> ] [ <b>rp-list</b> <i>access-list-name</i> ]  <b>Example:</b>  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.

	Command or Action	Purpose
<b>Step 5</b>	<p><b>t</b>tl-threshold <i>t</i>tl-value</p> <p><b>Example:</b></p> <pre>RP/0/RP0/CPU0:router(config-msdp)# ttl-threshold 8</pre>	<p>(Optional) Limits which multicast data is sent in SA messages to an MSDP peer.</p> <ul style="list-style-type: none"> <li>• Only multicast packets with an IP header TTL greater than or equal to the <i>t</i>tl-value argument are sent to the MSDP peer specified by the IP address or name.</li> <li>• 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.</li> <li>• This example configures a TTL threshold of eight hops.</li> </ul>
<b>Step 6</b>	<p><b>e</b>xit</p> <p><b>Example:</b></p> <pre>RP/0/RP0/CPU0:router(config-msdp)# exit</pre>	Exits the current configuration mode.
<b>Step 7</b>	<p><b>i</b>pv4 access-list <i>n</i>ame [<i>s</i>equen<i>c</i>e-<i>n</i>umber] <b>p</b>ermit <i>s</i>ource [<i>s</i>ource-wildcard]</p> <p><b>Example:</b></p> <pre>RP/0/RP0/CPU0:router(config)# ipv4 access-list 100 20 permit 239.1.1.1 0.0.0.0</pre>	<p>Defines an IPv4 access list to be used by SA filtering.</p> <ul style="list-style-type: none"> <li>• In this example, the access list 100 permits multicast group 239.1.1.1.</li> <li>• The <b>ipv4 access-list</b> command is required if the keyword <b>list</b> is configured for SA filtering in <a href="#">Step 3, on page 15</a>.</li> </ul>
<b>Step 8</b>	<b>c</b> ommit	

## PIM-Sparse Mode

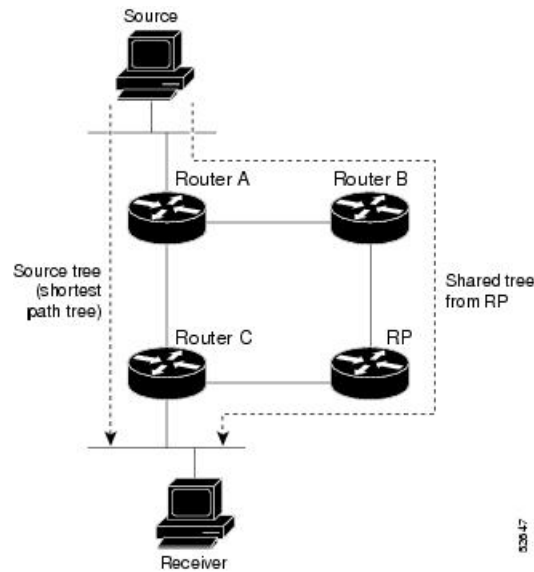
Typically, PIM in sparse mode (PIM-SM) operation is used in a multicast network when relatively few routers are involved in each multicast. Routers do not forward multicast packets for a group, unless there is an explicit request for traffic. Requests are accomplished using PIM join messages, which are 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).

---

## 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]
```

## Designated Routers

Cisco routers use PIM 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 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.

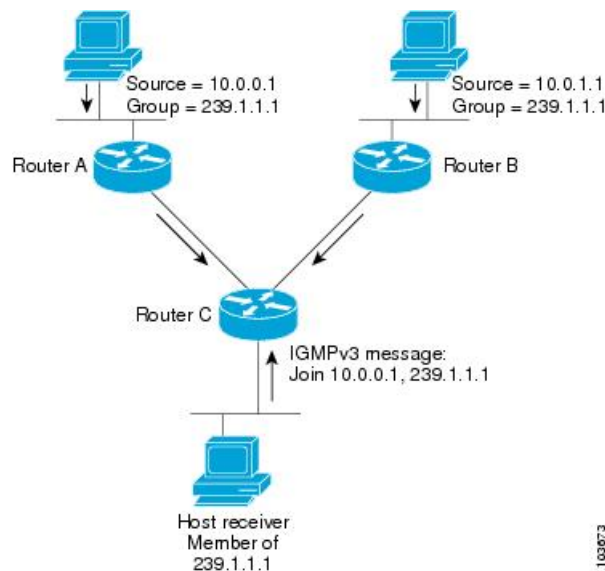


**Note** DR election process is required only on multi access LANs. The last-hop router directly connected to the host is the DR.

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 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 `mcast-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 0/11/0/1, 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 TenGigE0/11/0/1
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 TenGigE0/11/0/1
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    TenGigE0/11/0/1  on  1    45  4    this system

```

## Internet Group Management Protocol

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

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

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



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

### Restrictions

IGMP snooping under VPLS bridge domain is not supported.

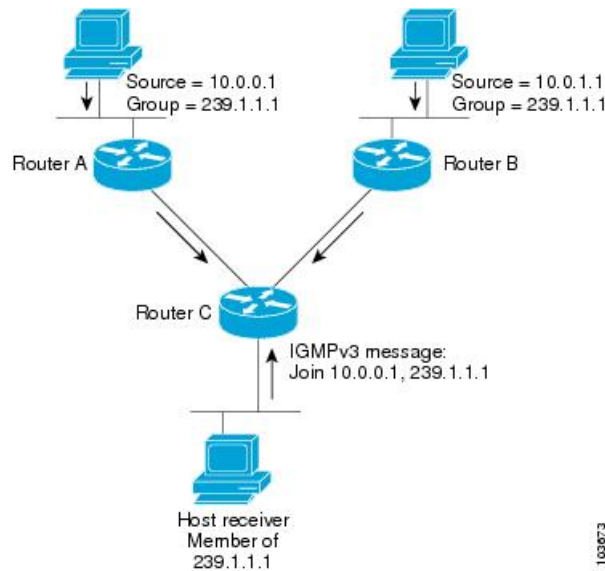
### Functioning of IGMP Routing

The following image "IGMP Singaling" , illustrates two sources, 10.0.0.1 and 10.0.1.1, that are multicasting to group 239.1.1.1.

The receiver wants to receive traffic addressed to group 239.1.1.1 from source 10.0.0.1 but not from source 10.0.1.1.

The host must send an IGMPv3 message containing a list of sources and groups (S, G) that it wants to join and a list of sources and groups (S, G) that it wants to leave. Router C can now use this information to prune traffic from Source 10.0.1.1 so that only Source 10.0.0.1 traffic is being delivered to Router C.

Figure 3: IGMP Signaling



## Configuring Maximum IGMP Per Interface Group Limit

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

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

### Configuration Example

Configures all interfaces with 4000 maximum groups per interface except TenGigE interface 0/11/0/0, which is set to 3000:

```
Router#config
Router(config)#router igmp
Router(config-igmp)#maximum groups-per-interface 4000
Router(config-igmp)#interface TenGigE0/11/0/0
Router(config-igmp-default-if)#maximum groups-per-interface 3000
Router(config-igmp-default-if)#commit
```

### Running Configuration

```
router igmp
 interface TenGigE0/11/0/0
   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
TenGigE0/11/0/0    5       4000
TenGigE0/11/0/1    5       4000
TenGigE0/11/0/2    0       4000
TenGigE0/11/0/3    5       4000
TenGigE0/11/0/4    5       3000
TenGigE0/11/0/5    5       4000
TenGigE0/11/0/6    5       4000
TenGigE0/11/0/6.1  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 TenGigE0/11/0/3
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 TenGigE0/11/0/3
 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
  TenGigE0/11/0/3 Flags: F NS LI, Up: 00:01:11

```

# IPv6 Multicast for Multiple Sources

## Use Case: Video Streaming

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

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

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



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

**Figure 4: Video Streaming**

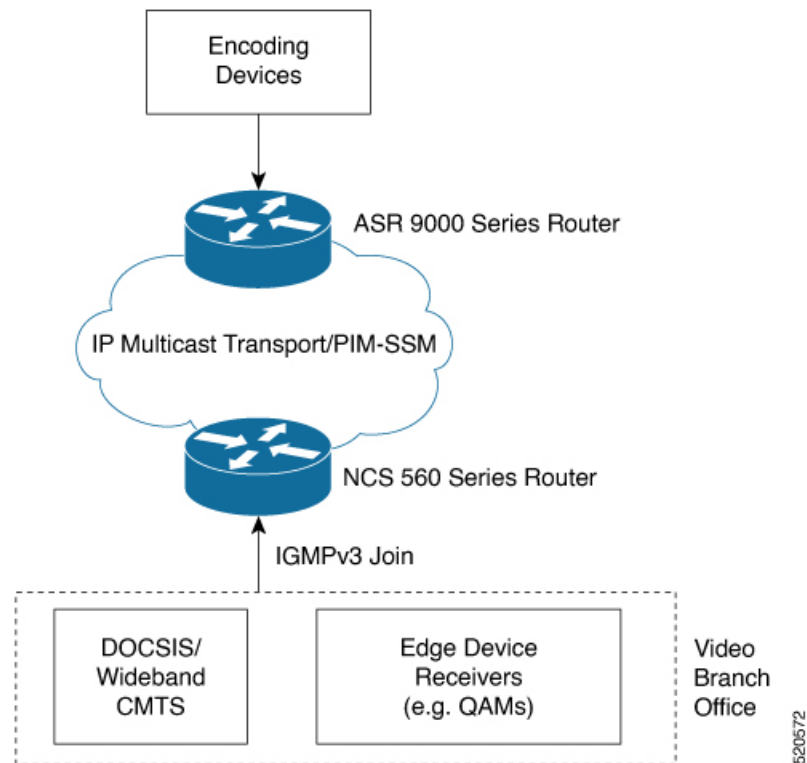
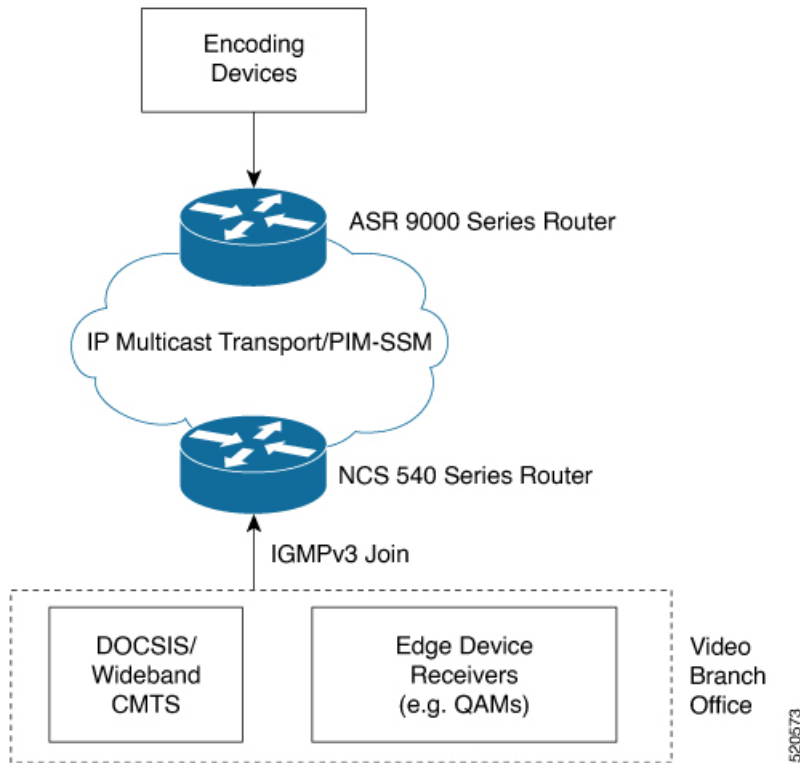


Figure 5: Video Streaming



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

Table 1: Feature History Table

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

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

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



**Note** IPv6 is not supported for profile 10. Also IPv4 SM is not supported for mLDP profiles on an edge router.

## Benefits of LSM MLDP based MVPN

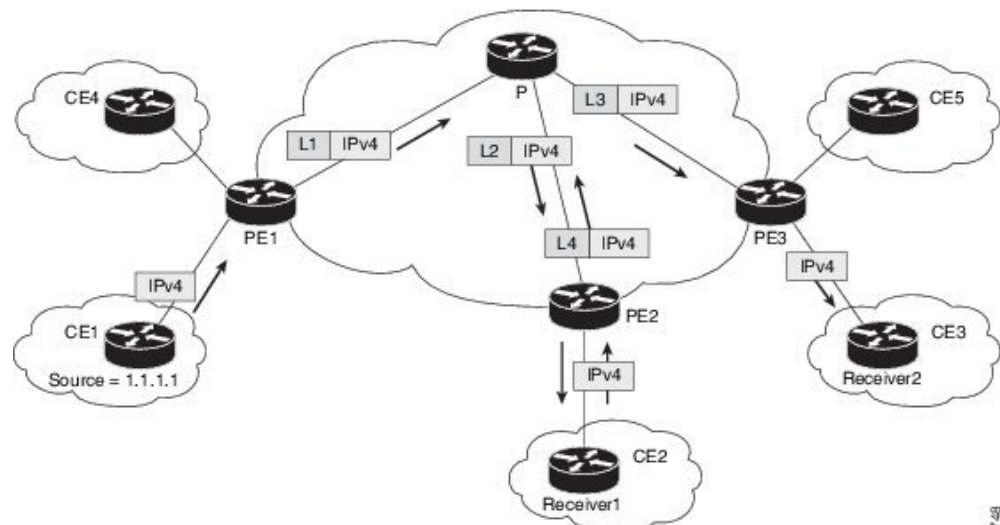
LSM provides these benefits when compared to GRE core tunnels that are currently used to transport customer traffic in the core:

- 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.

**Figure 6: MLDP based MPLS Network for Core and Edge Routers**



## Packet Flow in mLDP-based Multicast VPN

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

## Realizing a mLDP-based Multicast VPN

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

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

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

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

## Restrictions for mLDP on Edge Routers

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

- NETCONF/YANG on MVPN for Profile 6 and Profile 7 is not supported.
- MLDP ping traceroute is not supported.
- IPv6 BVI is not supported.
- Netflow for MPLS-encapsulated multicast packets is not supported.
- MLDP Fast-Reroute is not supported.

## Multicast Label Distribution Protocol (MLDP) for Core

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

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

## Characteristics of MLDP Profiles on Core

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

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

- Profile 12—Default MDT - MLDP - P2MP - BGP-AD - BGP C-mcast Signaling
- Profile 17—Default MDT - MLDP - P2MP - BGP-AD - PIM C-mcast Signaling

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

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

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

## Multicast MLDP for Edge Router

The following MLDP and P2MP-TE 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 8—Global Static - P2MP-TE
- Profile 10—VRF Static - P2MP TE - BGP-AD
- Profile 14—MLDP Partitioned MDT P2MP with BGP AD and BGP-C Multicast Signaling

## P2MP-TE and MLDP Scale Number

The following table lists the scaling numbers for MLDP and P2MP-TE:

Feature	Scale
OLE Scale (Egress replication) per P2MP tree	250 (with packet size of 128)
Number of P2MP Trees	100
Maximum (S,G) flows	00 (S,G) flows distributed among max 100 VRFs
Maximum OLE scale per system	64 * 250 = 16000
VLAN interfaces in Egress	400
Number of Active ISIS interfaces	400
Number of ISIS Peers	400
Number of iBGP v4 Peers	400
Number of iBGP v6 Peers	400

## Configuration Example for MLDP on Core

```

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

```

## 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 0/11/0/1
RP/0/RP0/CPU0:router(config-pim-ipv4-if)#enable
```

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

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

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

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

*/\* Enable MPLS MLDP \*/*

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

## Configuration Examples for MVPN Profiles

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

### Configuration Examples for Inband mLDP Profiles

Profile-6: VRF Inband mLDP

```
router bgp 100
  mvpn
  !
  multicast-routing
```

```

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

```

#### Profile-7: Global Inband mLDP

```

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

```

## Configuration Examples for Rosen mLDP profiles

#### Profile-12: Rosen mLDP P2MP with BGP-AD and BGP signaling

```

!
multicast-routing
  mdt source Loopback0
  vrf v121

```



```

    address-family ipv4
      mdt mtu 1600
    mdt default mldp p2mp
    mdt data 255
      interface all enable
      bgp auto-discovery mldp
    !
    address-family ipv6
      mdt mtu 1600
    mdt default mldp p2mp
    mdt data 255
      interface all enable
      bgp auto-discovery mldp
    !
  !
router pim
  vrf v121
    address-family ipv4
      rpf topology route-policy rosen-mldp
      mdt c-multicast-routing bgp
    !
    address-family ipv6
      rpf topology route-policy rosen-mldp
      mdt c-multicast-routing bgp
    !
  !
  route-policy rosen-mldp
    set core-tree mldp-default
  end-policy
  !

```

#### Profile 17 : Default MDT - MLDP - P2MP - BGP-AD - PIM C-Mcast Signaling

```

vrf one
  vpn id 1:1
  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
    !
    interface GigabitEthernet0/1/0/0
      enable
    !
  !
  !
  route-policy rpf-for-one
    set core-tree mldp-default
  end-policy
  !
multicast-routing
  vrf one
    address-family ipv4
      mdt source Loopback0

```

```

mdt default mldp p2mp
mdt data 100
rate-per-route
interface all enable
bgp auto-discovery mldp
!
accounting per-prefix
!
!
!

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

## 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 TenGigE0/11/0/1
  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          TenGigE0/11/0/1 LDP
  Adj list        : 12.1.1.2          TenGigE0/11/0/1
  Peer addr list  : 2.25.32.2
                  : 2.2.2.2
                  : 11.1.1.1
                  : 12.1.1.2
                  : 13.10.1.1

```

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

```

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

LSP-ID: 0x00001 Paths: 7 Flags:
0x00001 P2MP 5.5.5.5 [vpn6 1:1 2015:1:1::3 ff3e::1]
  Local Label: 70009
  Remote Label: 64018 NH: 12.1.1.2 Inft: TenGigE0/11/0/1
  Remote Label: 64022 NH: 50.1.1.1 Inft: TenGigE0/11/0/1
  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: TenGigE0/11/0/2
  Remote Label: 64022 NH: 40.1.1.1 Inft: TenGigE0/11/0/3

```

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
```