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.

- Enabling Multicast, on page 2
- Supported Multicast Features, on page 3
- Protocol Independent Multicast, on page 3
- PIM BFD Overview, on page 4
- PIM Bootstrap Router, on page 5
- Reverse Path Forwarding , on page 7
- RPF Vector Encoding Using IETF Standard , on page 8
- PIM-Source Specific Multicast, on page 8
- Configuring PIM Parameters, on page 12
- PIM-Sparse Mode, on page 13
- Multicast Source Discovery Protocol, on page 15
- Internet Group Management Protocol, on page 26
- Multicast Route Statistics , on page 30

- Bundle Member Selection , on page 33
- Multicast Over IPV4 Unicast GRE Tunnels, on page 34
- Use Case: Video Streaming, on page 41
- Multicast Label Distribution Protocol, on page 42
- Label Switching Multicast for Edge Router, on page 43
- Multicast MLDP Profile 14 support on an Edge Router, on page 43
- Label Switched Multicast (LSM) Multicast Label Distribution Protocol (mLDP) based Multicast VPN (mVPN) Support, on page 46
- Configuration Example for MLDP on Core, on page 50
- Overriding VRFs in IGMP Interfaces, on page 51
- Configuring IGMP VRF Override, on page 52
- Configure MVPN using Draft-Rosen (Profile 0), on page 56
- Configure mVPN using RSVP-TE P2MP (Profile 22), on page 68
- Flexible Algorithm for MLDP, on page 70
- Configure VRF MLDP In-Band Signaling on Edge Routers , on page 74
- Configure Global MLDP In-band Signaling on Edge Routers, on page 75
- Restrictions for MVPN Profiles, on page 76
- Configuration Examples for MVPN Profiles, on page 76
- Verification of MLDP Configuration on Edge Routers, on page 77

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
```

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

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                  Yes
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
```

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 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
25.25.25.2*                TenGigE 0/11/0/0       1w3d      00:01:41 1 (DR) B P E
32.32.32.2*                  TenGigE 0/11/0/1
       1w4d      00:01:40 1      B P E
32.32.32.3                   TenGigE 0/11/0/1
       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 TenGigE0/11/0/1.132 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 TenGigE0/11/0/1.132 to reach same destination.

This show run output shows that Bundle-Ether132 and TenGigE0/11/0/1.132 are connected to same destination router Turnin-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
    TenGigE0/1/0/6/3.132 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 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
```

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
```

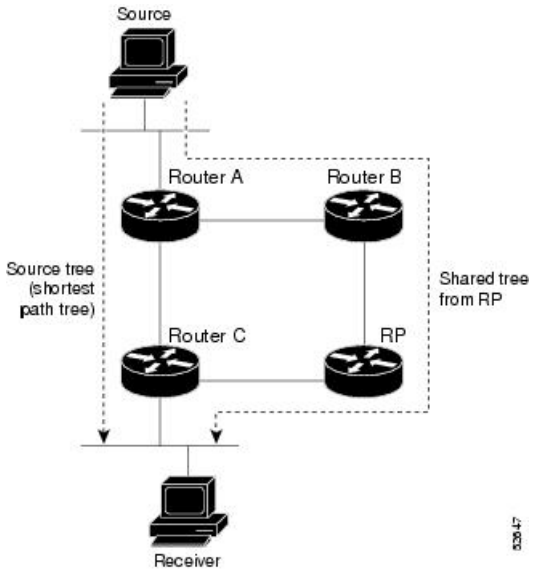
PIM-Sparse Mode

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.

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

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`	(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

	Command or Action	Purpose
	`10.0.1.3 255.255.255.0`	IP address for the TCP connection. See optional for information about configuring the **connect-source** command.
Step 4	**exit** **Example:** `RP/0/RP0/CPU0:router(config-if)# end`	Exits interface configuration mode.
Step 5	**router msdp** **Example:** `RP/0/RP0/CPU0:router(config)# router msdp`	Enters MSDP protocol configuration mode.
Step 6	**default-peer** *ip-address* [**prefix-list** *list*] **Example:** `RP/0/RP0/CPU0:router(config-msdp)# default-peer 172.23.16.0`	(Optional) Defines a default peer from which to accept all MSDP SA messages.
Step 7	**originator-id** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config-msdp)# originator-id /1/1/0`	(Optional) Allows an MSDP speaker that originates a (Source-Active) SA message to use the IP address of the interface as the RP address in the SA message.
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:**	(Optional) Configures an MSDP peer to be a member of a mesh group.

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

Controlling Source Information on MSDP Peer Routers

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

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

When forwarding SA messages you can do the following:

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

When receiving SA messages you can do the following:

- Filter all incoming SA messages from an MSDP peer
- Specify an extended access list to pass certain source/group pairs

- Filter based on match criteria in a route map

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

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

SUMMARY STEPS

1. **configure**
2. **router msdp**
3. **sa-filter** {**in** | **out**} [**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

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**router msdp** **Example:** `RP/0/RP0/CPU0:router(config)# router msdp`	Enters MSDP protocol configuration mode.
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 20. • 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:**	Creates and caches source/group pairs from received Source-Active (SA) messages and controls pairs through access lists.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-msdp)# cache-sa-state list 100`	
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.
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 19.
Step 8	**commit**	

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.

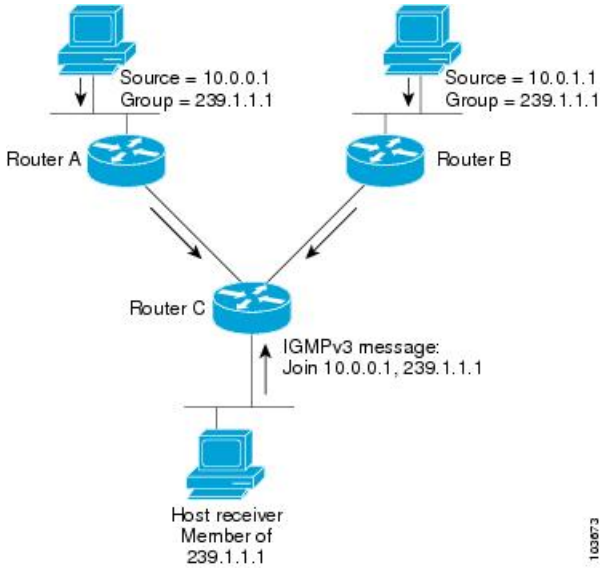
The designated router is responsible for sending PIM register and PIM join and prune messages toward the RP to inform it about host group membership.

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 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 0/11/0/1, 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
```

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.

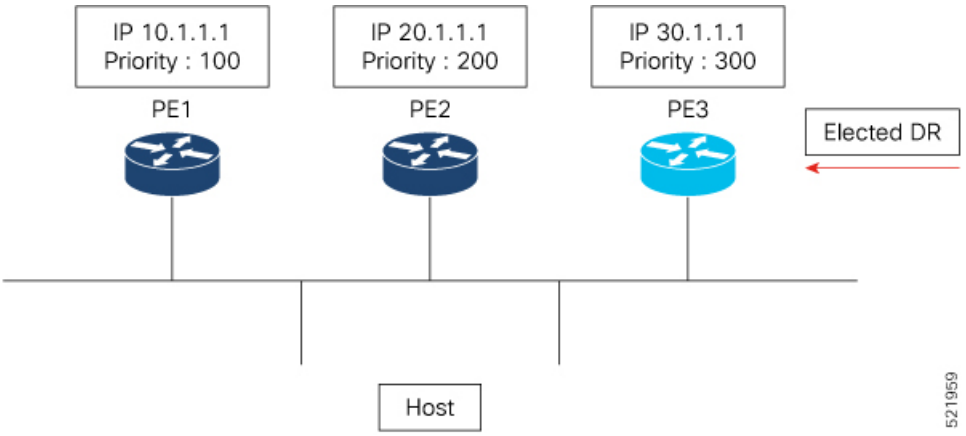
After you enable the sticky DR feature, the elected DR no longer advertises configured DR. Instead the router sends PIM Hello message with special PIM DR priority value which is reserved for Sticky PIM DR.

Restrictions

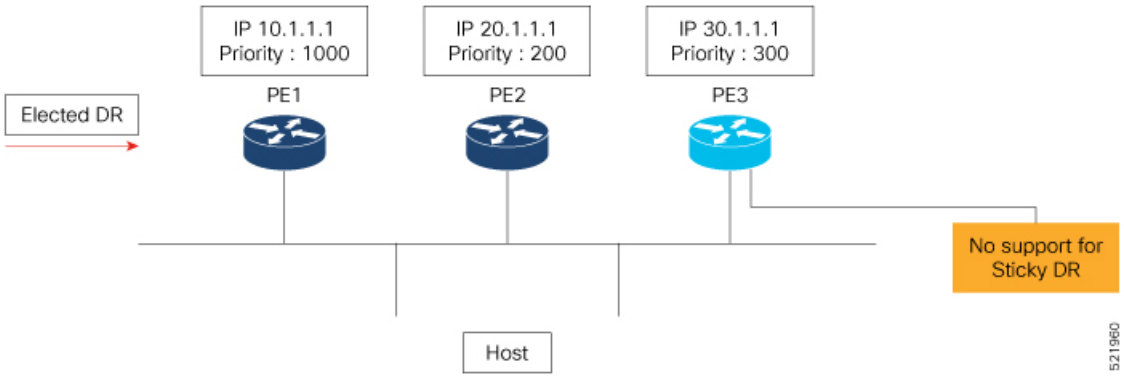
- The Sticky DR priority value is 4294967294. You must not configure DR priority with the value 4294967294 or any number greater than this value.

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.

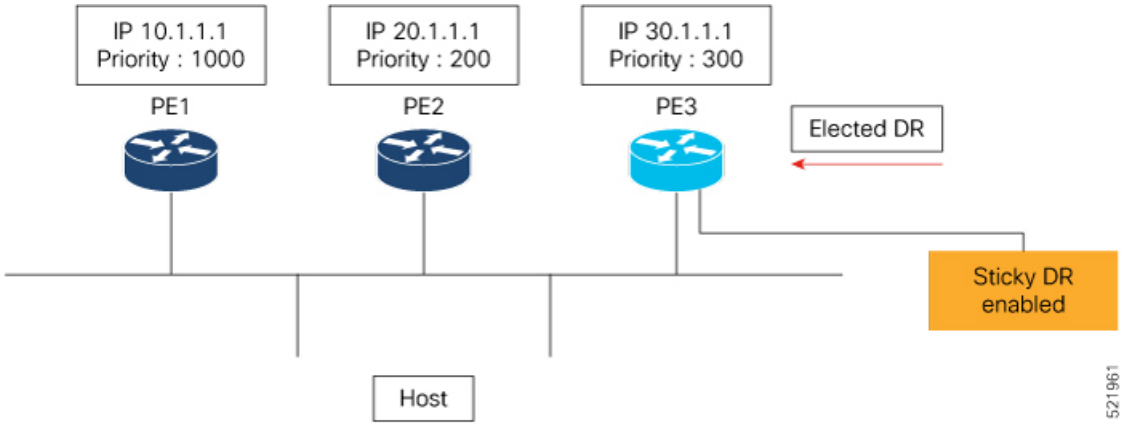


Now, when you configure PE1 with DR priority 1000, DR election process is re-initiated and PE1 becomes the new DR.



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
```

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.

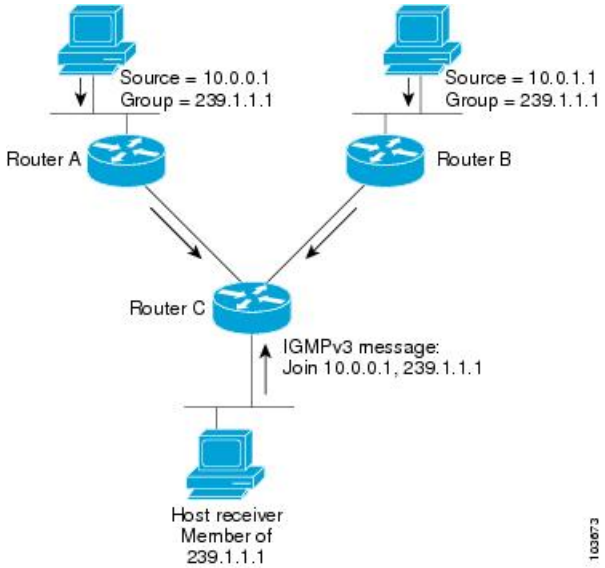
Functioning of IGMP Routing

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
```

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 not supported.
- 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 2: 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.

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
```

Bundle Member Selection

Table 3: 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 4: Feature History Table

Feature Name	Release Information	Feature Description
Support for Multicast Over IPV4 Unicast GRE Tunnels	Release 7.4.1	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.

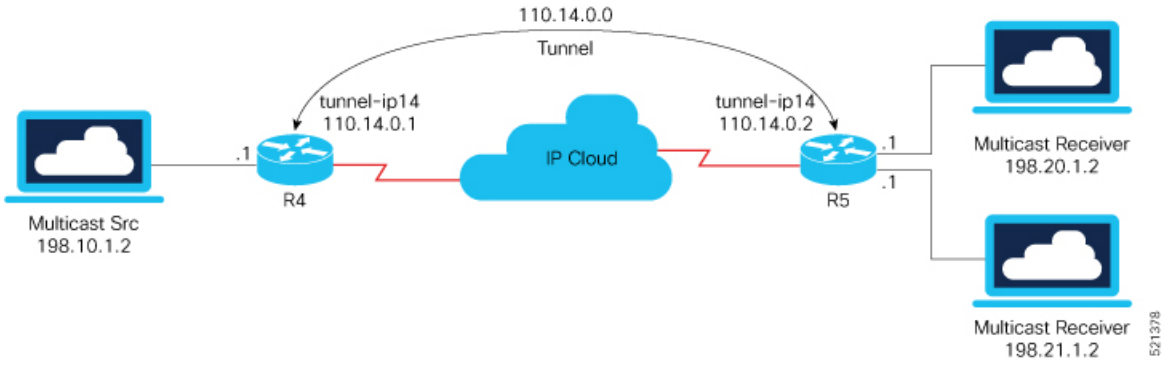
Restrictions

- 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

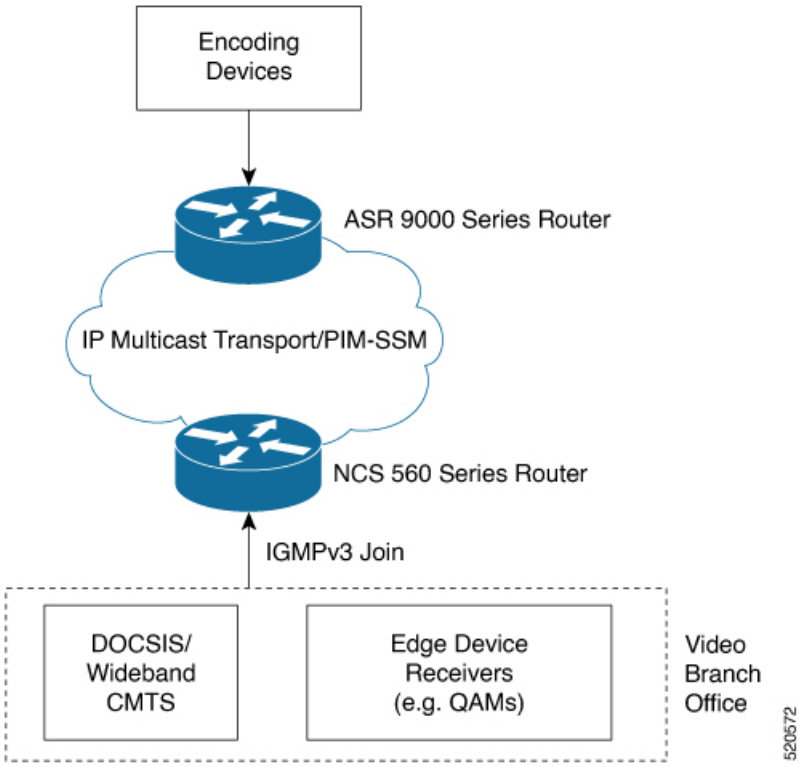
In today's broadcast video networks, proprietary transport systems are used to deliver entire channel line-ups to each video branch office. IP based transport network would be a cost efficient/convenient alternative to deliver video services combined with the delivery of other IP based services. (Internet 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 4: 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 5: Feature History Table

Feature Name	Release Information	Feature Description
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 560 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.

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.

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

For more information about the characteristics of each of the 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.

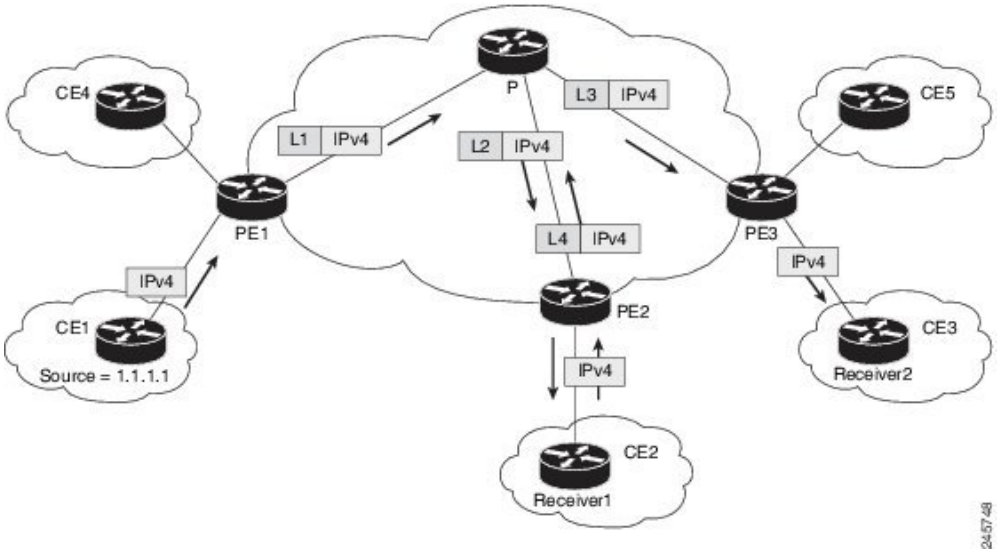
Configuring MLDP MVPN

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 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 5: MLDP based MPLS Network



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 , the label is removed and the IP multicast packet is replicated onto the VRF interface.

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).

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.
- The bud node does not support local receiver functionality for IPv6 multicast traffic when the incoming traffic from the core network includes an extra EXPLICIT NULL label at the Bottom of Stack (BOS) in addition to the MLDP label.

Configuration Process for MLDP MVPN (Intranet)

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

- Enabling MPLS MLDP
  - configure
  - mpls ldp mldp
- Configuring a VRF entry
  - configure
  - vrf vrf_name
  - address-family ipv4/ipv6 unicast
  - import route-target route-target-ext-community
  - export route-target route-target-ext-community

- Configuring VPN ID
  - configure
  - vrf *vrf_name*
  - vpn id *vpn_id*
- Configuring MVPN Routing and Forwarding instance
  - configure
  - multicast-routing vrf *vrf_name*
  - address-family ipv4
  - mdt default mldp ipv4 *root-node*
- Configuring the Route Distinguisher
  - configure
  - router bgp *AS Number*
  - vrf *vrf_name*
  - rd *rd_value*
- Configuring Data MDTs (optional)
  - configure
  - multicast-routing vrf *vrf_name*
  - address-family ipv4
  - mdt data <1-255>
- Configuring BGP MDT address family
  - configure
  - router bgp *AS Number*
  - address-family ipv4 mdt
- Configuring BGP vpnv4 address family
  - configure
  - router bgp *AS Number*
  - address-family vpnv4 unicast
- Configuring BGP IPv4 VRF address family
  - configure
  - router bgp *AS Number*

- vrf *vrf_name*
- address-family ipv4 unicast

- Configuring PIM SM/SSM Mode for the VRFs
  - configure
  - router pim
  - vrf *vrf_name*
  - address-family ipv4
  - rpf topology route-policy *rosen_mvpn_mldp*

  For each profile, a different route-policy is configured.
- Configuring route-policy
  - route-policy *rosen_mvpn_mldp*
  - set core-tree *tree-type*
  - pass
  - end-policy

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

Configuration Example for MLDP on Core

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

Overriding VRFs in IGMP Interfaces

Table 6: 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

SUMMARY STEPS

1. **configure**
2. **vrf** *vrf-name*
3. **address-family ipv4 unicast**
4. **import route-target 1:1**
5. **export route-target 1:1**
6. **commit**

DETAILED STEPS

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:**	Enables VRF import.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-vrf-af)# import route-target 1:1`	
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.

SUMMARY STEPS

1. **configure**
2. **multicast-routing vrf** [*vrf-name* | *default]*
3. **interface** {*type interface-path-id* | **all**} **enable**
4. **commit**

DETAILED STEPS

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

SUMMARY STEPS

1. **configure**
2. **interface** *type interface-path-id*
3. **vrf** *vrf-name*
4. **ipv4 address** *address mask*
5. **commit**

DETAILED STEPS

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.
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

SUMMARY STEPS

1. **configure**
2. **route-policy** *policy-name*
3. **set rpf-topology vrf default**
4. **end-policy**
5. **commit**

DETAILED STEPS

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

SUMMARY STEPS

1. **configure**
2. **router pim vrf** *vrf-name* **address-family ipv4**
3. **rpf-topology route-policy** *policy-name*
4. **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**router pim vrf** *vrf-name* **address-family ipv4**	Enters PIM address-family IPv4 submode.

	Command or Action	Purpose
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 7: Feature History Table

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 8: 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
```

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.

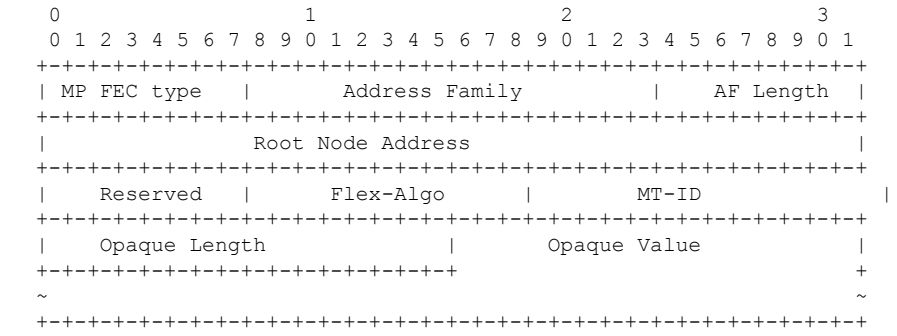
With this feature, you can get disjoint paths for two multicast streams. Each stream carries traffic within a separate network and helps reduce a connection loss or delay ensuring low latency and high availability of multicast traffic. This feature allows you to segregate multicast traffic to specific 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 0/11/0/1
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
```

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 on Core Routers

Profile-6: VRF Inband mLDP

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

Profile-7: Global Inband mLDP

```
multicast-routing
 address-family ipv4
   mdt source Loopback0
   mdt mldp in-band-signaling ipv4
   ssm range Global-SSM-Group
  interface all enable
```

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

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 [vpnv6 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
```