MVPN mLDP Partitioned MDT

The MVPN mLDP partitioned MDT feature uses Upstream Multicast Hop-Provider Multicast Service Interface (UMS-PMSI), a subset of provider edge routers (PEs) to transmit data to other PEs; similar to the usage of multiple selective-PMSI (S-PMSI) by data multicast distribution tree (MDT). In the partitioned MDT approach, egress PE routers that have interested receivers for traffic from a particular ingress PE joins a point-to-point (P2P) connection rooted at that ingress PE. This makes the number of ingress PE routers in a network to be low resulting in a limited number of trees in the core.

Prerequisites for MVPN mLDP Partitioned MDT

MVPN BGP auto discovery should be configured.

Restrictions for MVPN mLDP Partitioned MDT

- PIM Dense mode (except for Auto-RP) and PIM-Bidir in the VRF are not supported.
- BGP multicast signaling is supported and PIM signaling is not supported.
- Only point-to-multi point (P2MP) mLDP label switch path is supported.
- Same VRF (for which mLDP in-band signaling is configured) needs to be configured on IPv4 and IPv6 address families.
- mLDP Partitioned multicast distribution tree (MDT) supports PIM-Source Specific Multicast (SSM) traffic only.
- Rosen mLDP recursive FEC is not supported. Partitioned MDT is applicable to inter-AS VPN (Inter AS option B and option C are not supported).
- mLDP filtering is not supported.
- Only interface-based strict RPF is supported with partitioned MDT.

- The **strict-rpf interface** command is *not* supported.
- For mLDP Partitioned multicast distribution tree (MDT) to work with PIM-Sparse Mode (SM) traffic, configure only a single ingress PE and ensure that the **strict-rpf interface** command is disabled. Configuring multiple PE ingress is not allowed.

Information About MVPN mLDP Partitioned MDT

Overview of MVPN mLDP Partitioned MDT

MVPN allows a service provider to configure and support multicast traffic in an MPLS VPN environment. This type supports routing and forwarding of multicast packets for each individual VPN routing and forwarding (VRF) instance, and it also provides a mechanism to transport VPN multicast packets across the service provider backbone. In the MLDP case, the regular label switch path forwarding is used, so core does not need to run PIM protocol. In this scenario, the c-packets are encapsulated in the MPLS labels and forwarding is based on the MPLS Label Switched Paths (LSPs).

The MVPN mLDP service allows you to build a Protocol Independent Multicast (PIM) domain that has sources and receivers located in different sites.

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

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

Separate multicast routing and forwarding tables are maintained for each VPN routing and forwarding (VRF) instance, with traffic being sent through VPN tunnels across the service provider backbone.

In the Rosen MVPN mLDP solution, a multipoint-to-multipoint (MP2MP) default MDT is setup to carry control plane and data traffic. A disadvantage with this solution is that all PE routers that are part of the MVPN need to join this default MDT tree. Setting up a MP2MP tree between all PE routers of a MVPN is equivalent to creating N P2MP trees rooted at each PE (Where N is the number of PE routers). In an Inter-AS (Option A) solution this problem is exacerbated since all PE routers across all AS'es need to join the default MDT. Another disadvantage of this solution is that any packet sent through a default MDT reaches all the PE routers even if there is no requirement.

In the partitioned MDT approach, only those egress PE routers that receive traffic requests from a particular ingress PE join the PMSI configured at that ingress PE. This makes the number of ingress PE routers in a network to be low resulting in a limited number of trees in the core.

How to Configure MVPN mLDP Partitioned MDT

Configuring MVPN mLDP Partitioned MDT

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast-routing vrf** *vrf-name* **Example:** `Device(config)# ip multicast-routing vrf VRF`	Enables IP multicast routing for the MVPN VRF specified for the *vrf-name* argument.
Step 4	**ip vrf** *vrf-name* **Example:** `Device(config-vrf)# ip vrf VRF`	Defines a VRF instance and enters VRF configuration mode.
Step 5	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 50:11`	Creates a route distinguisher (RD) (in order to make the VRF functional). • Creates the routing and forwarding tables, associates the RD with the VRF instance, and specifies the default RD for a VPN.
Step 6	**route target export** *route-target-ext-community* **Example:** `Device(config-vrf)# route target export 100:100`	Creates an export route target extended community for the specified VRF.
Step 7	**route target import** *route-target-ext-community* **Example:**	Creates an import route target extended community for the specified VRF.

	Command or Action	Purpose
	`Device(config-vrf)# route target import 100:100`	
Step 8	**mdt partitioned mldp p2mp** **Example:** `Device(config-vrf)# mdt partitioned mldp p2mp`	Configures partitioned MDT. • If both IPv4 and IPv6 address-families need to be configured for partitioned MDT, configure this command under both the VRF address-family sub-modes.
Step 9	**mdt auto-discovery mldp** [**inter-as**] **Example:** `Device(config-vrf)# mdt auto-discovery mldp inter-as`	Enables inter-AS operation with BGP A-D.
Step 10	**exit** **Example:** `Device(config-vrf)# exit`	Exits the VRF configuration mode and returns to privileged EXEC mode.
Step 11	**show ip pim mdt** **Example:** `Device# show ip pim mdt`	Displays information on wildcard S-PMSI A-D route.
Step 12	**show ip pim vrf mdt** [**send** | **receive**] **Example:** `Device# show ip pim vrf mdt send`	Displays information on wildcard S-PMSI A-D route along with MDT group mappings received from other PE routers or the MDT groups that are currently in use.
Step 13	**show ip multicast mpls vif** **Example:** `Device# end`	Displays the LSPVIFs created for all the PEs.

Configuration Examples for MVPN mLDP Partitioned MDT

Example: MVPN mLDP Partitioned MDT

```
!
vrf definition cu1
 rd 1:1
 vpn id 1:1
 !
```

```
 address-family ipv4
  mdt auto-discovery mldp
  mdt partitioned mldp p2mp
  mdt data mpls mldp 1
  mdt overlay use-bgp
  route-target export 1:1
  route-target import 1:1
 exit-address-family
!
ip multicast-routing distributed
ip multicast-routing vrf cu1 distributed
!
mpls label protocol ldp
mpls ldp session protection
mpls ldp igp sync holddown 10000
mpls ldp discovery targeted-hello accept
no mpls mldp forwarding recursive
mpls mldp path traffic-eng
mpls traffic-eng tunnels
mpls traffic-eng auto-tunnel backup nhop-only
mpls traffic-eng auto-tunnel primary onehop

!
redundancy
 mode sso
bridge-domain 1
!
!
!
interface Loopback0
 ip address 10.10.10.1 255.255.255.255
 ip ospf 100 area 0
 load-interval 30
!
interface Loopback1
 vrf forwarding cu1
 ip address 11.11.11.1 255.255.255.0
 ip pim sparse-mode
 load-interval 30
!
!
interface GigabitEthernet0/3/0
 ip address 13.0.0.1 255.255.255.0
 ip ospf 100 area 0
 negotiation auto
 mpls ip
 mpls label protocol ldp
 mpls traffic-eng tunnels
 cdp enable
 ip rsvp bandwidth
!
interface GigabitEthernet0/3/4
 no ip address
 negotiation auto
 service instance 1 ethernet
  encapsulation dot1q 1
  rewrite ingress tag pop 1 symmetric
  bridge-domain 1
!
interface GigabitEthernet0/4/1
 ip address 12.0.0.1 255.255.255.0
 ip ospf 100 area 0
 load-interval 30
 negotiation auto
```

```
 mpls ip
 mpls label protocol ldp
 mpls traffic-eng tunnels
 cdp enable
 ip rsvp bandwidth
!
interface BDI1
 vrf forwarding cu1
 ip address 11.0.1.1 255.255.255.0
 ip pim sparse-mode
 load-interval 30
!
router ospf 100
 router-id 10.10.10.1
 fast-reroute per-prefix enable prefix-priority low
 timers throttle spf 50 200 5000
 timers throttle lsa 50 200 5000
 timers lsa arrival 100
 network 10.0.0.1 0.0.0.0 area 0
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 mpls traffic-eng multicast-intact
!
router bgp 100
 bgp log-neighbor-changes
 neighbor 10.10.10.2 remote-as 100
 neighbor 10.10.10.2 update-source Loopback0
 neighbor 10.10.10.3 remote-as 100
 neighbor 10.10.10.3 update-source Loopback0
 !
 address-family ipv4
  redistribute connected
  neighbor 10.10.10.2 activate
  neighbor 10.10.10.2 send-community extended
  neighbor 10.10.10.3 activate
  neighbor 10.10.10.3 send-community extended
 exit-address-family
 !
 address-family ipv4 mvpn
  neighbor 10.10.10.2 activate
  neighbor 10.10.10.2 send-community extended
  neighbor 10.10.10.3 activate
  neighbor 10.10.10.3 send-community extended
 exit-address-family
 !
 address-family vpnv4
  neighbor 10.10.10.2 activate
  neighbor 10.10.10.2 send-community extended
  neighbor 10.10.10.3 activate
  neighbor 10.10.10.3 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf cu1
  redistribute connected
 exit-address-family
!
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
ip pim vrf cu1 rp-address 11.11.11.1
!
```