

# CCIE Service Provider Lab Workbook v4.0

(<http://labs.ine.com/workbook/toc/service-provider-v4>) »

## CCIE SP v4 Advanced Technology Labs - Services

### Multicast VPNs - PIM Core Tree

« [LISP - IPv6 over IPv4 core \(/workbook/view/service-provider-v4/task/lisp-ipv6-over-ipv4-core-Mjk1Mg%3D%3D\)](/workbook/view/service-provider-v4/task/lisp-ipv6-over-ipv4-core-Mjk1Mg%3D%3D) | [Multicast VPNs - mLDP Core Tree \(/workbook/view/service-provider-v4/task/multicast-vpns-mldp-core-tree-Mjk0MA%3D%3D\)](/workbook/view/service-provider-v4/task/multicast-vpns-mldp-core-tree-Mjk0MA%3D%3D) »

Last updated: April 22, 2016

#### Note:

**Initial Configuration & Diagrams:** Load the initial configuration files for the section named **Multicast**, which can be found in [CCIE SPv4 Topology Diagrams & Initial Configurations \(http://labs.ine.com/workbook/view/service-provider-v4/task/ccie-spv4-topology-diagrams-initial-configs\)](http://labs.ine.com/workbook/view/service-provider-v4/task/ccie-spv4-topology-diagrams-initial-configs). Refer to the **Multicast Diagram** in order to complete this task.

**A Note Multicast Platform Support:** Multicast data-plane forwarding is not supported in the current implementations of XRv. The control-plane portion is fully functional however. Throughout the Multicast sections of this workbook, we will configure the different variations of Multicast on the XRv platform to explore IOS-XR specific syntax, and to validate the protocol mechanics via the control-plane state.

CONTENTS >

#### Task

- R2 and XR1 are preconfigured as PE routers for Unicast MPLS L3VPN service to VRF VPN\_A. Configure the network as defined below to also provide Multicast transport services to this customer.
- Configure Multicast Routing in the Service Provider Core as follows:
  - Enable PIM in Sparse Mode on all transit interfaces running IGP in the core, along with the Loopback0 interfaces of R2 and XR1.
  - Enable support for Source Specific Multicast using the group range 232.0.0.0/8 in the core.
  - Activate the MDT address-family for the BGP peering between the PE routers R2 and XR1.
  - Configure VRF VPN\_A on R2 and XR1 to use the default Multicast Distribution Tree (MDT) address 232.0.0.1, and the data MDT address 232.0.1.0/24.
- Configure the Multicast Routing in the Customer sites as follows:
  - Enable PIM in Sparse Mode on all transit interfaces of R1 & XR2, and on the CE facing links of R2 and XR1.
  - Enable PIM in Sparse Mode on the Loopback0 interface of XR2.
  - Configure XR2 to announce itself as the BSR and RP Candidate for all groups in the range 224.0.0.0/4.
  - Configure R7's link to R1 to generate an IGMP Report message for the group 227.7.7.7.
  - Configure R8's link to XR2 to generate an IGMP Report message for the group 228.8.8.8.
- When complete, you should be able to perform the following verifications:
  - R1 should have built a (\*,G) tree for the group (\*,227.7.7.7) back to the Rendezvous Point XR2.
  - XR2 should have a (\*,G) tree built for the group (\*,228.8.8.8) to its attached receiver, R8.
  - R7 should be able to ping the multicast group address 228.8.8.8, and have packets forwarded out to the receiver R8.

- R8 should be able to ping the multicast group address 227.7.7.7, and have packets forwarded out to the receiver R7.
- High volumes of multicast traffic generated by R7 or R8 for these group addresses should cause new MDT data tunnels to form between R2 and XR1.

## Configuration [Click to collapse](#)

---

```
R1:
ip multicast-routing distributed
!
interface GigabitEthernet1.12
 ip pim sparse-mode
!
interface GigabitEthernet1.17
 ip pim sparse-mode

R2:
ip multicast-routing distributed
ip pim ssm default
!
ip multicast-routing vrf VPN_A distributed
!

vrf definition VPN_A
 address-family ipv4
  mdt default 232.0.0.1
  mdt data 232.0.1.0 0.0.0.255
!
interface Loopback0
 ip pim sparse-mode
!
interface GigabitEthernet1.12
 ip pim sparse-mode
!
interface GigabitEthernet1.23
 ip pim sparse-mode
!
interface GigabitEthernet1.24
 ip pim sparse-mode
!
router bgp 100
 address-family ipv4 mdt
  neighbor 19.19.19.19 activate
  neighbor 19.19.19.19 send-community extended
 exit-address-family

R3:
ip multicast-routing distributed
ip pim ssm default
!
interface GigabitEthernet1.23
 ip pim sparse-mode
!
interface GigabitEthernet1.34
 ip pim sparse-mode
!
interface GigabitEthernet1.36
 ip pim sparse-mode

R4:
ip multicast-routing distributed
```

```
ip pim ssm default
!
interface GigabitEthernet1.24
 ip pim sparse-mode
!
interface GigabitEthernet1.34
 ip pim sparse-mode
!
interface GigabitEthernet1.45
 ip pim sparse-mode
!
interface GigabitEthernet1.46
 ip pim sparse-mode

R5:
ip multicast-routing distributed
```

---

```
ip pim ssm default
!
interface GigabitEthernet1.45
 ip pim sparse-mode
!
interface GigabitEthernet1.56
 ip pim sparse-mode
!
interface GigabitEthernet1.519
 ip pim sparse-mode

R6:
ip multicast-routing distributed
ip pim ssm default
!
interface GigabitEthernet1.36
 ip pim sparse-mode
!
interface GigabitEthernet1.46
 ip pim sparse-mode
!
interface GigabitEthernet1.56
 ip pim sparse-mode
!
interface GigabitEthernet1.619
 ip pim sparse-mode

R7:
interface GigabitEthernet1.17
 ip igmp join-group 227.7.7.7

R8:
interface GigabitEthernet1.820
 ip igmp join-group 228.8.8.8

XR1:
router bgp 100
 address-family ipv4 mdt
```

```

!
neighbor 2.2.2.2
 address-family ipv4 mdt
!
multicast-routing
 address-family ipv4
  mdt source Loopback0
  interface all enable
!
vrf VPN_A
 address-family ipv4
  mdt data 232.0.1.0/24
  mdt default ipv4 232.0.0.1
  interface all enable
!
end

```

```

XR2:
multicast-routing
 address-family ipv4
  interface all enable
!
!
router pim
 address-family ipv4
  bsr candidate-bsr 20.20.20.20 hash-mask-len 30 priority 1
  bsr candidate-rp 20.20.20.20 group-list 224/4 priority 192 interval 60

```

## Verification

Verification of Multicast over MPLS L3VPN service consists of two separate portions: verification of the SP core, and verification of the customer networks. In the SP core we must ensure that the PE routers can form an MDT tunnel for their VRF, which is essentially a mGRE tunnel that uses a multicast destination address.

PE routers servicing the same multicast customer VPN need to learn about other in order to establish the MDT tunnel. In designs where the core SP network is using Source Specific Multicast (SSM) for the MDT Group Addresses, the BGP MDT SAFI needs to be enabled in order for the PE to discover each other. The discovery is leveraged via the information exchanged in the BGP IPv4 MDT updates, which contain the "MDT Source interface" each PE router is using (i.e, the source interface for the BGP session), along with the MDT group address used for the VPN. Note that MDT group address is also manually configured under the VRF. Once a PE router extracts this information from BGP, it uses SSM in the core to build a multicast tree to the PEs for which it received an MDT BGP update. This underlying multicast tree in the core is what facilitates the establishment of the MDT tunnel - allowing PE routers to become PIM adjacent in their VRFs. Note that regular IOS routers need the command `ip pim ssm default` to know which PIM Join/Prune message should belong to SSM groups vs. regular Sparse Mode groups.

If SSM multicast is not used and Any Source Multicast (ASM) is used instead, the PE routers don't need to enable the BGP MDT SAFI to discover each other. Instead, the PE routers can rely on a Rendezvous Point in the core of the SP network as a "proxy" for this information. Each PE router knows what the default MDT multicast group address is, as this is manually configured under the VRF. PE routers join the (\*, G) shared tree towards the RP for the MDT group address, which sets up the needed state in the core to forward multicast packets accordingly to all PEs who joined the tree for this group - forming a multicast tree between the PEs. Essentially the PEs don't need to know who the other PEs are ahead of time, as the RP is ensuring that packets sent to the MDT group address are forwarded to all PEs servicing the multicast VPN.

If the Multicast Tree for the MDT tunnel is properly formed, the PE routers should have (S,G) entries for each other's BGP loopback addresses and the MDT default address, as seen below.

---

CONTENTS ▼

R2#show ip mroute 232.0.0.1

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,

L - Local, P - Pruned, R - RP-bit set, F - Register flag,

T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,

X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,

U - URD, I - Received Source Specific Host Report,

Z - Multicast Tunnel, z - MDT-data group sender,

Y - Joined MDT-data group, y - Sending to MDT-data group,

G - Received BGP C-Mroute, g - Sent BGP C-Mroute,

N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,

Q - Received BGP S-A Route, q - Sent BGP S-A Route,

V - RD & Vector, v - Vector, p - PIM Joins on route,

x - VxLAN group

Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(2.2.2.2, 232.0.0.1), 00:02:15/00:03:11, flags: sT

Incoming interface: Loopback0, RPF nbr 0.0.0.0

Outgoing interface list:

GigabitEthernet1.24, Forward/Sparse, 00:02:15/00:03:11

(19.19.19.19, 232.0.0.1), 00:02:15/00:00:44, flags: sTIZ

Incoming interface: GigabitEthernet1.24, RPF nbr 20.2.4.4

Outgoing interface list:

MVRF VPNA, Forward/Sparse, 00:02:15/00:00:44

RP/0/0/CPU0:XR1#show pim topology 232.0.0.1

Thu Jun 4 01:29:55.121 UTC

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

(2.2.2.2,232.0.0.1)SPT SSM Up: 00:03:25

JP: Join(00:00:20) RPF: GigabitEthernet0/0/0/0.619,20.6.19.6 Flags:

Loopback0 00:03:25 fwd LI LH

(19.19.19.19,232.0.0.1)SPT SSM Up: 00:05:27

JP: Join(never) RPF: Loopback0,19.19.19.19\* Flags:

Loopback0 00:05:27 fwd LI LH

GigabitEthernet0/0/0/0.619 00:03:25 fwd Join(00:03:00)

RP/0/0/CPU0:XR1#show mrib route 232.0.0.1 detail

Thu Jun 4 01:33:32.916 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, MF - MPLS Encap, 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

(2.2.2.2,232.0.0.1) Ver: 0x5680 RPF nbr: 20.6.19.6 Flags: MD MH CD, FMA: 0x10000

MVPN TID: 0xe000001b

MVPN REMOTE TID: 0x0

MVPN PAYLOAD: IPv4

MDT IFH: 0x1f80

Up: 00:07:03

RPF-ID: 1, Encap-ID: 0

Incoming Interface List

GigabitEthernet0/0/0/0.619 Flags: A, Up: 00:07:03

Outgoing Interface List

Loopback0 Flags: F NS, Up: 00:07:03

(19.19.19.19,232.0.0.1) Ver: 0x4eab RPF nbr: 19.19.19.19 Flags: ME MH, FMA: 0x10000

MVPN TID: 0xe000001b

MVPN REMOTE TID: 0x0

MVPN PAYLOAD: IPv4

MDT IFH: 0x1f80

Up: 00:09:05

RPF-ID: 1, Encap-ID: 0

Incoming Interface List

Loopback0 Flags: F A, Up: 00:09:05

Outgoing Interface List

Loopback0 Flags: F A, Up: 00:09:05

GigabitEthernet0/0/0/0.619 Flags: F NS, Up: 00:07:03

In regular IOS, the lowercase “s” flag means SSM, and the Outgoing Interface List for the (19.19.19.19, 232.0.0.1) entry points to the VRF. This essentially means that the outer GRE header gets decapsulated when the packet is received in, and then it is forwarded to the VRF VPN\_A Multicast Routing Table for further processing/forwarding. XR1 shows similar output for the (2.2.2.2, 232.0.0.1) group, but the OIL doesn’t explicitly list the MVPN. Notice however that in the detailed output, XR1 depicts these entries as being used for MVPN, with the respective MDT encap and decap flags. The MDT tunnel can also be verified working as seen below.

---

```
R2#show bgp ipv4 mdt all
BGP table version is 3, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

```

Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 100:1 (default for vrf VPN_A)
*> 2.2.2.2/32      0.0.0.0                    0 ?
*>i 19.19.19.19/32 19.19.19.19                100  0 i
```

```
R2#show bgp ipv4 mdt all 19.19.19.19
BGP routing table entry for 100:1:19.19.19.19/32 version 3
```

Paths: (1 available, best #1, table IPv4-MDI-BGP-table)

```

Not advertised to any peer
Refresh Epoch 1
Local
  19.19.19.19 from 19.19.19.19 (19.19.19.19)
  Origin IGP, localpref 100, valid, internal, best,
  MDT group address: 232.0.0.1

  rx pathid: 0, tx pathid: 0x0
```

```
RP/0/0/CPU0:XR1#show bgp ipv4 mdt
Thu Jun  4 01:38:44.375 UTC
BGP router identifier 19.19.19.19, local AS number 100
BGP generic scan interval 60 secs
BGP table state: Active
Table ID: 0xe0000000 RD version: 3
BGP main routing table version 3
BGP scan interval 60 secs
```

```
Status codes: s suppressed, d damped, h history, * valid, > best
               i - internal, r RIB-failure, S stale, N Nexthop-discard
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete
Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 100:1
*>i 2.2.2.2/96      2.2.2.2                    0  100  0 ?
*> 19.19.19.19/96  0.0.0.0                    0 i
```

Processed 2 prefixes, 2 paths

```
RP/0/0/CPU0:XR1#show bgp ipv4 mdt vrf VPN_A 2.2.2.2
Thu Jun  4 01:39:12.783 UTC
BGP routing table entry for 2.2.2.2/96, Route Distinguisher: 100:1
Versions:
Process          bRIB/RIB  SendTblVer
Speaker          2         2
Last Modified: Jun  4 01:26:29.424 for 00:12:43
Paths: (1 available, best #1)
Not advertised to any peer
```

```

Path #1: Received by speaker 0
Not advertised to any peer

Local
  2.2.2.2 (metric 4) from 2.2.2.2 (2.2.2.2)
    Origin incomplete, metric 0, localpref 100, valid, internal, best, group-best
    Received Path ID 0, Local Path ID 1, version 2
    MDT group address: 232.0.0.1

```

The above output shows that R2 and XR1 are learning that each other's BGP Loopback addresses are possible endpoints for MDT tunnels, specifically using the group address 232.0.0.1. Without learning these addresses through the MDT extensions of BGP, they would not know where to send their PIM SPT Join messages for the (S,G) pair of the MDT tunnel as previously stated. Another more concise variation of this is shown below:

```

R2#show ip pim mdt bgp
MDT (Route Distinguisher + IPv4)          Router ID      Next Hop
-----
MDT group 232.0.0.1
  100:1:19.19.19.19                        19.19.19.19   19.19.19.19

RP/0/0/CPU0:XR1#show pim bgp-safi
Thu Jun  4 01:40:18.148 UTC
Grp 232.0.0.1      Src 2.2.2.2      RD 100:1      Nexthop 2.2.2.2 Ext 0, BGP

```

If the tunnel is properly formed bi-directionally, the PE routers should form a PIM adjacency over the tunnel, but inside the VRF table, not the global table. This can be seen below:

```

R2#show ip pim vrf VPN_A neighbor
PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      P - Proxy Capable, S - State Refresh Capable, G - GenID Capable,
      L - DR Load-balancing Capable
Neighbor      Interface          Uptime/Expires  Ver  DR
Address                               Prio/Mode
-----
19.19.19.19   Tunnel0            00:13:54/00:01:34 v2  1 / DR P G
10.1.2.1      GigabitEthernet1.12 00:19:46/00:01:38 v2  1 / S P G

RP/0/0/CPU0:XR1#show pim vrf VPN_A neighbor
Thu Jun  4 01:41:05.385 UTC

PIM neighbors in VRF VPN_A
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
-----
10.19.20.19*         GigabitEthernet0/0/0/0.1920 00:16:38 00:01:30 1      B P E
10.19.20.20          GigabitEthernet0/0/0/0.1920 00:16:20 00:01:16 1 (DR) B P
2.2.2.2              mdtVPN/A          00:14:17 00:01:18 1      P
19.19.19.19*         mdtVPN/A          00:16:37 00:01:42 1 (DR) P

```

Regular IOS expresses this as Tunnel0, whereas IOS XR expresses it as mdtVPN/A. Both logical constructs essentially mean the same thing, that the routers have formed a PIM adjacency over the multicast destined GRE tunnels. If there are any problems in the design up to this point, such as RPF

failure in the core, PIM adjacencies not being established, etc., none of the customer's traffic will get properly encapsulated inside GRE after it enters the provider network.

```
R2#show derived-config interface tunnel0
Building configuration...

Derived configuration : 133 bytes
!
interface Tunnel0
 ip unnumbered Loopback0
 no ip redirects
 ip mtu 1500
 tunnel source Loopback0
 tunnel mode gre multipoint
```

The next set of verifications are then on the customer side. From the CE's point of view, the PE routers are simply normal PIM neighbors in the global routing table, as seen below on R1 and XR2.

CONTENTS

```
R1#show ip pim neighbor
PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      P - Proxy Capable, S - State Refresh Capable, G - GenID Capable,
      L - DR Load-balancing Capable
Neighbor      Interface          Uptime/Expires   Ver  DR
Address                               Prio/Mode
10.1.2.2      GigabitEthernet1.12  00:22:52/00:01:32 v2  1 / DR S P G

RP/0/0/CPU0:XR2#show pim neighbor
Thu Jun  4 01:44:13.902 UTC

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
10.8.20.20*          GigabitEthernet0/0/0/0.820 00:19:31 00:01:40 1 (DR) B P E
10.19.20.19          GigabitEthernet0/0/0/0.1920 00:19:27 00:01:23 1      B P
10.19.20.20*          GigabitEthernet0/0/0/0.1920 00:19:31 00:01:38 1 (DR) B P E
20.20.20.20*          Loopback0          00:19:31 00:01:20 1 (DR) B P E
```

XR2 has configured itself as a BSR and an RP Candidate. The XR command `show pim range-list` is the equivalent of the IOS command `show ip pim rp mapping`.

```
RP/0/0/CPU0:XR2#show pim bsr election
```

```
Thu Jun 4 01:44:51.680 UTC
```

```
PIM BSR Election State
```

Cand/Elect-State	Uptime	BS-Timer	BSR	C-BSR
Elected/Accept-Pref	00:19:54	00:00:20	20.20.20.20 [1, 30]	20.20.20.20 [1, 30]

```
RP/0/0/CPU0:XR2#show pim bsr candidate-rp
```

```
Thu Jun 4 01:45:48.416 UTC
```

```
PIM BSR Candidate RP Info
```

Cand-RP	mode	scope	priority	uptime	group-list
20.20.20.20	BD	16	192	00:20:51	224/4

```
RP/0/0/CPU0:XR2#show pim range-list
```

```
Thu Jun 4 01:46:24.813 UTC
```

```
config SSM Exp: never Learnt from : 0.0.0.0
```

```
232.0.0.0/8 Up: 00:21:42 Exp: never Learnt from : 0.0.0.0
```

```
bsr SM RP: 20.20.20.20 Exp: never Learnt from : 0.0.0.0
```

```
224.0.0.0/4 Up : 00:20:27 Exp: 00:02:17 Learnt from : 20.20.20.20
```

If the MDT tunnel is working end to end, R1 should learn that XR2 is the RP via BSR.

CONTENTS

```
R1#show ip pim rp mapping
```

```
PIM Group-to-RP Mappings
```

```
Group(s) 224.0.0.0/4
```

```
RP 20.20.20.20 (?), v2
```

```
Info source: 20.20.20.20 (?), via bootstrap, priority 192, holdtime 150
```

```
Uptime: 00:20:10, expires: 00:02:18
```

Now the entire control plane for multicast is verified as working. The next step is to verify the data plane. When a multicast sender comes onto the network, the PIM DR on the attached segment should notify the RP through a PIM Register message. This can be verified by pinging a multicast group address on R7 (behind R1), and seeing if the (S,G) entry is learned on the RP, XR2.

```
RP/0/0/CPU0:XR2#show pim topology 224.1.1.1
```

```
Thu Jun 4 01:48:14.626 UTC
```

```
No PIM topology table entries found.
```

```
R7#ping 224.1.1.1
```

```
Type escape sequence to abort.
```

```
Sending 1, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:
```

```
.
```

```
RP/0/0/CPU0:XR2#show pim topology 224.1.1.1
```

```
Thu Jun 4 01:48:38.584 UTC
```

```
IP PIM Multicast Topology Table
```

```
Entry state: (*S,G)[RPT/SPT] Protocol Uptime Info
```

```
Entry flags: KAI - 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
```

```
(10.1.7.7,224.1.1.1) SM Up: 00:00:08
```

```
JP: Null(never) RPF: GigabitEthernet0/0/0.1920,10.19.20.19 Flags: KAT(00:03:23) RA RR (00:04:26)
```

```
No interfaces in immediate olist
```

The RP has installed the (S,G) entry, which means that the Register message was successfully processed. Further verification of this could be performed with the debug ip pim in regular IOS or the debug pim protocol register in IOS XR. The next verification would be to see if a (\*,G) PIM Join can be properly processed over the MDT tunnel. This is accomplished as follows:

```

RP/0/0/CPU0:XR2#show pim topology 227.7.7.7

Thu Jun  4 01:50:12.118 UTC

No PIM topology table entries found.

R7#config t
Enter configuration commands, one per line. End with CNTL/Z.
R7(config)#int Gig1.17
R7(config-if)#ip igmp join 227.7.7.7
R7(config-if)#end

RP/0/0/CPU0:XR2#show pim topology 227.7.7.7

Thu Jun  4 01:50:47.545 UTC

IP PIM Multicast Topology Table
Entry state: (*S,G)[RPT/SPT] Protocol Uptime Info

Entry flags: KAI - 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

(*,227.7.7.7) SM Up: 00:23:35 RP: 20.20.20.20*
JP: Join(never) RPF: Decapstunnel0,20.20.20.20 Flags:
GigabitEthernet0/0/0.1920 00:23:35 fwd Join(00:02:58)

```

The RP knows about the (\*,G) tree, for which it is the root. Note that the incoming interface for this entry is "Decapstunnel0" - normally IOS would have "0.0.0.0" or "Null" for a (\*,G) for which it is the RP. This means that the PIM Join message was properly sent all the way across the MDT tunnel in the SP core. This control plane message was encapsulated with a GRE header by R1 and forwarded onto the SPT tree built in the core for 231.0.0.1.

The next verification would be to ensure that traffic can actually flow. This is done bidirectionally with a sender on R7 and a receiver on R8, and vice versa. Note that due to an implementation limitation on XRv, data-plane multicast forwarding is not supported, and thus pings to the multicast groups between R7 and R8 will not work. However, we can still validate the control-plane signaling that takes place during actual traffic forwarding.

```
R7#ping 228.8.8.8 repeat 10
```

```
Type escape sequence to abort.
```

```
Sending 10, 100-byte ICMP Echos to 228.8.8.8, timeout is 2 seconds:
```

```
.....
```

```
R8#ping 227.7.7.7 repeat 10
```

```
Type escape sequence to abort.
```

```
Sending 10, 100-byte ICMP Echos to 227.7.7.7, timeout is 2 seconds:
```

```
.....
```

The PE routers should now have both of these (S,G) states in the VRF specific Multicast Routing Tables. Notice that due to the limitation of XRv, R2 does not see the the MRIB entries from the multicast traffic being originated by R3. However, XR1 does have state for the multicast traffic generated by R7.

```
R2#show ip mroute vrf VPN_A
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
      L - Local, P - Pruned, R - RP-bit set, F - Register flag,
      T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
      X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
      U - URD, I - Received Source Specific Host Report,
      Z - Multicast Tunnel, z - MDT-data group sender,
      Y - Joined MDT-data group, y - Sending to MDT-data group,
      G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
      N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
      Q - Received BGP S-A Route, q - Sent BGP S-A Route,
      V - RD & Vector, v - Vector, p - PIM Joins on route,
      x - VxLAN group
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
```

```
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 228.8.8.8), 00:12:14/stopped, RP 20.20.20.20, flags: SP
  Incoming interface: Tunnel0, RPF nbr 19.19.19.19
  Outgoing interface list: Null

(10.1.7.7, 228.8.8.8), 00:12:14/00:01:12, flags: Ty
  Incoming interface: GigabitEthernet1.12, RPF nbr 10.1.2.1
  Outgoing interface list:
    Tunnel0, Forward/Sparse, 00:04:28/00:02:59

(*, 227.7.7.7), 00:46:14/00:03:26, RP 20.20.20.20, flags: S
  Incoming interface: Tunnel0, RPF nbr 19.19.19.19
  Outgoing interface list:
    GigabitEthernet1.12, Forward/Sparse, 00:46:14/00:03:26

(*, 224.0.1.40), 00:52:30/00:02:44, RP 0.0.0.0, flags: DCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1.12, Forward/Sparse, 00:52:28/00:02:44
```

```
RP/0/0/CPU0:XR1#show pim vrf VPN_A topology
Thu Jun  4 02:14:23.918 UTC

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
```

```
(* ,224.0.1.40) DM Up: 00:49:56 RP: 0.0.0.0
```

```
JP: Null(never) RPF: Null,0.0.0.0 Flags:
```

```
GigabitEthernet0/0/0.1920 00:49:56 off LI II
```

```
(* ,227.7.7.7) SM Up: 00:47:11 RP: 20.20.20.20
```

```
JP: Join(00:00:41) RPF: GigabitEthernet0/0/0.1920,10.19.20.20 Flags:
```

```
mdtVPN/A 00:47:11 fwd Join(00:02:36)
```

```
(10.1.7.7,228.8.8.8)SPT SM Up: 00:05:25
```

```
JP: Join(00:00:21) RPF: mdtVPN/A,2.2.2.2 Flags:
```

```
GigabitEthernet0/0/0.1920 00:05:25 fwd Join(00:03:01)
```

The issue due to XRv not being able to make the connection between incoming and outgoing interfaces. XR2 sees the (\*, G) from the Joins generated by R7, and it also has an (S, G) state for the traffic originated by R8 towards this group. Normally the multicast router would take the outgoing interface from the (\*,G) entry and use it in the (S, G), however due to the limitation, XR2 displays the outgoing interface of the (S, G) as "No interfaces in immediate olist".

```
RP/0/0/CPU0:XR2#show pim topology 227.7.7.7
```

```
Thu Jun 4 02:16:41.449 UTC
```

#### 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
```

```
(* ,227.7.7.7) SM Up: 00:49:29 RP: 20.20.20.20*
```

```
JP: Join(never) RPF: Decapstunnel0,20.20.20.20 Flags:
```

```
GigabitEthernet0/0/0.1920 00:49:29 fwd Join(00:03:04)
```

```
(10.8.20.8,227.7.7.7)SPT SM Up: 00:07:07
```

```
JP: Join(00:00:44) RPF: GigabitEthernet0/0/0.820,10.8.20.8* Flags: KAT(00:00:01) RA RR (00:03:24)
```

```
No interfaces in immediate olist
```

The final verification is to see if new MDT tunnels are formed after higher volumes of multicast feeds are generated. This can be tested by varying the source and destination addresses of the feeds, as shown below.

```
R7#config t
Enter configuration commands, one per line. End with CNTL/Z.
R7(config)#int g1.17
R7(config-if)#ip address 10.1.7.100 255.255.255.0
R7(config-if)#end
R7#
```

```
R8#config t
Enter configuration commands, one per line. End with CNTL/Z.
R8(config)#int g1.820
R8(config-if)#ip igmp join 228.88.88.88

R7#ping 228.88.88.88 repeat 100000 timeout 0 size 1400
```

Type escape sequence to abort.

Sending 100000, 1400-byte ICMP Echos to 228.88.88.88, timeout is 0 seconds:

```
.....
.....
.....
```

<snip>

```
R2#show ip pim vrf VPN_A mdt send
```

MDT-data send list for VRF: VPN\_A

(source, group)	MDT-data group/num	ref_count
(10.1.7.100, 228.88.88.88)	232.0.1.0	1

This new feed has caused a new MDT to form with the MDT Data address of 232.0.1.0. R2 should be the root of this tree, with XR1 as a receiver.

```
R2#show ip mroute 232.0.1.0
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
      L - Local, P - Pruned, R - RP-bit set, F - Register flag,
      T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
      X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
      U - URD, I - Received Source Specific Host Report,
      Z - Multicast Tunnel, z - MDT-data group sender,
      Y - Joined MDT-data group, y - Sending to MDT-data group,
      G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
      N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
      Q - Received BGP S-A Route, q - Sent BGP S-A Route,
      V - RD & Vector, v - Vector, p - PIM Joins on route,
      x - VxLAN group
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
```

```
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(2.2.2.2, 232.0.1.0), 00:03:39/00:03:11, flags: sT
  Incoming interface: Loopback0, RPF nbr 0.0.0.0
  Outgoing interface list:
    GigabitEthernet1.24, Forward/Sparse, 00:00:32/00:02:57
```

```
R4#show ip mroute 232.0.1.0
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
      L - Local, P - Pruned, R - RP-bit set, F - Register flag,
      T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
      X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
      U - URD, I - Received Source Specific Host Report,
      Z - Multicast Tunnel, z - MDT-data group sender,
      Y - Joined MDT-data group, y - Sending to MDT-data group,
      G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
      N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
      Q - Received BGP S-A Route, q - Sent BGP S-A Route,
      V - RD & Vector, v - Vector, p - PIM Joins on route,
      x - VxLAN group
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
```

```
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(2.2.2.2, 232.0.1.0), 00:01:03/00:03:24, flags: sT
  Incoming interface: GigabitEthernet1.24, RPF nbr 20.2.4.2
  Outgoing interface list:
    GigabitEthernet1.46, Forward/Sparse, 00:01:03/00:03:24
```

```
RP/0/0/CPU0:XR1#show pim topology 232.0.1.0
Thu Jun  4 02:25:12.684 UTC

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

```

```

( 2.2.2.2,232.0.1.0 )SPT SSM Up: 00:02:00

JP: Join(now) RPF: GigabitEthernet0/0/0.619,20.6.19.6 Flags:

Loopback0          00:02:00  fwd LI LH

```

Note that Multicast VPN changes how the RPF interface and neighbor are selected on the PEs. In regular unicast routing, the RPF neighbor and interface are by default derived from the unicast routing table. These rules change Multicast VPN however. The RPF interface is set to the VRF's MDT Tunnel on a PE if a route to the source of the multicast packets, in this case R7's 10.1.7.100, is installed in the VRF table via VPNv4 BGP. The RPF neighbor is set to the VPNv4 next-hop of the particular route, in this case 2.2.2.2.

CONTENTS

```

RP/0/0/CPU0:XR1#show route vrf VPN_A 10.1.7.100

Thu Jun  4 05:32:02.876 UTC

Routing entry for 10.1.7.0/24

  Known via "bgp 100", distance 200, metric 2, type internal

  Installed Jun  4 01:24:41.163 for 04:07:21

Routing Descriptor Blocks

  2.2.2.2, from 2.2.2.2

    Nexthop in Vrf: "default", Table: "default", IPv4 Unicast, Table Id: 0xe0000000

    Route metric is 2

  No advertising protos.

RP/0/0/CPU0:XR1#show pim vrf VPN_A rpf 10.1.7.100

Thu Jun  4 05:32:25.024 UTC

Table: IPv4-Unicast-default

* 10.1.7.100/32 [200/2]

  via mdtVPN/A with rpf neighbor 2.2.2.2

  Connector: 100:1:2.2.2.2, Nexthop: 2.2.2.2

```

Multicast in an MPLS Layer 3 VPN used to be referred to as "Draft Rosen" (<http://tools.ietf.org/html/draft-rosen-vpn-mcast-15>), as this was the name of the draft originally describing this technology. Note that this draft has been formally evolved into RFC-6037 (<http://tools.ietf.org/html/rfc6037>), and there have been several changes and enhancements done to the standard.

CONTENTS ▼