Strict RPF Check for mVPN

Document ID: 118677

Contributed by Luc De Ghein, Cisco TAC Engineer.
Dec 17, 2014

Contents

Introduction

This document describes the Strict Reverse Path Forwarding (RPF) feature for Multicast over VPN (mVPN). This document uses an example and the implementation in Cisco IOS® in order to illustrate the behavior.

Background Information

RPF implies that the incoming interface is checked towards the source. Although the interface is checked to determine that it is the correct one towards the source, it is not checked to determine that it is the correct RPF neighbor on that interface. On a multi−access interface, there could be more than one neighbor to which you could RPF. The result could be that the router receives twice the same multicast stream on that interface and forwards both.

In networks where Protocol Independent Multicast (PIM) runs on the multi−access interface, this is not an issue, because the duplicate multicast stream causes the assert mechanism to run and one multicast stream will no longer be received. In some cases, PIM does not run on the Multicast Distribution Tree (MDT), which is a multi−access interface. In those cases, Border Gateway Protocol (BGP) is the overlay signaling protocol.

In the profiles with Partitioned MDT, even if PIM runs as the overlay protocol, it can be impossible to have asserts. The reason for this is that one Ingress Provider Edge (PE) does not join the Partitioned MDT from another Ingress PE in the scenarios where there are two or more Ingress PE routers. Each Ingress PE router can forward the multicast stream onto its Partitioned MDT without the other Ingress PE router seeing the multicast traffic. The fact that two different Egress PE routers each join an MDT towards a different Ingress PE router for the same multicast stream is a valid scenario: it is called Anycast Source. This allows different receivers to join the same multicast stream but over a different path in the Multiprotocol Label Switching (MPLS) core. See Figure 1 for an example of Anycast Source.

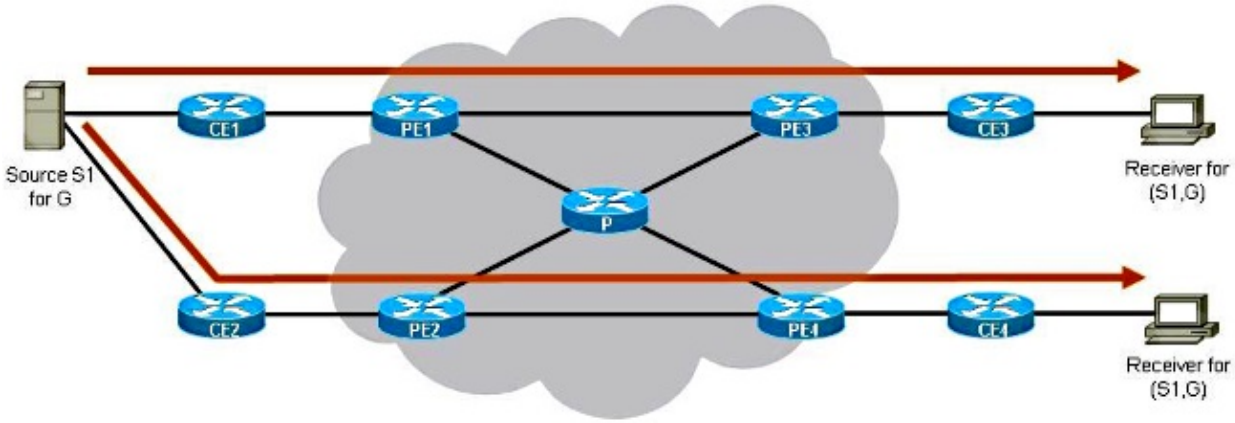

Figure 1

There are two Ingress PE routers: PE1 and PE2. There are two Egress PE routers: PE3 and PE4. Each Egress PE router has a different Ingress PE router as its RPF neighbor. PE3 has PE1 as its RPF neighbor. PE4 has PE2 as its RPF neighbor. The Egress PE routers pick their closest Ingress PE router as their RPF neighbor.

The Stream (S1,G) will go from S1 to Receiver 1 over the top path and from S1 to Receiver 2 over the bottom path. There is no intersection of the two streams over the two paths (each path in the MPLS core is a different Partitioned MDT).

If the MDT was a Default MDT − such as in in the Default MDT profiles − then this would not work because the two multicast streams would be on the same Default MDT and the assert mechanism would run. If the MDT is a Data MDT in the Default MDT profiles, then all Ingress PE routers join the Data MDT from the other Ingress PE routers and as such see the multicast traffic from each other and the assert mechanism runs again. If the overlay protocol is BGP, then there is Upstream Multicast Hop (UMH) selection and only one Ingress PE router is selected as the forwarder, but this is per MDT.

Anycast Source is one of the big advantages of running Partitioned MDT.

Problem

The regular RPF check confirms that the packets arrive at the router from the correct RPF interface. There is no check to confirm that the packets are received from the correct RPF neighbor on that interface.

See Figure 2. It shows an issue where duplicate traffic is persistently forwarded in a scenario with Partitioned MDT. It shows that the regular RPF check in the case of Partitioned MDT is not sufficient in order to avoid duplicate traffic.

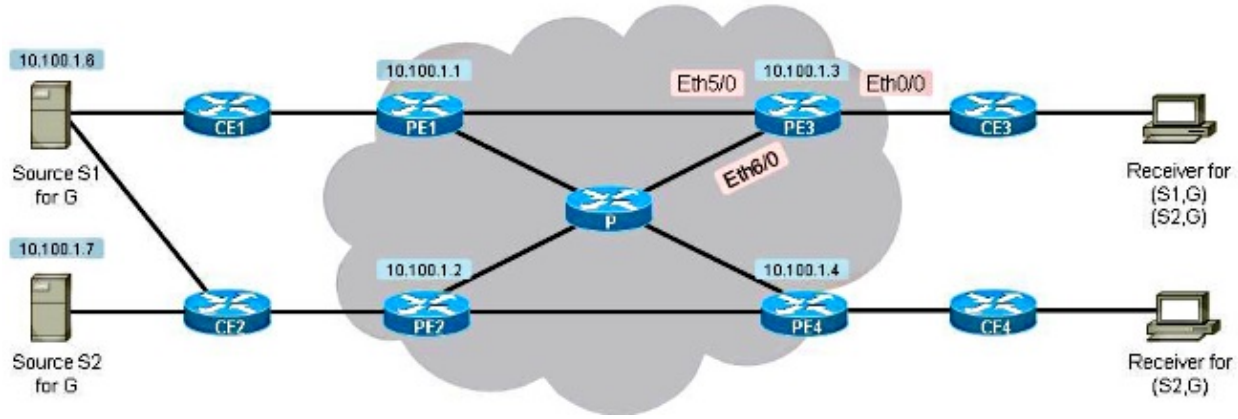

Figure 2

There are two receivers. The first receiver is set up to receive traffic for (S1,G) and (S2,G). The second receiver is set up to receive traffic for (S2,G) only. There is Partitioned MDT, and BGP is the overlay signaling protocol. Note that Source S1 is reachable via both PE1 and PE2. The core−tree protocol is Multipoint Label Distribution Protocol (mLDP).

Each PE router advertises a Type 1 BGP IPv4 mVPN route, which indicates that it is a candidate to be the root of a Partitioned MDT.

```
PE3#show bgp ipv4 mvpn vrf one
BGP table version is 257, local router ID is 10.100.1.3
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−pah, 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: 1:3 (default for vrf one)
*>i [1][1:3][10.100.1.1]/12
                      10.100.1.1               0   100     0 ?
*>i [1][1:3][10.100.1.2]/12
                       10.100.1.2               0   100     0 ?
*> [1][1:3][10.100.1.3]/12
                       0.0.0.0                            32768 ?
*>i [1][1:3][10.100.1.4]/12
                       10.100.1.4               0   100     0 ?
```

PE3 finds PE1 as the RPF neighbor for S1 after a lookup for the unicast route for S1.

```
PE3#show bgp vpnv4 unicast vrf one 10.100.1.6/32
BGP routing table entry for 1:3:10.100.1.6/32, version 16
Paths: (2 available, best #2, table one)
Advertised to update−groups:
     5
Refresh Epoch 2
65001, imported path from 1:2:10.100.1.6/32 (global)
   10.100.1.2 (metric 21) (via default) from 10.100.1.5 (10.100.1.5)
     Origin incomplete, metric 0, localpref 100, valid, internal
     Extended Community: RT:1:1 MVPN AS:1:0.0.0.0 MVPN VRF:10.100.1.2:1
     Originator: 10.100.1.2, Cluster list: 10.100.1.5
     mpls labels in/out nolabel/20
     rx pathid: 0, tx pathid: 0
Refresh Epoch 2
65001, imported path from 1:1:10.100.1.6/32 (global)
   10.100.1.1 (metric 11) (via default) from 10.100.1.5 (10.100.1.5)
```

```
     Origin incomplete, metric 0, localpref 100, valid, internal, best
     Extended Community: RT:1:1 MVPN AS:1:0.0.0.0 MVPN VRF:10.100.1.1:1
     Originator: 10.100.1.1, Cluster list: 10.100.1.5
     mpls labels in/out nolabel/29
     rx pathid: 0, tx pathid: 0x0

PE3#show ip rpf vrf one 10.100.1.6
RPF information for ? (10.100.1.6)
  RPF interface: Lspvif0
  RPF neighbor: ? (10.100.1.1)
  RPF route/mask: 10.100.1.6/32
  RPF type: unicast (bgp 1)
  Doing distance-preferred lookups across tables
  RPF topology: ipv4 multicast base, originated from ipv4 unicast base
```

PE3 selects PE1 as the RPF neighbor for (S1,G) and joins the Partitioned MDT with PE1 as root. PE3 selects PE2 as the RPF neighbor for (S2,G) and joins the Partitioned MDT with PE2 as root.

```
PE3#show bgp vpnv4 unicast vrf one 10.100.1.7/32
BGP routing table entry for 1:3:10.100.1.7/32, version 18
Paths: (1 available, best #1, table one)
Advertised to update-groups:
     6
Refresh Epoch 2
65002, imported path from 1:2:10.100.1.7/32 (global)
     10.100.1.2 (metric 21) (via default) from 10.100.1.5 (10.100.1.5)
     Origin incomplete, metric 0, localpref 100, valid, internal, best
     Extended Community: RT:1:1 MVPN AS:1:0.0.0.0 MVPN VRF:10.100.1.2:1
     Originator: 10.100.1.2, Cluster list: 10.100.1.5
     mpls labels in/out nolabel/29
     rx pathid: 0, tx pathid: 0x0

PE3#show ip rpf vrf one 10.100.1.7
RPF information for ? (10.100.1.7)
  RPF interface: Lspvif0
  RPF neighbor: ? (10.100.1.2)
  RPF route/mask: 10.100.1.7/32
  RPF type: unicast (bgp 1)
  Doing distance-preferred lookups across tables
  RPF topology: ipv4 multicast base, originated from ipv4 unicast base
```

PE4 selects PE2 as the RPF neighbor for (S1,G) and joins the Partitioned MDT with PE1 as root.

```
PE4#show bgp vpnv4 unicast vrf one 10.100.1.6/32
BGP routing table entry for 1:4:10.100.1.6/32, version 138
Paths: (2 available, best #1, table one)
Advertised to update-groups:
     2
Refresh Epoch 2
65001, imported path from 1:2:10.100.1.6/32 (global)
     10.100.1.2 (metric 11) (via default) from 10.100.1.5 (10.100.1.5)
     Origin incomplete, metric 0, localpref 100, valid, internal, best
     Extended Community: RT:1:1 MVPN AS:1:0.0.0.0 MVPN VRF:10.100.1.2:1
     Originator: 10.100.1.2, Cluster list: 10.100.1.5
     mpls labels in/out nolabel/20
     rx pathid: 0, tx pathid: 0x0
Refresh Epoch 2
65001, imported path from 1:1:10.100.1.6/32 (global)
   10.100.1.1 (metric 21) (via default) from 10.100.1.5 (10.100.1.5)
     Origin incomplete, metric 0, localpref 100, valid, internal
     Extended Community: RT:1:1 MVPN AS:1:0.0.0.0 MVPN VRF:10.100.1.1:1
     Originator: 10.100.1.1, Cluster list: 10.100.1.5
     mpls labels in/out nolabel/29
     rx pathid: 0, tx pathid: 0
```

```
PE4#show ip rpf vrf one 10.100.1.6
RPF information for ? (10.100.1.6)
  RPF interface: Lspvif0
  RPF neighbor: ? (10.100.1.2)
  RPF route/mask: 10.100.1.6/32
  RPF type: unicast (bgp 1)
  Doing distance-preferred lookups across tables
  RPF topology: ipv4 multicast base, originated from ipv4 unicast base
```

Notice that the RPF interface is Lspvif0 for both S1 (10.100.1.6) and S2 (10.100.1.7).

PE3 joins the Partitioned MDT from PE2 for (S2,G), and PE4 joins the Partitioned MDT from PE2 for (S1,G). PE1 joins the Partitioned MDT from PE1 for (S1,G). You can see this by the type 7 BGP IPv4 mVPN routes received on PE1 and PE2.

```
PE1#show bgp ipv4 mvpn vrf one
BGP table version is 302, local router ID is 10.100.1.1
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: 1:1 (default for vrf one)
*>i [7][1:1][1][10.100.1.6/32][232.1.1.1/32]/22
                       10.100.1.3               0   100     0 ?
```

```
PE2#show bgp ipv4 mvpn vrf one
BGP table version is 329, local router ID is 10.100.1.2
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: 1:2 (default for vrf one)
*>i [7][1:2][1][10.100.1.6/32][232.1.1.1/32]/22
                       10.100.1.4               0   100     0 ?
*>i [7][1:2][1][10.100.1.7/32][232.1.1.1/32]/22
                       10.100.1.3               0   100     0 ?
```

The multicast entries on PE3 and PE4:

```
PE3#show ip mroute vrf one 232.1.1.1
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

(10.100.1.7, 232.1.1.1), 21:18:24/00:02:46, flags: sTg
 Incoming interface: Lspvif0, RPF nbr 10.100.1.2
```

```
 Outgoing interface list:
   Ethernet0/0, Forward/Sparse, 00:11:48/00:02:46

(10.100.1.6, 232.1.1.1), 21:18:27/00:03:17, flags: sTg
 Incoming interface: Lspvif0, RPF nbr 10.100.1.1
 Outgoing interface list:
   Ethernet0/0, Forward/Sparse, 00:11:48/00:03:17

PE4#show ip mroute vrf one 232.1.1.1
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

(10.100.1.6, 232.1.1.1), 20:50:13/00:02:37, flags: sTg
 Incoming interface: Lspvif0, RPF nbr 10.100.1.2
 Outgoing interface list:
   Ethernet0/0, Forward/Sparse, 20:50:13/00:02:37
```

This shows that PE3 joins the Point−to−Multipoint (P2MP) tree rooted at PE1 and also the tree rooted at PE2:

```
PE3#show mpls mldp database
* Indicates MLDP recursive forwarding is enabled

LSM ID : A   Type: P2MP   Uptime : 00:18:40
 FEC Root           : 10.100.1.1
 Opaque decoded     : [gid 65536 (0x00010000)]
 Opaque length     : 4 bytes
 Opaque value       : 01 0004 00010000
 Upstream client(s) :
   10.100.1.1:0   [Active]
     Expires       : Never         Path Set ID : A
     Out Label (U) : None         Interface   : Ethernet5/0*
     Local Label (D): 29           Next Hop     : 10.1.5.1
 Replication client(s):
   MDT (VRF one)
     Uptime         : 00:18:40     Path Set ID : None
     Interface     : Lspvif0

LSM ID : B   Type: P2MP   Uptime : 00:18:40
 FEC Root           : 10.100.1.2
 Opaque decoded     : [gid 65536 (0x00010000)]
 Opaque length     : 4 bytes
 Opaque value       : 01 0004 00010000
 Upstream client(s) :
   10.100.1.5:0  [Active]
     Expires       : Never         Path Set ID : B
     Out Label (U) : None         Interface   : Ethernet6/0*
     Local Label (D): 30           Next Hop     : 10.1.3.5
 Replication client(s):
   MDT (VRF one)
     Uptime       : 00:18:40     Path Set ID : None
     Interface     : Lspvif0
```

This shows that PE4 joins the P2MP tree rooted at PE2:

```
PE4#show mpls mldp database
* Indicates MLDP recursive forwarding is enabled

LSM ID : 3   Type: P2MP   Uptime : 21:17:06
 FEC Root           : 10.100.1.2
 Opaque decoded     : [gid 65536 (0x00010000)]

 Opaque value       : 01 0004 00010000
 Upstream client(s) :
   10.100.1.2:0   [Active]
     Expires       : Never         Path Set ID : 3
     Out Label (U) : None         Interface   : Ethernet5/0*
     Local Label (D): 29           Next Hop     : 10.1.6.2
 Replication client(s):
   MDT (VRF one)
     Uptime         : 21:17:06     Path Set ID : None
     Interface     : Lspvif0
```

S1 and S2 stream for the Group 232.1.1.1 with 10 pps. You can see the streams at PE3 and PE4. However, at PE3, you can see the rate for (S1,G) as 20 pps.

```
PE3#show ip mroute vrf one 232.1.1.1 count
Use "show ip mfib count" to get better response time for a large number of mroutes.

IP Multicast Statistics
3 routes using 1692 bytes of memory
2 groups, 1.00 average sources per group
Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kilobits per second
Other counts: Total/RPF failed/Other drops(OIF-null, rate-limit etc)

Group: 232.1.1.1, Source count: 2, Packets forwarded: 1399687, Packets received:
2071455
  Source: 10.100.1.7/32, Forwarding: 691517/10/28/2, Other: 691517/0/0
  Source: 10.100.1.6/32, Forwarding: 708170/20/28/4, Other: 1379938/671768/0

PE4#show ip mroute vrf one 232.1.1.1 count
Use "show ip mfib count" to get better response time for a large number of mroutes.

IP Multicast Statistics
2 routes using 1246 bytes of memory
2 groups, 0.50 average sources per group
Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kilobits per second
Other counts: Total/RPF failed/Other drops(OIF-null, rate-limit etc)

Group: 232.1.1.1, Source count: 1, Packets forwarded: 688820, Packets received:
688820
  Source: 10.100.1.6/32, Forwarding: 688820/10/28/2, Other: 688820/0/0

PE3#show interfaces ethernet0/0 | include rate
 Queueing strategy: fifo
  30 second input rate 0 bits/sec, 0 packets/sec
  30 second output rate 9000 bits/sec, 30 packets/sec
```

There is a duplicate stream. This duplication is the result of the presence of stream (S1,G) on the Partitioned MDT from PE1 and on the Partitioned MDT from PE2. This second Partitioned MDT, from PE2, was joined by PE3 in order to get the stream (S2,G). But, because PE4 joined the Partitioned MDT from PE2 in order to get (S1,G), (S1,G) is also present on the Partitioned MDT from PE2. Hence, PE3 receives the stream (S1,G) from both Partitioned MDTs it joined.

PE3 cannot discriminate between the packets for (S1,G) it receives from PE1 and PE2. Both streams are

received on the correct RPF interface: Lspvif0.

```
PE3#show ip multicast vrf one mpls vif

Interface   Next-hop            Application     Ref-Count   Table / VRF name   Flags
 Lspvif0     0.0.0.0             MDT               N/A       1   (vrf one) 0x1
```

The packets could arrive on different incoming physical interfaces on PE3 or on the same interface. In any case, the packets from the different streams for (S1,G) do arrive with a different MPLS label at PE3:

```
PE3#show mpls forwarding-table vrf one
Local     Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label     Label     or Tunnel Id     Switched     interface
29   [T] No Label   [gid 65536 (0x00010000)][V]   \
                                      768684       aggregate/one
30   [T] No Label   [gid 65536 (0x00010000)][V]   \
                                      1535940       aggregate/one

[T]     Forwarding through a LSP tunnel.
       View additional labelling info with the 'detail' option
```

Solution

The solution is to have a stricter RPF. With strict RPF, the router checks from which neighbor the packets are received on the RPF interface. Without strict RPF, the only check is to determine if the incoming interface is the RPF interface, but not if the packets are received from the correct RPF neighbor on that interface.

Notes for Cisco IOS

Here are some important notes about RPF with Cisco IOS.

- When you change to/from strict RPF mode, either configure it before you configure the Partitioned MDT or clear BGP. If you only configure the strict RPF command, it will not create another Lspvif interface immediately.
- Strict RPF is not enabled by default in Cisco IOS.
- It is not supported to have the ***strict-rpf*** command with Default MDT profiles.

Configuration

You can configure strict RPF on PE3 for the Virtual Routing and Forwarding (VRF).

```
vrf definition one
 rd 1:3
 !
 address-family ipv4
  mdt auto-discovery mldp
  mdt strict-rpf interface
  mdt partitioned mldp p2mp
  mdt overlay use-bgp
  route-target export 1:1
  route-target import 1:1
 exit-address-family
!
```

The RPF information has changed:

```
PE3#show ip rpf vrf one 10.100.1.6
RPF information for ? (10.100.1.6)
  RPF interface: Lspvif0
  Strict-RPF interface: Lspvif1
  RPF neighbor: ? (10.100.1.1)
  RPF route/mask: 10.100.1.6/32
  RPF type: unicast (bgp 1)
  Doing distance-preferred lookups across tables
  RPF topology: ipv4 multicast base, originated from ipv4 unicast base

PE3#show ip rpf vrf one 10.100.1.7
RPF information for ? (10.100.1.7)
  RPF interface: Lspvif0
  Strict-RPF interface: Lspvif2
  RPF neighbor: ? (10.100.1.2)
  RPF route/mask: 10.100.1.7/32
  RPF type: unicast (bgp 1)
  Doing distance-preferred lookups across tables
  RPF topology: ipv4 multicast base, originated from ipv4 unicast base
```

PE3 created a Lspvif interface per Ingress PE. The Lspvif interface is created per Ingress PE, per Address Family (AF), and per VRF. The RPF for 10.100.1.6 now points to interface Lspvif1 and the RPF for 10.100.1.7 now points to interface Lspvif2.

```
PE3#show ip multicast vrf one mpls vif

Interface   Next-hop          Application     Ref-Count   Table / VRF name   Flags
Lspvif0     0.0.0.0             MDT               N/A       1   (vrf one) 0x1
Lspvif1     10.100.1.1          MDT               N/A       1   (vrf one) 0x1
Lspvif2     10.100.1.2          MDT               N/A       1   (vrf one) 0x1
```

Now, the RPF check for packets (S1,G) from PE1 are checked against RPF interface Lspvif1. These packets come in with MPLS label 29. The RPF check for packets (S2,G) from PE2 are checked against RPF interface Lspvif2. These packets come in with MPLS label 30. The streams arrive on PE3 through different incoming interfaces, but this could also be the same interface. However, due to the fact that mLDP never uses Penultimate-Hop-Popping (PHP), there is always a regular MPLS label on top of the multicast packets. The (S1,G) packets that arrive from PE1 and from PE2 are on two different Partitioned MDTs and therefore have a different MPLS label. Hence, PE3 can discriminate between the (S1,G) stream that comes from PE1 and the (S1,G) stream that comes from PE2. In this way, the packets can be kept apart by PE3 and an RPF can be performed against different Ingress PE routers.

The mLDP database on PE3 now shows the different Lspvif interfaces per Ingress PE.

```
PE3#show mpls mldp database
* Indicates MLDP recursive forwarding is enabled

LSM ID : C   Type: P2MP   Uptime : 00:05:58
FEC Root           : 10.100.1.1
Opaque decoded     : [gid 65536 (0x00010000)]
Opaque length     : 4 bytes
Opaque value       : 01 0004 00010000
Upstream client(s) :
   10.100.1.1:0   [Active]
     Expires       : Never         Path Set ID : C
     Out Label (U) : None         Interface   : Ethernet5/0*
     Local Label (D): 29           Next Hop     : 10.1.5.1
Replication client(s):
   MDT (VRF one)
     Uptime         : 00:05:58     Path Set ID : None
     Interface     : Lspvif1

LSM ID : D   Type: P2MP   Uptime : 00:05:58
```

```
FEC Root           : 10.100.1.2
Opaque decoded     : [gid 65536 (0x00010000)]
Opaque length     : 4 bytes
Opaque value       : 01 0004 00010000
Upstream client(s) :
   10.100.1.5:0   [Active]
     Expires       : Never         Path Set ID : D
     Out Label (U) : None         Interface   : Ethernet6/0*
     Local Label (D): 30           Next Hop     : 10.1.3.5
Replication client(s):
   MDT (VRF one)
     Uptime         : 00:05:58     Path Set ID : None
     Interface     : Lspvif2
```

Strict RPF or RPF per Ingress PE works due to the fact that the multicast streams come in to the Ingress PE with a different MPLS label per ingress PE:

```
PE3#show mpls forwarding-table vrf one
Local     Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label     Label     or Tunnel Id     Switched     interface
29   [T] No Label   [gid 65536 (0x00010000)][V]   \
                                      162708       aggregate/one
30   [T] No Label   [gid 65536 (0x00010000)][V]   \
                                      162750       aggregate/one

[T]     Forwarding through a LSP tunnel.
       View additional labelling info with the 'detail' option
```

The proof that strict RPF works is that there is no longer a duplicate stream (S1,G) forwarded on PE3. The duplicate stream still arrives on PE3, but it is dropped due to the RPF failure. The RPF failure counter is at 676255 and increases constantly at a rate of 10 pps.

```
PE3#show ip mroute vrf one 232.1.1.1 count
Use "show ip mfib count" to get better response time for a large number of mroutes.

IP Multicast Statistics
3 routes using 1692 bytes of memory
2 groups, 1.00 average sources per group
Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kilobits per second
Other counts: Total/RPF failed/Other drops(OIF-null, rate-limit etc)

Group: 232.1.1.1, Source count: 2, Packets forwarded: 1443260, Packets received:
2119515
Source: 10.100.1.7/32, Forwarding: 707523/10/28/2, Other: 707523/0/0
Source: 10.100.1.6/32, Forwarding: 735737/10/28/2, Other: 1411992/676255/0
```

The output rate at PE3 is now 20 pps, which is 10 pps for each stream (S1,G) and (S2,G):

```
PE3#show interfaces ethernet0/0 | include rate
Queueing strategy: fifo
30 second input rate 0 bits/sec, 0 packets/sec
30 second output rate 6000 bits/sec, 20 packets/sec
```

Conclusion

Strict RPF Check must be used for the mVPN deployment models that use Partitioned MDT.

Things might appear to work, even if you do not configure the strict RPF check for the mVPN deployment models with Partitioned MDT: the multicast streams are delivered to the receivers. However, there is the possibility that there is duplicate multicast traffic when sources are connected to multiple Ingress PE routers. This leads to a waste of bandwidth in the network and can adversely affect the multicast application on the

receivers. Hence, it is a must to configure strict RPF Check for the mVPN deployment models that uses Partitioned MDT.

Updated: Dec 17, 2014 Document ID: 118677