

mVPN的严格RPF检查

目录

[简介](#)

[背景信息](#)

[问题](#)

[解决方案](#)

[Cisco IOS的笔记](#)

[配置](#)

[结论](#)

简介

本文描述组播的严格反向路径转发(RPF)功能在VPN (mVPN)。本文在Cisco IOS使用一示例和实施为了说明行为。

背景信息

RPF暗示流入接口被检查往来源。虽然接口被检查确定是正确一个往来源，没有被检查确定它是该接口的正确RPF邻居。在多路访问接口，可能有超过您可能RPF的一个邻居。结果可能是路由器两次接收在该接口的同一组播流并且转发两个。

在独立于协议的组播的网络中(PIM)在多路访问接口运行，这不是问题，因为重复项组播流造成主张机制运行，并且一组播流不再将接收。有时，PIM在组播分配树(MDT)不运行，是多路访问接口。在那些情况下，边界网关协议(BGP)是重叠信令协议。

在与分成的MDT的配置文件，即使PIM运行作为重叠协议，有主张可以无法的。对此的原因是一个入口服务商边缘不加入从另一个入口PE的分成的MDT在有两个或多个入口PE路由器的方案。每个入口PE路由器能转发在其分成的MDT上的组播流，不用看到组播数据流的另一个入口PE路由器。事实两个不同的出口PE路由器每加入往一个不同的入口PE路由器的一MDT同一组播流的是一个有效方案：它呼叫Anycast Source。这允许不同的接收方加入同一组播流，但是在多协议标签交换(MPLS)核心的一个不同的路径。请参阅图1关于任播来源示例。

图 1

有两个入口PE路由器：PE1及PE2。有两个出口PE路由器：PE3和PE4。每个出口PE路由器有一个不同的入口PE路由器作为其RPF邻居。PE3有PE1作为其RPF邻居。PE4有PE2作为其RPF邻居。出口PE路由器选择他们最接近的入口PE路由器作为他们的RPF邻居。

数据流(S1,G)将去从S1到在顶部路径的接收方1和从S1到在底下路径的接收方2。没有两数据流的交叉点在两个路径的(MPLS核心的每个路径是一不同的分成的MDT)。

如果MDT是默认MDT -例如默认MDT配置文件-那么这不会工作，因为两组播流在同样默认

MDT，并且主张机制将运行。如果MDT是在默认MDT配置文件的一数据MDT，则所有入口PE路由器加入从其他入口PE路由器和，因为这样看到组播数据流从彼此，并且主张机制的数据MDT再运作。如果重叠协议是BGP，则有上行组播跳(UMH)选择，并且仅一个入口PE路由器选择作为转发器，但是这是每MDT。

任播来源是其中一个运行分成的MDT大优点。

问题

正常RPF检查确认数据包到达在从正确RPF接口的路由器。没有确认的检查数据包从该接口的正确RPF邻居接收。

请参阅图 2。它显示问题重复项流量在与分成的MDT的地方一个方案不变转发。它显示一旦分成的MDT的正常RPF检查不是满足为了避免重复的流量。

图 2

有两个接收方。第一个接收方设置收到流量为(S1,G)和(S2,G)。第二个接收方设置收到(S2,G)只的流量。有分成的MDT，并且BGP是重叠信令协议。注意来源S1通过两PE1及PE2是可及的。核心树协议是多点标签转发协议(mLDP)。

每个PE路由器通告类型1 BGP IPv4 mVPN路由，表明它是的候选一分成的MDT的根。

```
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查找PE1作为S1的RPF邻居，在单播路由的查找S1的后。

```
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选择PE1作为RPF邻居为(S1,G)并且加入与PE1的分成的MDT作为根。PE3选择PE2作为RPF邻居为(S2,G)并且加入与PE2的分成的MDT作为根。

```
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选择PE2作为RPF邻居为(S1,G)并且加入与PE1的分成的MDT作为根。

```
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

注意RPF接口是S1 (10.100.1.6)和S2的(10.100.1.7) Lspvif0。

PE3加入从PE2的分成的MDT为(S2,G)，并且PE4加入从PE2的分成的MDT为(S1,G)。PE1加入从PE1的分成的MDT为(S1,G)。您能由在PE1及PE2接收的类型7 BGP IPv4 mVPN路由看到此。

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	?

在PE3和PE4的组播条目：

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

这显示PE3加入点对多点(P2MP)树根源在PE1并且树根源在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
```

这显示PE4加入P2MP树根源在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和S2为有10 pps的组232.1.1.1放出。您能看到数据流在PE3和PE4。然而，在PE3，您能看到速率为(S1,G)作为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
```

有一重复的数据流。此复制是出现的结果数据流(S1,G)在从PE1的分成的MDT和在从PE2的分成的MDT。此秒钟分成的MDT，从PE2，由PE3加入为了获得数据流(S2,G)。但是，因为PE4加入从PE2的分成的MDT为了获得(S1,G)，(S1,G)也是存在从PE2的分成的MDT。因此，PE3接收加入的数据流(S1,G)从两个分成的MDTs。

PE3不能歧视在(S1,G)它的数据包之间从PE1及PE2接收。两数据流在正确RPF接口接收：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	

数据包在不同的流入物理接口能到达在PE3或在同一个接口。无论如何，从不同的数据流的数据包为(S1,G)用一个不同的MPLS标签到达在PE3：

```
PE3#show mpls forwarding-table vrf one
```

Local Label	Outgoing Label	Prefix or Tunnel Id	Bytes Switched	Label	Outgoing interface	Next Hop
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
```

解决方案

解决方案将有更加严格的RPF。使用严格RPF，路由器从哪个邻居检查数据包在RPF接口接收。没有严格RPF，唯一的检查是确定流入接口是否是RPF接口，但是没有，如果数据包从该接口的正确RPF邻居接收。

Cisco IOS的笔记

这是关于RPF的一些重要提示与Cisco IOS。

- 当您更改到/从严格RPF模式时，二者之一配置它，在您配置分成的MDT或clear bgp前。如果只配置严格RPF命令，不会立即创建另一个Lspvif接口。
- 默认情况下严格RPF在Cisco IOS没有启用。
- 不支持它有与默认MDT配置文件的严格RPF命令。

配置

您能配置在PE3的严格RPF虚拟路由和转发的(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
!

```

RPF信息更改：

```

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创建一个Lspvif接口每个入口PE。Lspvif接口创建每个入口PE，每个地址家族(AF)和每个VRF。10.100.1.6的RPF当前指向建立接口Lspvif1，并且10.100.1.7的RPF当前指向建立接口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

现在，数据包的(S1,G) RPF检查从PE1根据RPF接口Lspvif1核对。这些数据包用MPLS标签29进来。数据包的(S2,G) RPF检查从PE2根据RPF接口Lspvif2核对。这些数据包用MPLS标签30进来。数据流在PE3到达通过不同的流入接口，但是这可能也是同一个接口。然而，由于这样的事实mLDP从未使用Penultimate Hop Popping (PHP)，总是有一个正常MPLS标签在组播信息包顶部。到达从PE1和从PE2的(S1,G)数据包在两不同分成的MDTs并且有一个不同的MPLS标签。因此，PE3能歧视在来自PE1和的(S1,G)数据流之间(S1,G)数据流来自PE2。这样，数据包可以由PE3保持单独，并且RPF可以执行不同的入口PE路由器。

在PE3的mLDP数据库当前显示不同的Lspvif接口每个入口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

```

严格RPF或RPF每入口PE工作由于这样的事实组播流进入到入口PE用一个不同的MPLS标签每个入口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

```

严格RPF工作的认证是不再有在PE3 (S1,G)转发的一重复的数据流。重复的数据流在PE3仍然到达，但是丢弃的归结于RPF故障。RPF故障计数器经常在676255和增加以10 pps的速率。

```

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

```

输出速率以PE3当前是20 pps，是每数据流的(S1,G) 10 pps和(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
```

结论

必须用于严格RPF检查使用分成的MDT的mVPN部署模型。

事也许看上去工作，即使您不配置mVPN部署模型的严格RPF检查有分成的MDT的：组播流传送到接收方。然而，有可能性有重复的组播数据流，当来源连接到广泛入口PE路由器时。这导致浪费在网络的带宽，并且能相反影响在接收方的组播应用。因此，它是使用分成的MDT配置mVPN部署模型的严格RPF检查的当务之急。