Le RPF strict vérifient le mVPN

Contenu

[Introduction](#)
[Informations générales](#)
[Problème](#)
[Solution](#)
[Notes pour le Cisco IOS](#)
[Configuration](#)
[Conclusion](#)

Introduction

Ce document décrit la caractéristique stricte du Reverse Path Forwarding (RPF) pour la Multidiffusion au-dessus de VPN (mVPN). Ce document emploie un exemple et l'implémentation dans le Cisco IOS® afin de montrer le comportement.

Informations générales

Le RPF implique que l'interface entrante est vérifiée vers la source. Bien que l'interface soit vérifiée pour déterminer qu'elle est la correcte vers la source, on ne le vérifie pas pour déterminer que c'est le voisin correct RPF sur cette interface. Sur une interface à plusieurs accès, il pourrait y avoir plus d'un voisin auquel vous pourriez RPF. Le résultat pourrait être que le routeur reçoit deux fois le même flot de Multidiffusion sur cette interface et en avant chacun des deux.

Dans les réseaux où le Protocol Independent Multicast (PIM) fonctionne sur l'interface à plusieurs accès, ce n'est pas une question, parce que le flot en double de Multidiffusion fait fonctionner le mécanisme d'affirmation et un flot de Multidiffusion ne sera plus reçu. Dans certains cas, PIM ne fonctionne pas sur l'arbre de distribution de Multidiffusion (MDT), qui est une interface à plusieurs accès. Dans des ces cas, le Protocole BGP (Border Gateway Protocol) est le protocole de signalisation de recouvrement.

Dans les profils avec MDT divisé, même si PIM fonctionne comme protocole de recouvrement, il peut être impossible à avoir affirme. La raison pour ceci est qu'un Provider Edge d'entrée (PE) ne joint pas le MDT divisé d'un autre PE d'entrée dans les scénarios où il y a deux Routeurs ou plus de PE d'entrée. Chaque routeur PE d'entrée peut expédier le flot de Multidiffusion sur son MDT divisé sans l'autre routeur PE d'entrée voyant le trafic de multidiffusion. Le fait que deux Routeurs différents chacun de PE de sortie joignent un MDT vers un différent routeur PE d'entrée pour le même flot de Multidiffusion est un scénario valide : ce s'appelle Anycast Source. Ceci permet à différents récepteurs pour joindre le même flot de Multidiffusion mais au-dessus d'un différent chemin dans le noyau de Commutation multiprotocole par étiquette (MPLS). Voir la figure 1 pour un exemple de source de cantonade.

Figure 1

Il y a deux Routeurs de PE d'entrée : PE1 et PE2. Il y a deux Routeurs de PE de sortie : PE3 et PE4. Chaque routeur PE de sortie a un différent routeur PE d'entrée en tant que son voisin RPF. PE3 a PE1 en tant que son voisin RPF. PE4 a PE2 en tant que son voisin RPF. Les Routeurs de PE de sortie sélectionnent leur routeur PE d'entrée plus étroit en tant que leur voisin RPF.

Le flot (S1,G) ira du S1 au récepteur 1 au-dessus du chemin supérieur et du S1 au récepteur 2 au-dessus du chemin inférieur. Il n'y a aucune intersection des deux flots au-dessus des deux chemins (chaque chemin dans le noyau MPLS est un MDT divisé différent).

Si le MDT était un par défaut MDT - tel que dedans dedans les profils du par défaut MDT - puis ceci ne fonctionnerait pas parce que les deux flots de Multidiffusion seraient sur le même par défaut MDT et le mécanisme d'affirmation fonctionnerait. Si le MDT est des données MDT dans les profils du par défaut MDT, alors tous les Routeurs de PE d'entrée joignent les données MDT des autres Routeurs de PE d'entrée et car tels voient le trafic de multidiffusion entre eux et du mécanisme d'affirmation s'exécutent de nouveau. Si le protocole de recouvrement est BGP, alors il y a sélection en amont du saut de Multidiffusion (UMH) et seulement un routeur PE d'entrée est sélectionné comme expéditeur, mais c'est par MDT.

La source de cantonade est l'un des grands avantages de MDT divisé par exécution.

Problème

Le contrôle du militaire de carrière RPF confirme que les paquets arrivent au routeur de l'interface correcte RPF. Il n'y a aucun contrôle à confirmer que les paquets sont reçus du voisin correct RPF sur cette interface.

Voir la figure 2. Il affiche à une question où le trafic de doublon est constamment expédié dans un scénario avec MDT divisé. Il prouve que le contrôle du militaire de carrière RPF dans le cas de MDT divisé n'est pas suffisant afin d'éviter le trafic en double.

Figure 2

Il y a deux récepteurs. Le premier récepteur est installé pour recevoir le trafic pour (S1,G) et (S2,G). Le deuxième récepteur est installé pour recevoir le trafic pour (S2,G) seulement. Il y a de MDT divisés, et le BGP est le protocole de signalisation de recouvrement. Notez que la source S1 est accessible par l'intermédiaire de PE1 et de PE2. Le protocole de noyau-arborescence est protocole de distribution d'étiquette multipoint (mLDP).

Chaque routeur PE annonce une artère de mVPN d'ipv4 BGP de type 1, qui indique que c'est un candidat à être la racine d'un MDT divisé.

```
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 trouve PE1 en tant que voisin RPF pour le S1 après qu'une consultation pour l'artère d'unicast pour le 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 sélectionne PE1 en tant que voisin RPF pour (S1,G) et joint le MDT divisé avec PE1 comme racine. PE3 sélectionne PE2 en tant que voisin RPF pour (S2,G) et joint le MDT divisé avec PE2 comme racine.

```
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 sélectionne PE2 en tant que voisin RPF pour (S1,G) et joint le MDT divisé avec PE1 comme racine.

```
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
```

Notez que l'interface RPF est Lspvif0 pour S1 (10.100.1.6) et S2 (10.100.1.7).

PE3 joint le MDT divisé de PE2 pour (S2,G), et PE4 joint le MDT divisé de PE2 pour (S1,G). PE1 joint le MDT divisé de PE1 pour (S1,G). Vous pouvez voir ceci par les artères de mVPN d'ipv4 BGP du type 7 reçues sur PE1 et 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 ?
```

Les entrées multicasts sur PE3 et 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
```

Ceci prouve que PE3 joint l'arborescence (P2MP) point-à-multipoint enracinée à PE1 et également l'arborescence enracinée à 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
```

Ceci prouve que PE4 joint l'arborescence P2MP enracinée à 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
```

Le S1 et les S2 coulent pour le groupe 232.1.1.1 avec 10 PPS. Vous pouvez voir les flots à PE3 et

à PE4. Cependant, à PE3, vous pouvez voir le débit pour (S1,G) en tant que 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
```

Il y a un flot en double. Cette duplication est le résultat de la présence du flot (S1,G) sur le MDT divisé de PE1 et sur le MDT divisé de PE2. Ce MDT divisé par seconde, de PE2, a été joint par PE3 afin d'obtenir le flot (S2,G). Mais, parce que PE4 a joint le MDT divisé de PE2 afin d'obtenir (S1,G), (S1,G) est également présent sur le MDT divisé de PE2. Par conséquent, PE3 reçoit le flot (S1,G) de chacun des deux MDTs divisé qu'il s'est joint.

PE3 ne peut pas distinguer entre les paquets pour (S1,G) lui reçoit de PE1 et de PE2. Les deux flots sont reçus sur l'interface correcte RPF : 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
```

Les paquets ont pu arriver sur différentes interfaces physiques entrantes sur PE3 ou sur la même interface. En tous cas, les paquets des différents flots pour (S1,G) arrivent avec des mpls label différents à 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

La solution est d'avoir un RPF plus strict. Avec le RPF strict, le routeur vérifie de quel voisin les paquets sont reçu sur l'interface RPF. Sans RPF strict, le seul contrôle est de déterminer si l'interface entrante est l'interface RPF, mais pas si les paquets sont reçus du voisin correct RPF sur cette interface.

Notes pour le Cisco IOS

Voici quelques informations importantes au sujet de RPF avec le Cisco IOS.

- Quand vous changez à/de le mode strict RPF, l'un ou l'autre le configure avant que vous configuriez le MDT ou le clear bgp divisé. Si vous configurez seulement la commande RPF Stricte, elle ne créera pas une autre interface de Lspvif immédiatement.
- Le RPF strict n'est pas activé par défaut dans le Cisco IOS.
- Il n'est pas pris en charge pour avoir la commande RPF avec des profils par défaut MDT.

Configuration

Vous pouvez configurer le RPF strict sur PE3 pour le 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
!
```

Les informations RPF ont changé :

```
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 a créé une interface de Lspvif par PE d'entrée. L'interface de Lspvif est créée par PE d'entrée, par famille d'adresse (AF), et par VRF. Le RPF pour 10.100.1.6 se dirige maintenant pour relier Lspvif1 et le RPF pour 10.100.1.7 se dirige maintenant pour relier 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
```

Maintenant, le RPF vérifient les paquets (S1,G) de PE1 sont vérifiés contre l'interface Lspvif1 RPF. Ces paquets entrés avec les mpls label 29. Le RPF vérifient les paquets (S2,G) de PE2 sont vérifiés contre l'interface Lspvif2 RPF. Ces paquets entrés avec les mpls label 30. Les flots arrivent sur PE3 par différentes interfaces entrantes, mais ceci pourrait également être la même interface. Cependant, étant donné que le mLDP n'utilise jamais Pénultième-Saut-sauter (PHP), il y a toujours des mpls label réguliers sur les paquets de multidiffusion. Les paquets (S1,G) qui arrivent de PE1 et de PE2 sont sur deux MDTs divisé différent et ont donc des mpls label différents. Par conséquent, PE3 peut distinguer entre le flot (S1,G) qui provient PE1 et le flot (S1,G) qui provient PE2. De cette façon, les paquets peuvent être maintenus distants par PE3 et un RPF peut être exécuté contre différents Routeurs de PE d'entrée.

La base de données de mLDP sur PE3 affiche maintenant les différentes interfaces de Lspvif par PE d'entrée.

```
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
```

RPF strict ou RPF par travaux de PE d'entrée étant donné que les flots de Multidiffusion entrés au PE d'entrée avec des mpls label différents par PE d'entrée :

```
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
```

La preuve que le RPF strict fonctionne est qu'il n'y a plus un flot en double (S1,G) expédié sur PE3. Le flot en double arrive toujours sur PE3, mais il est dû lâché à la panne RPF. Le compteur de panne RPF est à 676255 et à augmentations constamment à un taux de 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
```

Le débit sortant à PE3 est maintenant de 20 PPS, qui est de 10 PPS pour chaque flot (S1,G) et (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

Le contrôle strict RPF doit être utilisé pour les modèles de déploiement de mVPN qui utilisent MDT divisé.

Les choses pourraient sembler fonctionner, même si vous ne configurez pas le RPF strict vérifiez les modèles de déploiement de mVPN avec MDT divisé : les flots de Multidiffusion sont livrés aux récepteurs. Cependant, il y a la possibilité qu'il y a du trafic de multidiffusion en double quand des

sources sont connectées à de plusieurs Routeurs de PE d'entrée. Ceci mène à un gaspillage de bande passante dans le réseau et peut compromettre l'application de Multidiffusion sur les récepteurs. Par conséquent, c'est une nécessité pour configurer le RPF strict vérifient les modèles de déploiement de mVPN qui utilise MDT divisé.