

Contrôle RPF strict pour mVPN

Contenu

[Introduction](#)

[Informations générales](#)

[Problème](#)

[Solution](#)

[Notes pour Cisco IOS](#)

[Configuration](#)

[Conclusion](#)

Introduction

Ce document décrit la fonctionnalité RPF (Strict Reverse Path Forwarding) pour la multidiffusion sur VPN (mVPN). Ce document utilise un exemple et la mise en oeuvre dans Cisco IOS[®] afin d'illustrer le comportement.

Informations générales

RPF implique que l'interface entrante est vérifiée vers la source. Bien que l'interface soit vérifiée pour déterminer qu'elle est la bonne vers la source, elle n'est pas vérifiée pour déterminer qu'elle est le voisin RPF correct sur cette interface. Sur une interface à accès multiple, il peut y avoir plusieurs voisins auxquels vous pouvez accéder au protocole RPF. Il peut en résulter que le routeur reçoit deux fois le même flux de multidiffusion sur cette interface et transfère les deux.

Dans les réseaux où le protocole PIM (Protocol Independent Multicast) s'exécute sur l'interface à accès multiple, ce n'est pas un problème, car le flux de multidiffusion dupliqué entraîne l'exécution du mécanisme d'assertion et un flux de multidiffusion ne sera plus reçu. Dans certains cas, PIM ne fonctionne pas sur l'arbre de distribution multidiffusion (MDT), qui est une interface à accès multiple. Dans ces cas, le protocole BGP (Border Gateway Protocol) est le protocole de signalisation de superposition.

Dans les profils avec MDT partitionné, même si PIM fonctionne comme protocole de superposition, il peut être impossible d'avoir des assertions. Cela s'explique par le fait qu'une périphérie de fournisseur d'entrée (PE) ne rejoint pas le MDT partitionné d'un autre PE d'entrée dans les scénarios où il y a deux routeurs PE d'entrée ou plus. Chaque routeur PE d'entrée peut transférer le flux de multidiffusion sur son MDT partitionné sans que l'autre routeur PE d'entrée ne voie le trafic de multidiffusion. Le fait que deux routeurs PE de sortie différents rejoignent chacun un MDT vers un routeur PE de sortie différent pour le même flux de multidiffusion est un scénario valide : il s'appelle Anycast Source. Cela permet à différents récepteurs de rejoindre le même flux de multidiffusion mais sur un chemin différent dans le coeur de la commutation multiprotocole par étiquette (MPLS). Reportez-vous à la Figure 1 pour un exemple de source Anycast.

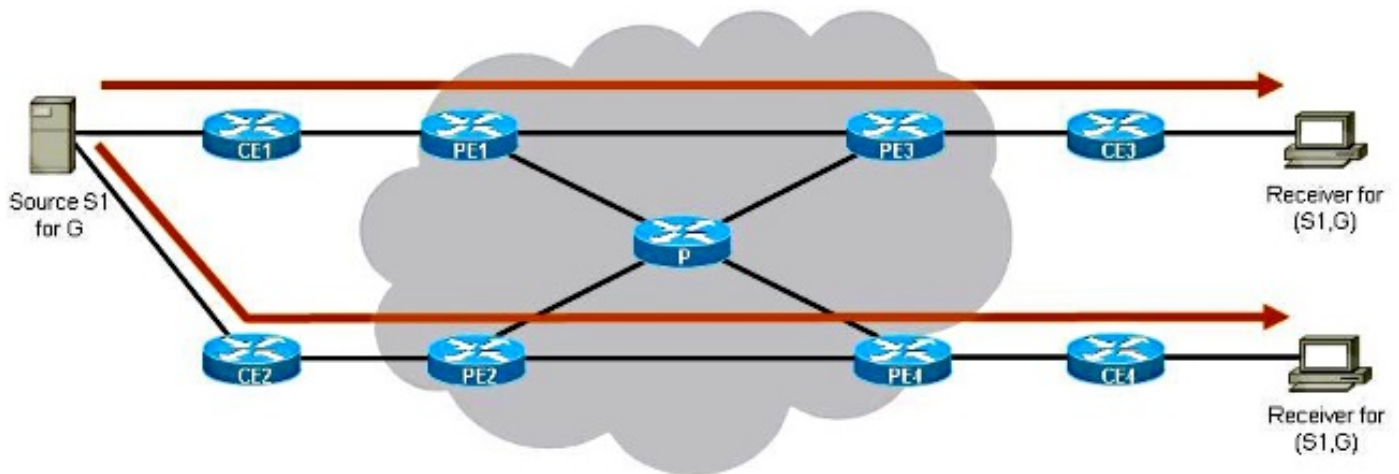


Figure 1

Il existe deux routeurs PE d'entrée : PE1 et PE2. Il existe deux routeurs PE de sortie : PE3 et PE4. Chaque routeur PE de sortie a un routeur PE de sortie différent de son voisin RPF. PE3 a PE1 comme voisin RPF. PE4 a PE2 comme voisin RPF. Les routeurs PE de sortie choisissent leur routeur PE de sortie le plus proche comme voisin RPF.

Le flux (S1, G) va de S1 au récepteur 1 sur le chemin supérieur et de S1 au récepteur 2 sur le chemin inférieur. Il n'y a pas d'intersection des deux flux sur les deux chemins (chaque chemin dans le coeur de MPLS est un MDT partitionné différent).

Si le MDT était un MDT par défaut (comme dans les profils MDT par défaut), cela ne fonctionnerait pas car les deux flux de multidiffusion se trouveraient sur le même MDT par défaut et le mécanisme d'assertion s'exécuterait. Si le MDT est un MDT de données dans les profils MDT par défaut, tous les routeurs PE d'entrée joignent le MDT de données des autres routeurs PE d'entrée et, à ce titre, voient le trafic de multidiffusion les uns des autres et le mécanisme d'assertion s'exécute à nouveau. Si le protocole de superposition est BGP, il y a la sélection UMH (Upstream Multicast Hop) et un seul routeur PE d'entrée est sélectionné comme redirecteur, mais c'est par MDT.

Anycast Source est l'un des principaux avantages de l'exécution de Partited MDT.

Problème

Le contrôle RPF régulier confirme que les paquets arrivent au routeur à partir de l'interface RPF correcte. Il n'y a aucune vérification pour confirmer que les paquets sont reçus du voisin RPF correct sur cette interface.

Voir la figure 2. Il montre un problème où le trafic dupliqué est transmis de façon persistante dans un scénario avec MDT partitionné. Il montre que le contrôle RPF régulier dans le cas de MDT partitionné n'est pas suffisant pour éviter les trafics en double.

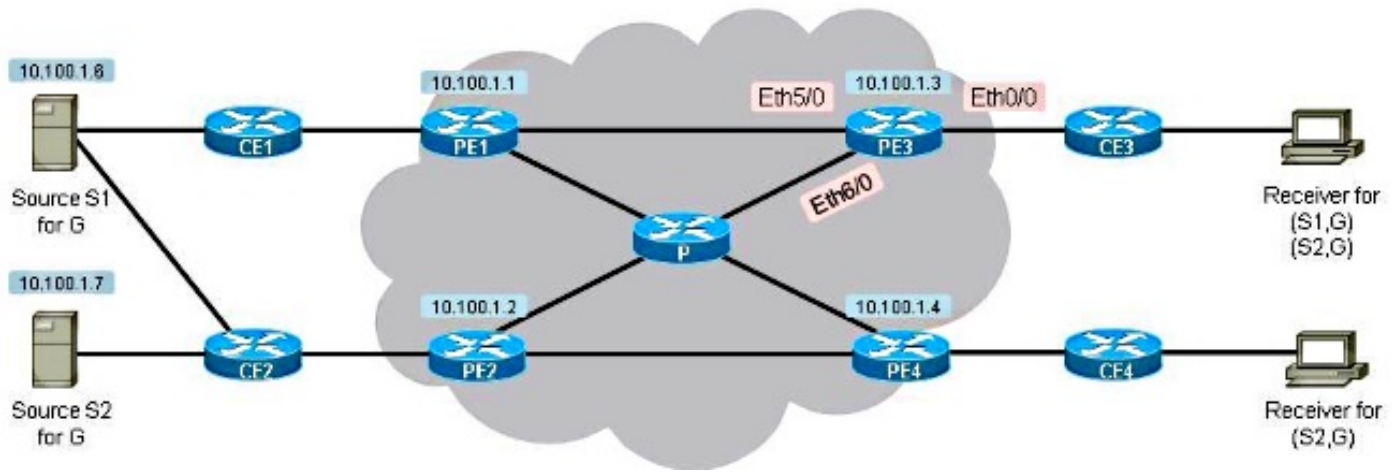


Figure 2

Il y a deux récepteurs. Le premier récepteur est configuré pour recevoir le trafic pour (S1, G) et (S2, G). Le second récepteur est configuré pour recevoir le trafic pour (S2, G) uniquement. Il y a MDT partitionné et BGP est le protocole de signalisation de superposition. Notez que la source S1 est accessible via PE1 et PE2. Le protocole de l'arbre principal est le protocole mLDP (Multipoint Label Distribution Protocol).

Chaque routeur PE annonce une route mVPN IPv4 BGP de type 1, qui indique qu'il est candidat pour être la racine d'un MDT partitionné.

```
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 comme voisin RPF pour S1 après une recherche de la route de monodiffusion pour 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 comme voisin RPF pour (S1, G) et joint le MDT partitionné avec PE1 comme racine. PE3 sélectionne PE2 comme voisin RPF pour (S2, G) et joint le MDT partitionné 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 comme voisin RPF pour (S1, G) et joint le MDT partitionné 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 rejoint le MDT partitionné de PE2 pour (S2, G) et PE4 rejoint le MDT partitionné de PE2 pour (S1, G). PE1 rejoint le MDT partitionné de PE1 pour (S1, G). Vous pouvez le voir par les routes mVPN IPv4 BGP de 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 de multidiffusion 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 montre que PE3 rejoint l'arborescence point à multipoint (P2MP) enracinée à PE1 et aussi 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 montre que PE4 rejoint 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
```

Flux S1 et S2 pour le groupe 232.1.1.1 avec 10 pps. Vous pouvez voir les flux sur PE3 et PE4. Cependant, à PE3, vous pouvez voir le taux pour (S1, G) comme 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 existe un flux dupliqué. Cette duplication est le résultat de la présence de flux (S1, G) sur le MDT partitionné de PE1 et sur le MDT partitionné de PE2. Ce deuxième MDT partitionné, de PE2, a été rejoint par PE3 afin d'obtenir le flux (S2, G). Mais, parce que PE4 a rejoint le MDT partitionné de PE2 afin d'obtenir (S1, G), (S1, G) est également présent sur le MDT partitionné de PE2. Par conséquent, PE3 reçoit le flux (S1, G) des deux MDT partitionnés qu'il a joints.

PE3 ne peut pas distinguer les paquets pour (S1, G) qu'il reçoit de PE1 et PE2. Les deux flux sont reçus sur l'interface RPF correcte : 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 peuvent arriver sur différentes interfaces physiques entrantes sur PE3 ou sur la même interface. Dans tous les cas, les paquets des différents flux pour (S1, G) arrivent avec une étiquette MPLS différente à 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

Solution

La solution est d'avoir un RPF plus strict. Avec le protocole RPF strict, le routeur vérifie à partir de quel voisin les paquets sont reçus sur l'interface RPF. Sans RPF strict, la seule vérification consiste à déterminer si l'interface entrante est l'interface RPF, mais pas si les paquets sont reçus du voisin RPF correct sur cette interface.

Notes pour Cisco IOS

Voici quelques notes importantes sur RPF avec Cisco IOS.

- Lorsque vous passez en mode RPF strict, configurez-le avant de configurer le MDT partitionné ou effacez le BGP. Si vous configurez uniquement la commande RPF stricte, elle

ne créera pas immédiatement une autre interface Lspvif.

- RPF strict n'est pas activé par défaut dans Cisco IOS.
- Il n'est pas pris en charge d'avoir la commande **strict-rpf** avec les profils MDT par défaut.

Configuration

Vous pouvez configurer un RPF strict sur PE3 pour le routage et transfert virtuels (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 Lspvif par PE d'entrée. L'interface Lspvif est créée par PE d'entrée, par famille d'adresses (AF) et par VRF. Le RPF pour 10.100.1.6 pointe maintenant vers l'interface Lspvif1 et le RPF pour 10.100.1.7 pointe maintenant vers l'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

Maintenant, le contrôle RPF des paquets (S1, G) de PE1 est vérifié par rapport à l'interface RPF Lspvif1. Ces paquets sont fournis avec l'étiquette MPLS 29. Le contrôle RPF des paquets (S2, G)

de PE2 est vérifié par rapport à l'interface RPF Lspvif2. Ces paquets sont fournis avec l'étiquette MPLS 30. Les flux arrivent sur PE3 via différentes interfaces entrantes, mais il peut également s'agir de la même interface. Cependant, étant donné que mLDP n'utilise jamais le protocole PHP, il existe toujours une étiquette MPLS régulière au-dessus des paquets de multidiffusion. Les paquets (S1, G) qui arrivent de PE1 et de PE2 se trouvent sur deux MDT partitionnés différents et ont donc une étiquette MPLS différente. Par conséquent, PE3 peut discriminer entre le flux (S1, G) qui provient de PE1 et le flux (S1, G) qui provient de PE2. De cette manière, les paquets peuvent être séparés par PE3 et un RPF peut être exécuté sur différents routeurs PE d'entrée.

La base de données mLDP sur PE3 présente désormais les différentes interfaces 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 ou RPF strict par PE d'entrée fonctionne en raison du fait que les flux de multidiffusion arrivent au PE d'entrée avec une étiquette MPLS différente par PE d'entrée :

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]	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 RPF strict fonctionne est qu'il n'y a plus de flux dupliqué (S1, G) transféré sur PE3.

Le flux dupliqué arrive toujours sur PE3, mais il est abandonné en raison d'une défaillance RPF. Le compteur de défaillance RPF est de 676255 et augmente 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 de sortie de PE3 est maintenant de 20 pps, soit 10 pps pour chaque flux (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 RPF strict doit être utilisé pour les modèles de déploiement mVPN qui utilisent le MDT partitionné.

Les choses peuvent sembler fonctionner, même si vous ne configurez pas le contrôle strict RPF pour les modèles de déploiement mVPN avec MDT partitionné : les flux de multidiffusion sont transmis aux récepteurs. Cependant, il est possible qu'il y ait un trafic de multidiffusion en double lorsque des sources sont connectées à plusieurs routeurs PE d'entrée. Cela entraîne une perte de bande passante sur le réseau et peut affecter l'application de multidiffusion sur les récepteurs. Par conséquent, il est nécessaire de configurer un contrôle RPF strict pour les modèles de déploiement mVPN qui utilisent le MDT partitionné.