

Controllo rigoroso RPF per mVPN

Sommario

[Introduzione](#)

[Premesse](#)

[Problema](#)

[Soluzione](#)

[Note per Cisco IOS](#)

[Configurazione](#)

[Conclusioni](#)

Introduzione

Questo documento descrive la funzione RPF (Strict Reverse Path Forwarding) per Multicast over VPN (mVPN). Per illustrarne il comportamento, questo documento utilizza un esempio e l'implementazione in Cisco IOS®.

Premesse

RPF implica che l'interfaccia in ingresso viene controllata verso l'origine. Anche se l'interfaccia viene controllata per determinare che sia quella corretta verso l'origine, non viene verificata per determinare che sia la porta RPF adiacente corretta su quell'interfaccia. Su un'interfaccia ad accesso multiplo, è possibile utilizzare più router adiacenti per RPF. Di conseguenza, il router potrebbe ricevere il doppio dello stesso flusso multicast su quell'interfaccia e inoltrare entrambi.

Nelle reti in cui il protocollo PIM (Protocol Independent Multicast) viene eseguito sull'interfaccia di accesso multiplo, questo non è un problema, in quanto il flusso multicast duplicato causa l'esecuzione del meccanismo di asserzione e la ricezione di un flusso multicast non viene più effettuata. In alcuni casi, PIM non viene eseguito nell'albero di distribuzione multicast (MDT), che è un'interfaccia ad accesso multiplo. In questi casi, il Border Gateway Protocol (BGP) è il protocollo di segnalazione della sovrimpressione.

Nei profili con MDT partizionato, anche se PIM viene eseguito come protocollo di overlay, può essere impossibile avere asserzioni. Ciò si verifica perché un server PE (Ingress Provider Edge) non si unisce al database MDT partizionato da un altro server PE in ingresso negli scenari in cui sono presenti due o più router PE in ingresso. Ogni router PE in ingresso è in grado di inoltrare il flusso multicast al relativo MDT partizionato senza che l'altro router PE in ingresso rilevi il traffico multicast. Il fatto che due router PE in uscita diversi si uniscano a un MDT verso un router PE in entrata diverso per lo stesso flusso multicast è uno scenario valido: si chiama Anycast Source. Ciò consente a ricevitori diversi di unirsi allo stesso flusso multicast ma su un percorso diverso nel nucleo MPLS (Multiprotocol Label Switching). Vedere la Figura 1 per un esempio di Anycast Source.

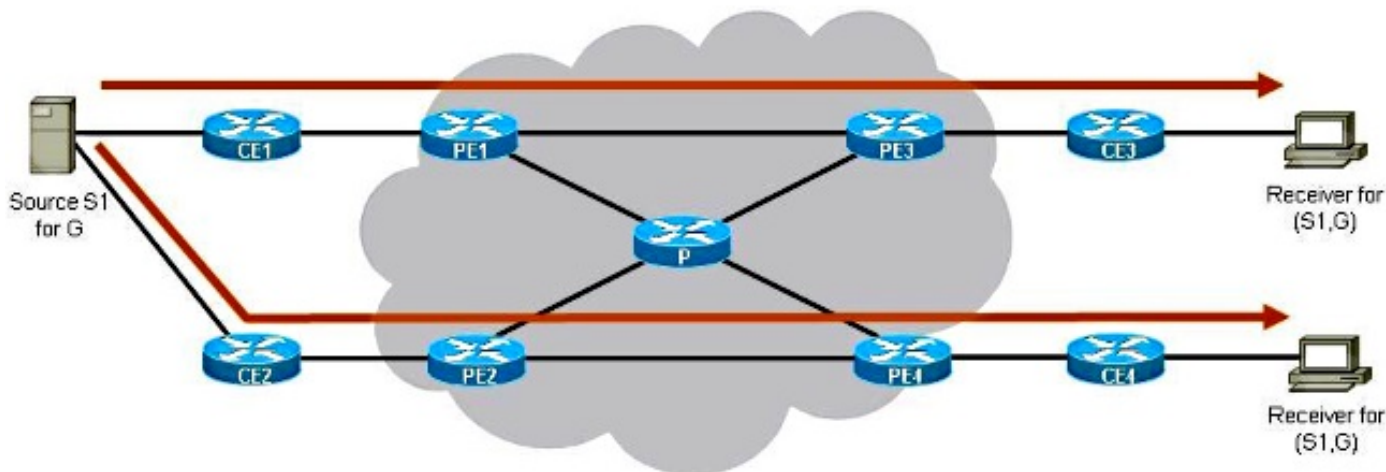


Figura 1

Sono disponibili due router PE in ingresso: PE1 e PE2. Sono disponibili due router PE in uscita: PE3 e PE4. Ogni router PE in uscita ha un router PE in entrata diverso come router adiacente RPF. PE3 ha PE1 come router adiacente RPF. PE4 ha PE2 come router adiacente RPF. I router PE in uscita scelgono il router PE in entrata più vicino come router adiacente di RPF.

Lo streaming (S1,G) va da S1 a Receiver 1 sul percorso superiore e da S1 a Receiver 2 sul percorso inferiore. Non esiste alcuna intersezione dei due flussi sui due percorsi (ogni percorso nel core MPLS è un MDT partizionato diverso).

Se l'MDT è un MDT predefinito, come nei profili MDT predefiniti, non funzionerà perché i due flussi multicast si troveranno sullo stesso MDT predefinito e verrà eseguito il meccanismo di asserzione. Se l'MDT è un MDT dati nei profili MDT predefiniti, tutti i router PE in ingresso si uniscono all'MDT dati dagli altri router PE in ingresso e come tale vedono il traffico multicast tra loro e il meccanismo di asserzione viene eseguito di nuovo. Se il protocollo di sovrapposizione è BGP, è presente la selezione UMH (Upstream Multicast Hop) e come server di inoltro viene selezionato un solo router PE in ingresso, ma questo è per MDT.

Anycast Source è uno dei grandi vantaggi dell'esecuzione di Partitioned MDT.

Problema

Il controllo regolare di RPF conferma che i pacchetti arrivano al router dall'interfaccia RPF corretta. Non è possibile verificare che i pacchetti vengano ricevuti dal router adiacente RPF corretto su quell'interfaccia.

Vedere la Figura 2. Mostra un problema in cui il traffico duplicato viene inoltrato in modo permanente in uno scenario con MDT partizionato. Esso dimostra che il controllo regolare di RPF nel caso di MDT partizionato non è sufficiente per evitare traffico duplicato.

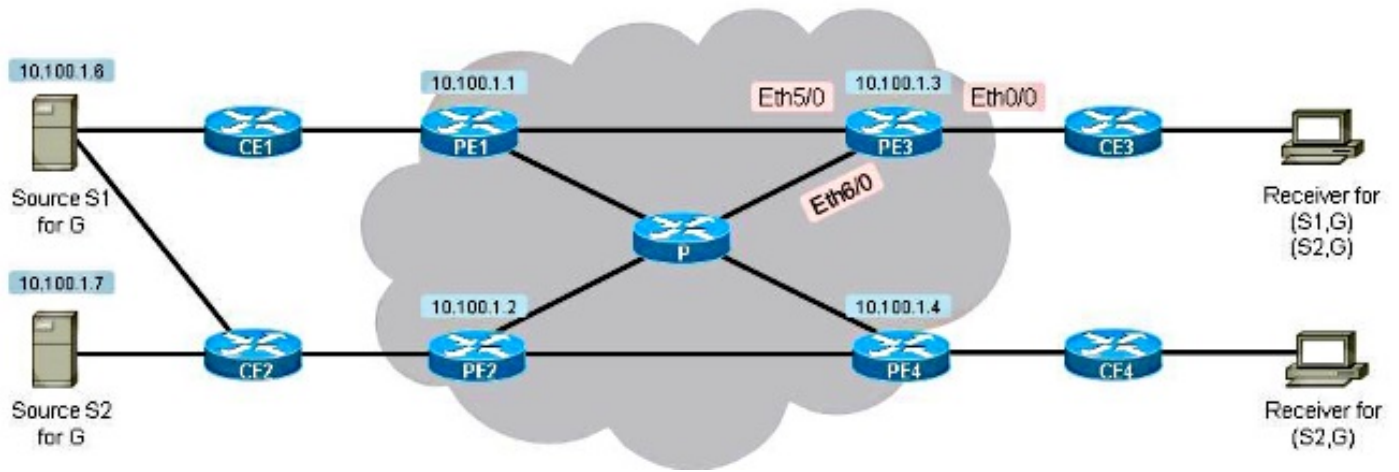


Figura 2

Ci sono due ricevitori. Il primo ricevitore è configurato per la ricezione del traffico per (S1,G) e (S2,G). Il secondo ricevitore è configurato per ricevere traffico solo per (S2,G). È presente Partitioned MDT, e BGP è il protocollo di segnalazione della sovrimpressione. Si noti che l'origine S1 è raggiungibile sia tramite PE1 che PE2. Il protocollo dell'albero principale è Multipoint Label Distribution Protocol (mLDP).

Ogni router PE annuncia una route mVPN BGP IPv4 di tipo 1, che indica che è un candidato per essere la radice di un MDT partizionato.

```
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 trova PE1 come router adiacente RPF per S1 dopo una ricerca della route unicast per 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 seleziona PE1 come RPF adiacente per (S1,G) e unisce l'MDT partizionato con PE1 come root. PE3 seleziona PE2 come RPF adiacente per (S2,G) e unisce l'MDT partizionato con PE2 come radice.

```
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 seleziona PE2 come RPF adiacente per (S1,G) e unisce l'MDT partizionato con PE1 come radice.

```
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

Notare che l'interfaccia RPF è Lspvif0 sia per S1 (10.100.1.6) che per S2 (10.100.1.7).

PE3 unisce il MDT partizionato da PE2 per (S2,G) e PE4 unisce il MDT partizionato da PE2 per (S1,G). PE1 unisce il MDT partizionato da PE1 per (S1,G). Ciò è possibile in base alle route mVPN IPv4 BGP di tipo 7 ricevute in PE1 e 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	?

Voci multicast in PE3 e 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

Ciò mostra che PE3 unisce l'albero Punto-Multipoint (P2MP) con radice in PE1 e l'albero con radice in 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

```

Ciò mostra che PE4 si unisce alla struttura P2MP con radice in 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

```

Flusso S1 e S2 per il gruppo 232.1.1.1 con 10 p/s. È possibile visualizzare i flussi in PE3 e PE4. Tuttavia, in PE3, è possibile visualizzare la velocità per (S1,G) come 20 punti percentuali.

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
```

Flusso duplicato. Questa duplicazione è il risultato della presenza di un flusso (S1,G) nel MDT partizionato da PE1 e nel MDT partizionato da PE2. Questo secondo MDT partizionato, da PE2, è stato unito da PE3 per ottenere il flusso (S2,G). Tuttavia, poiché PE4 ha unito l'MDT partizionato da PE2 per ottenere (S1,G), (S1,G) è presente anche nell'MDT partizionato da PE2. PE3 riceve quindi il flusso (S1,G) da entrambi gli MDT partizionati a cui è stato unito.

PE3 non può distinguere tra i pacchetti per (S1,G) che riceve da PE1 e PE2. Entrambi i flussi vengono ricevuti sull'interfaccia RPF corretta: 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	

I pacchetti potrebbero arrivare su diverse interfacce fisiche in ingresso su PE3 o sulla stessa interfaccia. In ogni caso, i pacchetti dei diversi flussi per (S1,G) arrivano con un'etichetta MPLS diversa in 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
```

Soluzione

La soluzione è avere un RPF più rigido. Con il protocollo RPF rigido, il router controlla da quale router adiacente vengono ricevuti i pacchetti sull'interfaccia RPF. Senza un RPF rigido, l'unico controllo da eseguire è quello di determinare se l'interfaccia in entrata è l'interfaccia RPF, ma non se i pacchetti vengono ricevuti dal router adiacente RPF corretto su quell'interfaccia.

Note per Cisco IOS

Di seguito sono riportate alcune note importanti su RPF con Cisco IOS.

- Quando si passa alla/dalla modalità RPF rigida, configurarla prima di configurare l'MDT partizionato o cancellare il contenuto di BGP. Se si configura solo il comando rigoroso RPF, non verrà creata immediatamente un'altra interfaccia Lspvif.

- RPF rigoroso non è abilitato per impostazione predefinita in Cisco IOS.
- L'uso del comando **strict-rpf** con profili MDT predefiniti non è supportato.

Configurazione

In PE3 è possibile configurare RPF rigide per il VRF (Virtual Routing and Forwarding).

```
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
!
```

Le informazioni RPF sono cambiate:

```
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
```

In PE3 è stata creata un'interfaccia Lspvif per ogni PE in ingresso. L'interfaccia Lspvif viene creata per PE in ingresso, per AF (Address Family) e per VRF. L'RPF per 10.100.1.6 ora punta all'interfaccia Lspvif1 e l'RPF per 10.100.1.7 ora punta all'interfaccia 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

A questo punto, il controllo RPF per i pacchetti (S1,G) provenienti da PE1 viene eseguito sull'interfaccia RPF Lspvif1. Questi pacchetti vengono forniti con l'etichetta MPLS 29. Il controllo RPF per i pacchetti (S2,G) provenienti da PE2 viene eseguito sull'interfaccia RPF Lspvif2. Questi pacchetti vengono forniti con l'etichetta MPLS 30. I flussi arrivano su PE3 attraverso diverse

interfacce in ingresso, ma questa potrebbe anche essere la stessa interfaccia. Tuttavia, poiché mLDP non usa mai il Penultimate-Hop-Popping (PHP), sui pacchetti multicast è sempre presente un'etichetta MPLS regolare. I pacchetti (S1,G) provenienti da PE1 e PE2 si trovano su due MDT partizionati diversi e pertanto hanno un'etichetta MPLS diversa. Pertanto, PE3 può distinguere tra il flusso (S1,G) che proviene da PE1 e il flusso (S1,G) che proviene da PE2. In questo modo, i pacchetti possono essere separati da PE3 e un RPF può essere eseguito su router PE in ingresso diversi.

Il database mLDP in PE3 ora mostra le diverse interfacce Lspvif per ogni PE in ingresso.

```
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 o RPF rigorosi per PE in ingresso funzionano perché i flussi multicast vengono inviati al PE in ingresso con un'etichetta MPLS diversa per PE in ingresso:

```
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 prova che un RPF rigido funziona è che non esiste più un flusso duplicato (S1,G) inoltrato su PE3. Il flusso duplicato arriva ancora su PE3, ma viene eliminato a causa dell'errore di RPF. Il contatore degli errori RPF è a 676255 e aumenta costantemente a una velocità di 10 punti

percentuali.

```
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
```

La velocità di output in PE3 è ora di 20 pps, ovvero 10 pps per ogni flusso (S1,G) e (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
```

Conclusioni

Per i modelli di distribuzione mVPN che utilizzano MDT partizionato è necessario utilizzare il controllo rigoroso delle cartelle pubbliche (RPF).

È possibile che le funzionalità funzionino, anche se non si configura il controllo rigoroso di RPF per i modelli di distribuzione mVPN con MDT partizionato: i flussi multicast vengono consegnati ai ricevitori. Tuttavia, esiste la possibilità che vi sia traffico multicast duplicato quando le origini sono connesse a più router PE in ingresso. Ciò comporta uno spreco di larghezza di banda nella rete e può influire negativamente sull'applicazione multicast sui ricevitori. Pertanto, è necessario configurare un rigoroso controllo RPF per i modelli di distribuzione mVPN che utilizzano MDT partizionato.