Verificación RPF estricta para mVPN

Contenido

Introducción

Este documento describe la función Estricto Reverse Path Forwarding (RPF) para multidifusión sobre VPN (mVPN). Este documento utiliza un ejemplo y la implementación en Cisco IOS® para ilustrar el comportamiento.

Antecedentes

RPF implica que la interfaz entrante se verifica hacia el origen. Aunque la interfaz se verifica para determinar que es la correcta hacia el origen, no se verifica para determinar que es el vecino RPF correcto en esa interfaz. En una interfaz de acceso múltiple, podría haber más de un vecino al que podría RPF. El resultado podría ser que el router recibe el doble de la misma secuencia multicast en esa interfaz y reenvía ambos.

En las redes donde se ejecuta Protocol Independent Multicast (PIM) en la interfaz de acceso múltiple, esto no es un problema, porque el flujo de multidifusión duplicado hace que se ejecute el mecanismo de aserción y ya no se recibirá una secuencia de multidifusión. En algunos casos, PIM no se ejecuta en el árbol de distribución de multidifusión (MDT), que es una interfaz de acceso múltiple. En estos casos, el protocolo de señalización de la superposición es el protocolo de gateway fronterizo (BGP).

En los perfiles con MDT particionado, incluso si PIM se ejecuta como protocolo de superposición, puede ser imposible tener aserciones. La razón de esto es que un extremo del proveedor de entrada (PE) no se une al MDT particionado de otro PE de entrada en los escenarios donde hay dos o más routers PE de entrada. Cada router PE de entrada puede reenviar el flujo multicast a su MDT particionado sin que el otro router PE de entrada vea el tráfico multicast. El hecho de que dos routers PE de salida diferentes se unan a un MDT hacia un router PE de entrada diferente para el mismo flujo multicast es un escenario válido: se denomina Anycast Source. Esto permite que diferentes receptores se unan a la misma secuencia de multidifusión pero a través de una ruta diferente en el núcleo MPLS (Multiprotocol Label Switching). Consulte la Figura 1 para ver un ejemplo de Anycast Source.

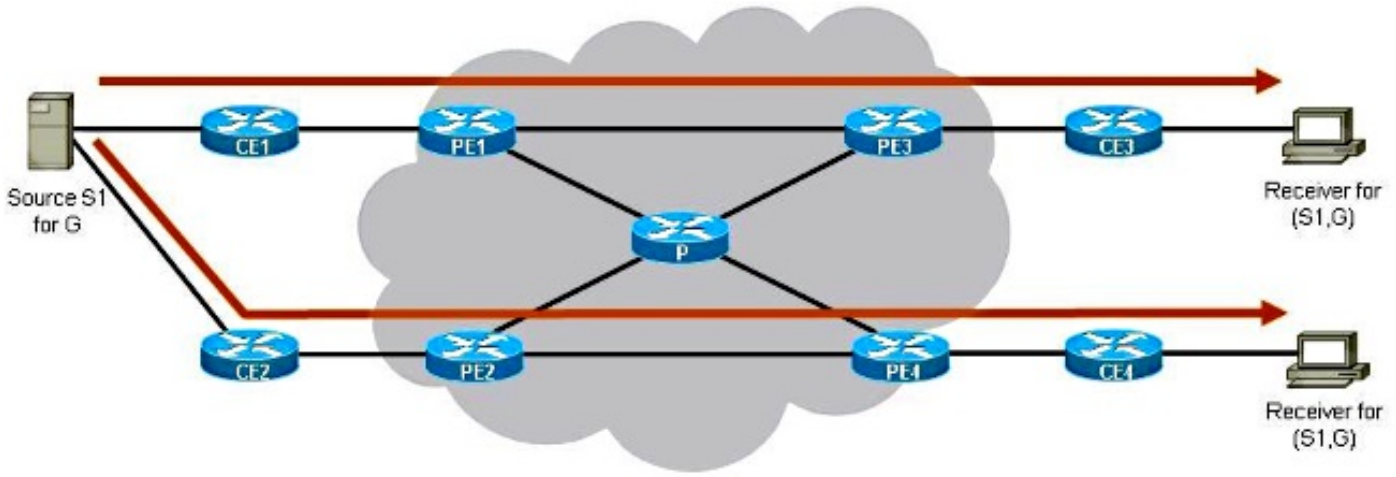

Figure 1

Hay dos routers PE de ingreso: PE1 y PE2. Hay dos routers PE de salida: PE3 y PE4. Cada router PE de salida tiene un router PE de entrada diferente como vecino RPF. PE3 tiene PE1 como vecino RPF. PE4 tiene PE2 como vecino RPF. Los routers PE de egreso eligen su router PE de entrada más cercano como su vecino RPF.

El Stream (S1,G) pasará de S1 al Receptor 1 por la trayectoria superior y de S1 al Receptor 2 por la trayectoria inferior. No hay intersección de los dos flujos en las dos trayectorias (cada trayectoria en el núcleo MPLS es un MDT dividido diferente).

Si el MDT fuera un MDT predeterminado - como en los perfiles MDT predeterminados - esto no funcionaría porque las dos secuencias multicast estarían en el mismo MDT predeterminado y el mecanismo de aserción se ejecutaría. Si el MDT es un MDT de datos en los perfiles MDT predeterminados, todos los routers PE de entrada se unen al MDT de datos de los otros routers PE de entrada y, como tal, ven el tráfico multicast entre sí y el mecanismo de aserción se ejecuta de nuevo. Si el protocolo de superposición es BGP, hay una selección de Salto de multidifusión ascendente (UMH) y sólo se selecciona un router PE de entrada como reenviador, pero esto es por MDT.

Anycast Source es una de las grandes ventajas de ejecutar MDT particionado.

Problema

La verificación RPF regular confirma que los paquetes llegan al router desde la interfaz RPF correcta. No hay ninguna verificación para confirmar que los paquetes se reciben del vecino RPF correcto en esa interfaz.

Consulte la Figura 2. Muestra un problema donde el tráfico duplicado se reenvía de forma persistente en un escenario con MDT particionado. Muestra que la verificación RPF regular en el caso de MDT particionado no es suficiente para evitar el tráfico duplicado.

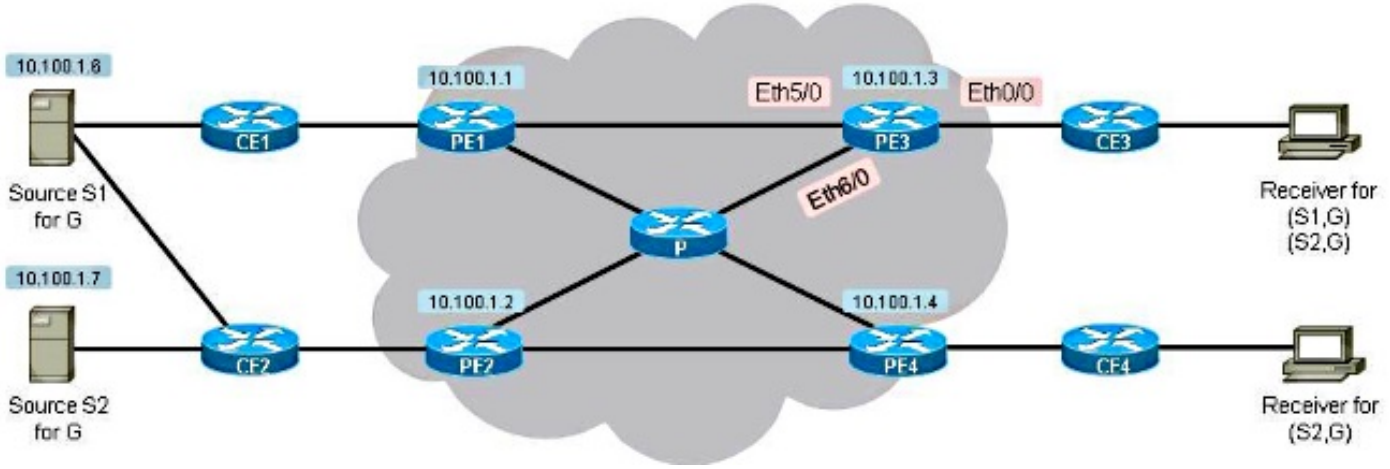

Figure 2

Hay dos receptores. El primer receptor se configura para recibir tráfico para (S1,G) y (S2,G). El segundo receptor se configura para recibir tráfico solamente para (S2,G). Hay MDT particionado y BGP es el protocolo de señalización superpuesta. Tenga en cuenta que el Origen S1 se puede alcanzar a través de PE1 y PE2. El protocolo de árbol de núcleo es el protocolo de distribución de etiquetas multipunto (mLDP).

Cada router PE anuncia una ruta BGP IPv4 mVPN de Tipo 1, que indica que es candidato para ser la raíz de un MDT particionado.

```
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 encuentra PE1 como el vecino RPF para S1 después de una búsqueda de la ruta unicast para 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 selecciona PE1 como vecino RPF para (S1,G) y se une al MDT particionado con PE1 como root. PE3 selecciona PE2 como vecino RPF para (S2,G) y se une al MDT particionado con PE2 como root.

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

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

PE4 selecciona PE2 como vecino RPF para (S1,G) y se une al MDT particionado con PE1 como root.

```
PE4#show bgp vpnv4 unicast vrf one 10.100.1.6/32
BGP routing table entry for 1:4:10.100.1.6/32, version 138
Paths: (2 available, best #1, table one)
Advertised to update-groups:
    2
Refresh Epoch 2
65001, imported path from 1:2:10.100.1.6/32 (global)
10.100.1.2 (metric 11) (via default) from 10.100.1.5 (10.100.1.5)
    Origin incomplete, metric 0, localpref 100, valid, internal, best
     Extended Community: RT:1:1 MVPN AS:1:0.0.0.0 MVPN VRF:10.100.1.2:1
```

```
    Originator: 10.100.1.2, Cluster list: 10.100.1.5
    mpls labels in/out nolabel/20
    rx pathid: 0, tx pathid: 0x0
Refresh Epoch 2
65001, imported path from 1:1:10.100.1.6/32 (global)
  10.100.1.1 (metric 21) (via default) from 10.100.1.5 (10.100.1.5)
    Origin incomplete, metric 0, localpref 100, valid, internal
    Extended Community: RT:1:1 MVPN AS:1:0.0.0.0 MVPN VRF:10.100.1.1:1
    Originator: 10.100.1.1, Cluster list: 10.100.1.5
    mpls labels in/out nolabel/29
    rx pathid: 0, tx pathid: 0
```

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

Observe que la interfaz RPF es Lspvif0 tanto para S1 (10.100.1.6) como para S2 (10.100.1.7).

PE3 se une al MDT particionado de PE2 para (S2,G) y PE4 se une al MDT particionado de PE2 para (S1,G). PE1 se une al MDT particionado de PE1 para (S1,G). Puede ver esto por las rutas mVPN IPv4 BGP de tipo 7 recibidas en PE1 y 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 ?
```

Las entradas multicast en PE3 y 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
```

Esto muestra que PE3 se une al árbol de punto a multipunto (P2MP) con raíces en PE1 y también al árbol con raíces en PE2:

```
PE3#show mls 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
```

Esto muestra que PE4 se une al árbol P2MP con raíces en 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
```

Secuencia S1 y S2 para el grupo 232.1.1.1 con 10 pps. Puede ver las secuencias en PE3 y PE4. Sin embargo, en PE3, puede ver la velocidad para (S1,G) como 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
```

Hay una secuencia duplicada. Esta duplicación es el resultado de la presencia de la secuencia (S1,G) en el MDT particionado de PE1 y en el MDT particionado de PE2. Este segundo MDT particionado, de PE2, se unió a PE3 para obtener el flujo (S2,G). Pero, como PE4 se unió al MDT particionado de PE2 para obtener (S1,G), (S1,G) también está presente en el MDT particionado de PE2. Por lo tanto, PE3 recibe la secuencia (S1,G) de los dos MDT particionados a los que se unió.

PE3 no puede discriminar entre los paquetes para (S1,G) que recibe de PE1 y PE2. Ambos flujos se reciben en la interfaz RPF correcta: 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
```

Los paquetes podrían llegar a diferentes interfaces físicas entrantes en PE3 o en la misma interfaz. En cualquier caso, los paquetes de las diferentes secuencias para (S1,G) llegan con una etiqueta MPLS diferente en 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
```

Solución

La solución es tener un RPF más estricto. Con RPF estricto, el router verifica desde qué vecino se reciben los paquetes en la interfaz RPF. Sin RPF estricto, la única verificación es determinar si la interfaz entrante es la interfaz RPF, pero no si los paquetes se reciben del vecino RPF correcto en esa interfaz.

Notas para Cisco IOS

A continuación se muestran algunas notas importantes sobre RPF con Cisco IOS.

- Cuando cambia al modo RPF estricto o lo cambia de uno a otro, configúrelo antes de

configurar el MDT particionado o despeje el BGP. Si sólo configura el comando RPF estricto, no creará otra interfaz Lspvif inmediatamente.

- El RPF estricto no está habilitado de forma predeterminada en Cisco IOS.
- No se soporta tener el comando **strict-rpf** con perfiles MDT predeterminados.

Configuración

Puede configurar un RPF estricto en PE3 para el routing y reenvío virtuales (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
!
```

La información de RPF ha cambiado:

```
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 creó una interfaz Lspvif por PE de entrada. La interfaz Lspvif se crea por PE de entrada, por familia de direcciones (AF) y por VRF. El RPF para 10.100.1.6 ahora apunta a la interfaz Lspvif1 y el RPF para 10.100.1.7 ahora apunta a la interfaz 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
```

Ahora, la verificación RPF de los paquetes (S1,G) de PE1 se verifica en la interfaz RPF Lspvif1. Estos paquetes entran con la etiqueta MPLS 29. La verificación de RPF para los paquetes (S2,G) de PE2 se verifica con la interfaz RPF Lspvif2. Estos paquetes entran con la etiqueta MPLS 30. Los flujos llegan a PE3 a través de diferentes interfaces entrantes, pero esto también podría ser la misma interfaz. Sin embargo, debido al hecho de que mLDP nunca utiliza PenUltima-Hop-Popping (PHP), siempre hay una etiqueta MPLS regular sobre los paquetes multicast. Los paquetes (S1,G) que llegan de PE1 y de PE2 están en dos MDTs Particionados diferentes y por lo tanto tienen una etiqueta MPLS diferente. Por lo tanto, PE3 puede discriminar entre la secuencia (S1,G) que viene de PE1 y la secuencia (S1,G) que viene de PE2. De esta manera, los paquetes se pueden mantener separados por PE3 y un RPF se puede realizar contra diferentes routers PE de entrada.

La base de datos mLDP en PE3 ahora muestra las diferentes interfaces Lspvif por PE de entrada.

```
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 estricto o RPF por PE de entrada funciona debido al hecho de que las secuencias multicast ingresan al PE de entrada con una etiqueta MPLS diferente por PE de ingreso:

```
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 prueba de que RPF estricto funciona es que ya no hay una secuencia duplicada (S1,G) reenviada en PE3. La secuencia duplicada aún llega en PE3, pero se descarta debido a la falla de RPF. El contador de fallas de RPF es de 676255 y aumenta constantemente a una velocidad 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
```

La velocidad de salida en PE3 es ahora de 20 pps, que es de 10 pps para cada flujo (S1,G) y (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
```

Conclusión

Se debe utilizar la Verificación RPF estricta para los modelos de implementación mVPN que utilizan MDT particionado.

Puede parecer que las cosas funcionan, incluso si no configura la verificación RPF estricta para los modelos de implementación mVPN con MDT particionado: los flujos multicast se entregan a los receptores. Sin embargo, existe la posibilidad de que haya tráfico multicast duplicado cuando los orígenes están conectados a varios routers PE de entrada. Esto lleva a un desperdicio de ancho de banda en la red y puede afectar negativamente a la aplicación multicast en los receptores. Por lo tanto, es necesario configurar la verificación RPF estricta para los modelos de implementación mVPN que utilizan MDT particionado.