Strict RPF Check für mVPN

Inhalt

Einführung

In diesem Dokument wird die Funktion Strict Reverse Path Forwarding (RPF) für Multicast over VPN (mVPN) beschrieben. In diesem Dokument wird ein Beispiel und die Implementierung in Cisco IOS® verwendet, um das Verhalten zu veranschaulichen.

Hintergrundinformationen

RPF impliziert, dass die eingehende Schnittstelle an die Quelle geprüft wird. Obwohl die Schnittstelle überprüft wird, um festzustellen, ob sie der Quelle entspricht, wird nicht überprüft, ob sie der richtige RPF-Nachbar auf dieser Schnittstelle ist. Auf einer Multiaccess-Schnittstelle kann es mehrere Nachbarn geben, zu denen Sie RPF erstellen können. Das Ergebnis könnte sein, dass der Router doppelt denselben Multicast-Stream auf dieser Schnittstelle empfängt und beide weiterleitet.

In Netzwerken, in denen Protocol Independent Multicast (PIM) auf der Multiaccess-Schnittstelle ausgeführt wird, stellt dies kein Problem dar, da der doppelte Multicast-Stream die Ausführung des Assertionsmechanismus bewirkt und ein Multicast-Stream nicht mehr empfangen wird. In einigen Fällen wird PIM nicht auf dem Multicast Distribution Tree (MDT) ausgeführt, einer Schnittstelle mit mehreren Zugriffen. In diesen Fällen ist Border Gateway Protocol (BGP) das Overlay-Signalisierungsprotokoll.

In Profilen mit partitioniertem MDT kann es selbst dann unmöglich sein, Asserts zu haben, wenn PIM als Overlay-Protokoll ausgeführt wird. Der Grund dafür ist, dass ein Eingangs-Provider-Edge (PE) in Szenarien, in denen es zwei oder mehr Eingangs-PE-Router gibt, nicht mit dem partitionierten MDT eines anderen Eingangs-PE verbunden ist. Jeder Eingangs-PE-Router kann den Multicast-Stream an seinen Partitionierten MDT weiterleiten, ohne dass der andere Eingangs-PE-Router den Multicast-Datenverkehr sieht. Die Tatsache, dass zwei verschiedene Egress-PE-Router jeweils einem MDT zu einem anderen Ingress-PE-Router für denselben Multicast-Stream gehören, ist ein gültiges Szenario: wird als Anycast Source bezeichnet. Dadurch können verschiedene Empfänger demselben Multicast-Stream beitreten, jedoch über einen anderen Pfad im MPLS-Core (Multiprotocol Label Switching). Ein Beispiel für Anycast Source finden Sie in

Abbildung 1.

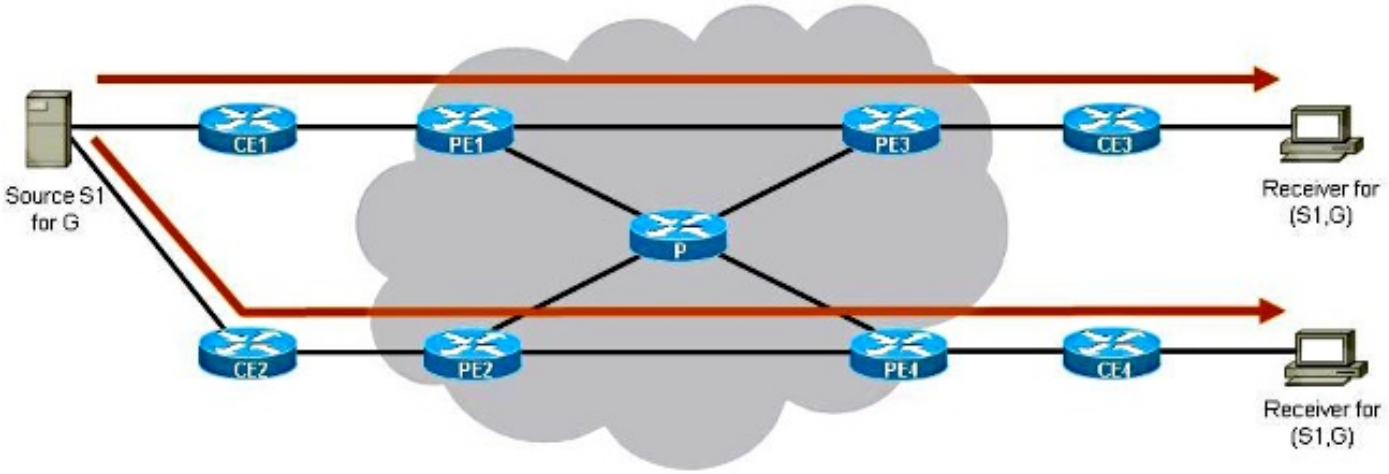

Abbildung 1

Es gibt zwei Eingangs-PE-Router: PE1 und PE2. Es gibt zwei Egress-PE-Router: PE3 und PE4. Jeder Egress-PE-Router hat einen anderen Ingress-PE-Router als seinen RPF-Nachbarn. PE3 hat PE1 als RPF-Nachbarn. PE4 hat PE2 als RPF-Nachbarn. Die Egress-PE-Router wählen ihren nächsten Eingangs-PE-Router als RPF-Nachbarn aus.

Der Stream (S1,G) wechselt von S1 zum Empfänger 1 über den oberen Pfad und von S1 zum Empfänger 2 über den unteren Pfad. Es gibt keine Überschneidung der beiden Streams über die beiden Pfade (jeder Pfad im MPLS-Core ist ein anderer partitionierter MDT).

Wenn der MDT ein Standard-MDT ist (wie in den Standard-MDT-Profilen), würde dies nicht funktionieren, da sich die beiden Multicast-Streams im gleichen Standard-MDT befinden und der Assert-Mechanismus ausgeführt wird. Wenn der MDT ein Daten-MDT in den Standard-MDT-Profilen ist, werden alle Eingangs-PE-Router dem Daten-MDT von den anderen Eingangs-PE-Routern hinzugefügt. Daher wird der Multicast-Datenverkehr zwischen den Routern überwacht, und der Assert-Mechanismus wird erneut ausgeführt. Wenn das Overlay-Protokoll BGP lautet, wird Upstream Multicast Hop (UMH) ausgewählt, und es wird nur ein Ingress-PE-Router als Forwarder ausgewählt, jedoch pro MDT.

Anycast Source ist einer der großen Vorteile der Ausführung von Partitioniertem MDT.

Problem

Die reguläre RPF-Prüfung bestätigt, dass die Pakete über die richtige RPF-Schnittstelle am Router ankommen. Es wird nicht überprüft, ob die Pakete vom richtigen RPF-Nachbarn auf dieser Schnittstelle empfangen werden.

Siehe Abbildung 2. Es zeigt ein Problem, bei dem doppelter Datenverkehr in einem Szenario mit partitioniertem MDT dauerhaft weitergeleitet wird. Es zeigt, dass die reguläre RPF-Prüfung bei einem partitionierten MDT nicht ausreicht, um doppelten Datenverkehr zu vermeiden.

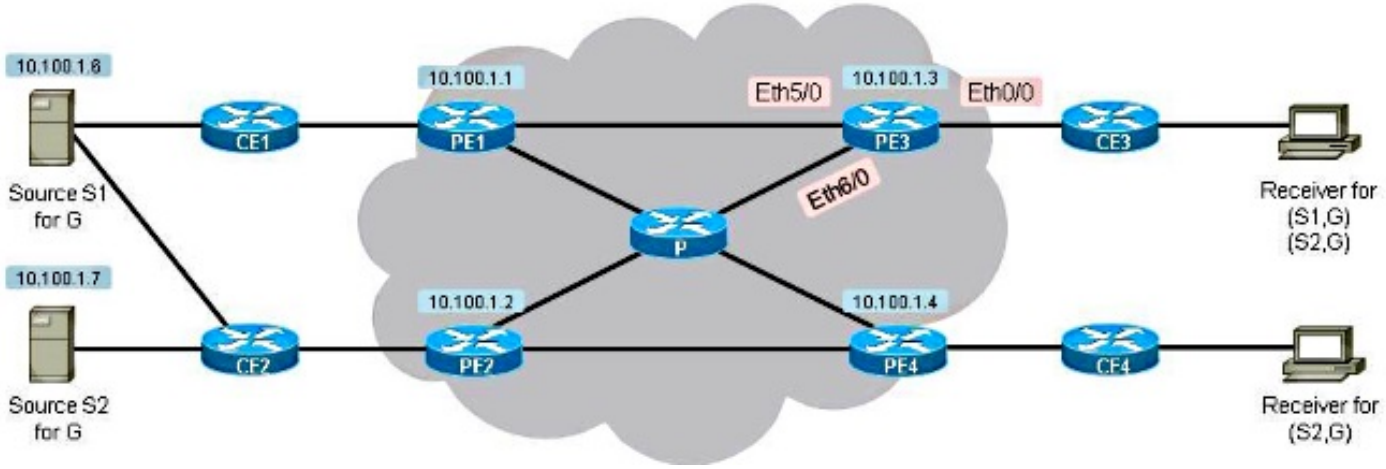

Abbildung 2

Es gibt zwei Empfänger. Der erste Empfänger ist so eingerichtet, dass er Datenverkehr für (S1,G) und (S2,G) empfängt. Der zweite Empfänger ist so eingerichtet, dass er nur Datenverkehr für (S2,G) empfängt. Es gibt einen partitionierten MDT, und BGP ist das Overlay-Signalisierungsprotokoll. Beachten Sie, dass die Quelle S1 sowohl über PE1 als auch über PE2 erreichbar ist. Das Core-Tree-Protokoll ist Multipoint Label Distribution Protocol (mLDP).

Jeder PE-Router kündigt eine BGP-IPv4-mVPN-Route vom Typ 1 an, was anzeigt, dass er der Root eines partitionierten MDT sein kann.

```
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 findet PE1 nach einer Suche nach der Unicast-Route für S1 als RPF-Nachbar für 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 wählt PE1 als RPF-Nachbar für (S1,G) aus und bindet den partitionierten MDT mit PE1 als Root an. PE3 wählt PE2 als RPF-Nachbar für (S2,G) und fügt sich dem partitionierten MDT als Root PE2 hinzu.

```
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 wählt PE2 als RPF-Nachbar für (S1,G) aus und bindet den partitionierten MDT mit PE1 als Root an.

```
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
```

Beachten Sie, dass die RPF-Schnittstelle LSPVIF0 für S1 (10.100.1.6) und S2 (10.100.1.7) ist.

PE3 fügt sich dem partitionierten MDT von PE2 für (S2,G) bei, und PE4 bindet den partitionierten MDT von PE2 für (S1,G) ein. PE1 fügt sich in den partitionierten MDT von PE1 für (S1,G) ein. Sie sehen dies anhand der BGP-IPv4-mVPN-Routen vom Typ 7, die auf PE1 und PE2 empfangen wurden.

```
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 ?
```

Die Multicast-Einträge auf PE3 und 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
```

Dies zeigt, dass PE3 mit dem Point-to-Multipoint (P2MP)-Tree verbunden ist, dessen Ursprung PE1 ist, sowie mit dem Tree, dessen Ursprung PE2 ist:

```
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
```

Dies zeigt, dass PE4 mit dem P2MP-Tree verbunden ist, dessen Ursprung PE2 ist:

```
PE4#show mpls mldp database
* Indicates MLDP recursive forwarding is enabled

LSM ID : 3   Type: P2MP   Uptime : 21:17:06
 FEC Root           : 10.100.1.2
 Opaque decoded     : [gid 65536 (0x00010000)]

Opaque value       : 01 0004 00010000
Upstream client(s) :
  10.100.1.2:0   [Active]
    Expires       : Never         Path Set ID : 3
    Out Label (U) : None         Interface   : Ethernet5/0*
    Local Label (D): 29           Next Hop     : 10.1.6.2
Replication client(s):
  MDT (VRF one)
    Uptime         : 21:17:06     Path Set ID : None
    Interface     : Lspvif0
```

S1- und S2-Stream für die Gruppe 232.1.1.1 mit 10 pps. Sie können die Streams an PE3 und PE4 sehen. Bei PE3 können Sie die Rate für (S1,G) jedoch als 20 pps anzeigen.

```
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
```

Es gibt einen doppelten Stream. Diese Duplizierung ist das Ergebnis des Streams (S1,G) auf dem partitionierten MDT von PE1 und dem partitionierten MDT von PE2. Dieser zweite partitionierte MDT aus PE2 wurde von PE3 hinzugefügt, um den Stream (S2,G) abzurufen. Da jedoch PE4 dem partitionierten MDT von PE2 beigetreten ist, um (S1,G) zu erhalten, ist (S1,G) auch auf dem partitionierten MDT von PE2 vorhanden. Daher empfängt PE3 den Stream (S1,G) von beiden partitionierten MDTs, zu denen er gehört.

PE3 kann nicht zwischen den Paketen für (S1,G) unterscheiden, die er von PE1 und PE2 empfängt. Beide Streams werden auf der richtigen RPF-Schnittstelle empfangen: 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
```

Die Pakete können auf verschiedenen eingehenden physischen Schnittstellen auf PE3 oder auf derselben Schnittstelle eingehen. Auf jeden Fall werden die Pakete aus den verschiedenen Streams für (S1,G) mit einem anderen MPLS-Label am PE3 empfangen:

```
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
```

Lösung

Die Lösung ist eine strengere RPF. Bei striktem RPF überprüft der Router, von welchem Nachbarn die Pakete auf der RPF-Schnittstelle empfangen werden. Ohne die strikte RPF-Methode kann nur geprüft werden, ob die eingehende Schnittstelle die RPF-Schnittstelle ist, nicht jedoch, ob die Pakete vom richtigen RPF-Nachbarn auf dieser Schnittstelle empfangen werden.

Hinweise zu Cisco IOS

Hier einige wichtige Hinweise zu RPF mit Cisco IOS.

- Wenn Sie in den/vom strikten RPF-Modus wechseln, konfigurieren Sie ihn entweder vor der Konfiguration des Partitionierten MDT oder des BGP-Leerstellen. Wenn Sie nur den Befehl

strict RPF konfigurieren, wird nicht sofort eine weitere Lspvif-Schnittstelle erstellt.

- Strict RPF ist in Cisco IOS standardmäßig nicht aktiviert.
- Es wird nicht unterstützt, den Befehl **strict-rpf** mit Standard-MDT-Profilen zu verwenden.

Konfiguration

Sie können auf PE3 für VRF (Virtual Routing and Forwarding) ein striktes RPF konfigurieren.

```
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
!
```

Die RPF-Informationen haben sich geändert:

```
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 hat pro Eingangs-PE eine LSPVIF-Schnittstelle erstellt. Die LSPVIF-Schnittstelle wird pro Eingangs-PE, pro Adressfamilie (AF) und pro VRF erstellt. Der RPF für 10.100.1.6 zeigt nun auf die Schnittstelle Lspvif1 und der RPF für 10.100.1.7 auf die Schnittstelle 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
```

Jetzt wird die RPF-Prüfung auf Pakete (S1,G) von PE1 mit der RPF-Schnittstelle Lspvif1 abgeglichen. Diese Pakete werden mit dem MPLS-Label 29 geliefert. Die RPF-Prüfung auf Pakete

(S2,G) von PE2 wird mit der RPF-Schnittstelle Lspvif2 abgeglichen. Diese Pakete werden mit dem MPLS-Label 30 geliefert. Die Streams gelangen über verschiedene eingehende Schnittstellen auf PE3, aber auch über dieselbe Schnittstelle. Da mLDP jedoch niemals Penultimate-Hop-Popping (PHP) verwendet, wird auf den Multicast-Paketen immer ein reguläres MPLS-Label angezeigt. Die (S1,G)-Pakete, die von PE1 und PE2 eintreffen, befinden sich in zwei verschiedenen partitionierten MDTs und verfügen daher über ein anderes MPLS-Label. Daher kann PE3 zwischen dem (S1,G)-Stream von PE1 und dem (S1,G)-Stream von PE2 unterscheiden. Auf diese Weise können die Pakete vom PE3 getrennt gehalten werden, und ein RPF kann für verschiedene Eingangs-PE-Router ausgeführt werden.

Die mLDP-Datenbank auf PE3 zeigt jetzt die verschiedenen LSPVIF-Schnittstellen pro Eingangs-PE an.

```
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
```

Strict RPF oder RPF pro Eingangs-PE ist darauf zurückzuführen, dass die Multicast-Streams mit einem anderen MPLS-Label pro Eingangs-PE in den Eingangs-PE eingehen:

```
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
```

Der Beweis für die Funktion eines strikten RPF besteht darin, dass auf PE3 kein doppelter Stream (S1,G) weitergeleitet wird. Der doppelte Stream kommt weiterhin auf PE3 an, wird aber aufgrund des RPF-Ausfalls verworfen. Der RPF-Fehlerzähler liegt bei 676255 und erhöht sich konstant um 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
```

Die Ausgangsrate bei PE3 beträgt jetzt 20 pps, d. h. 10 pps für jeden Stream (S1,G) und (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
```

Schlussfolgerung

Für die mVPN-Bereitstellungsmodelle, die einen partitionierten MDT verwenden, muss eine strikte RPF-Prüfung verwendet werden.

Es scheint, dass die Dinge funktionieren, selbst wenn Sie die strenge RPF-Prüfung für die mVPN-Bereitstellungsmodelle mit partitioniertem MDT nicht konfigurieren: die Multicast-Streams werden an die Empfänger übertragen. Es besteht jedoch die Möglichkeit, dass Multicast-Datenverkehr dupliziert wird, wenn Quellen mit mehreren Eingangs-PE-Routern verbunden sind. Dies führt zu einer Verschwendung von Bandbreite im Netzwerk und kann die Multicast-Anwendung auf den Empfängern negativ beeinflussen. Daher muss eine strenge RPF-Prüfung für die mVPN-Bereitstellungsmodelle konfiguriert werden, die einen partitionierten MDT verwenden.