Dual-Homed Source- und Daten-MDT in mVPN

Inhalt

Einführung

In diesem Dokument wird das mVPN (Multicast Virtual Provider Network) mit Dual-Homed Source und Daten-MDT (Multicast Distribution Tree) beschrieben. Ein Beispiel in Cisco IOS® dient zur Veranschaulichung des Verhaltens.

Das Problem

Wenn eine Quelle in der mVPN-Welt über Dual-Homed mit zwei Ingress Provider Edge (PE)-Routern verbunden ist, können die beiden Ingress-PE-Router den Datenverkehr für einen (S,G) in die Multiprotocol Label Switching (MPLS)-Cloud weiterleiten. Dies ist beispielsweise möglich, wenn es zwei Egress-PE-Router und jeden Reverse Path Forwarding (RPF) zu einem anderen Ingress-PE-Router gibt. Wenn beide Eingangs-PE-Router an den Standard-MDT weitergeleitet werden, tritt der Assert-Mechanismus ein, und ein Ingress-PE gewinnt den Assert-Mechanismus, der andere verliert, sodass der eine und nur ein Ingress-PE den Kunden (C-) (S,G) weiter an den MDT weiterleitet. Wenn der Assert-Mechanismus jedoch nicht auf dem Standard-MDT gestartet wurde, können beide Eingangs-PE-Router beginnen, den C-(S,G)-Multicast-Datenverkehr auf einen von ihnen initiierten Daten-MDT zu übertragen. Da der Datenverkehr nicht mehr dem Standard-MDT, sondern den Daten-MDTs entspricht, empfangen beide Eingangs-PE-Router den C-(S,G)-Datenverkehr auf der MDT-/Tunnel-Schnittstelle nicht mehr voneinander. Dies kann zu einem permanenten Duplikat des Datenverkehrs im Downstream führen. Dieses Dokument erläutert die Lösung dieses Problems.

Assert-Mechanismus auf dem Standard-MDT

Die Informationen in diesem Abschnitt gelten für den Standard-MDT, unabhängig vom Core-Tree-Protokoll. Das gewählte Core-Tree-Protokoll ist Protocol Independent Multicast (PIM).

Für die Beispiele wird Cisco IOS verwendet, aber alles, was erwähnt wird, gilt gleichermaßen für Cisco IOS-XR. Alle verwendeten Multicast-Gruppen sind Source Specific Multicast (SSM)-Gruppen.

Sehen Sie sich Abbildung 1 an. Dual-Homed-Source-1 Es gibt zwei Eingangs-PE-Router (PE1 und PE2) und zwei Egress-PE-Router (PE3 und PE4). Die Quelle ist CE1 mit der IP-Adresse 10.100.1.6. CE1 ist Dual-Homed zu PE1 und PE2.

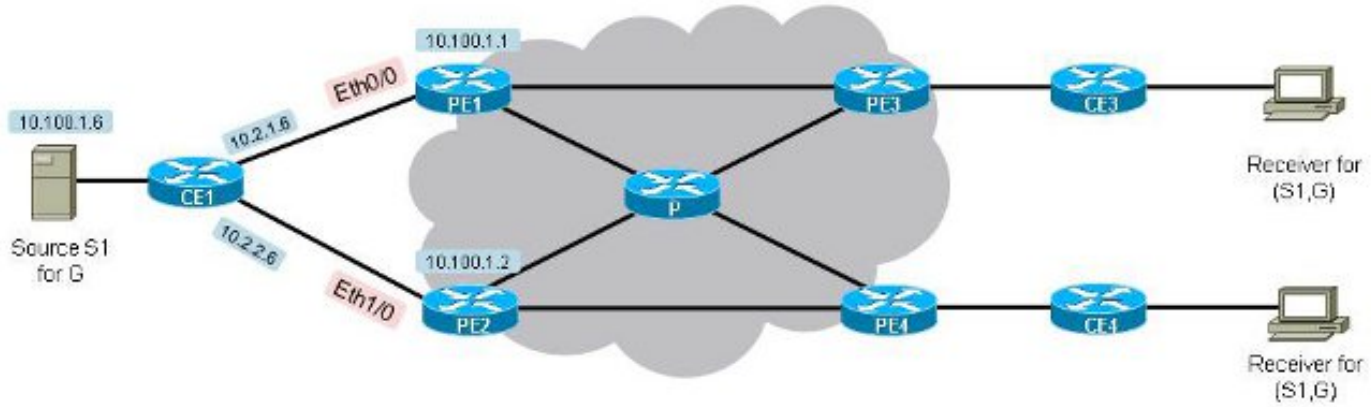

Abbildung 1: Dual-Homed-Source-1

Die Konfiguration aller PE-Router (der Route Distinguisher (RD) kann auf den PE-Routern unterschiedlich sein) ist wie folgt:

```
vrf definition one
 rd 1:1
 !
 address-family ipv4
 mdt default 232.10.10.10
 route-target export 1:1
 route-target import 1:1
 exit-address-family
!
```

Damit beide Eingangs-PE-Router den Multicast-Stream (10.100.1.6.232.1.1.1) an den Standard-MDT weiterleiten können, müssen sie beide eine Join-Nachricht von einem Egress-PE erhalten. Sehen Sie sich die Topologie in Abbildung 1 an. Dual-Homed-Source-1 Wenn alle Kosten der Edge-Verbindungen identisch sind und alle Kosten für die Core-Verbindungen gleich sind, wird PE3 standardmäßig für PE1 und PE4 für PE2 (10.100.1.6.232.1.1.1) für RPF zu PE2 festgelegt. Beide RPFs sind mit ihrem nächsten Eingangs-PE verbunden. Diese Ausgabe bestätigt Folgendes:

```
PE3#show ip rpf vrf one 10.100.1.6
RPF information for ? (10.100.1.6)
 RPF interface: Tunnel0
 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
 BGP originator: 10.100.1.1
 RPF topology: ipv4 multicast base, originated from ipv4 unicast base
```

PE3 verfügt über RPF zu PE1.

```
PE4#show ip rpf vrf one 10.100.1.6
RPF information for ? (10.100.1.6)
 RPF interface: Tunnel0
 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
 BGP originator: 10.100.1.2
 RPF topology: ipv4 multicast base, originated from ipv4 unicast base
```

PE4 verfügt über RPF zu PE2. PE3 wählt PE1 als RPF-Nachbarn aus, weil die Unicast-Route zu 10.100.1.6/32 in Virtual Routing/Forwarding (VRF) die beste Route über PE1 ist. PE3 empfängt die Route 10.100.1.6/32 von PE1 und PE2. Alle Kriterien im Border Gateway Protocol (BGP) Best Path Calcation Algorithm sind identisch, mit Ausnahme der Kosten für die BGP Next-Hop-Adresse.

```
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 333
Paths: (2 available, best #1, table one)
 Advertised to update-groups:
    21
 Refresh Epoch 1
 Local, 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 11, localpref 100, valid, internal,best
     Extended Community: RT:1:1 OSPF DOMAIN ID:0x0005:0x000000640200
       OSPF RT:0.0.0.0:2:0 OSPF ROUTER ID:10.2.4.1:0
     Originator: 10.100.1.1, Cluster list: 10.100.1.5
     Connector Attribute: count=1
      type 1 len 12 value 1:1:10.100.1.1
     mpls labels in/out nolabel/32
     rx pathid: 0, tx pathid: 0x0
 Refresh Epoch 1
 Local, 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 11, localpref 100, valid, internal
     Extended Community: RT:1:1 OSPF DOMAIN ID:0x0005:0x000000640200
       OSPF RT:0.0.0.0:2:0 OSPF ROUTER ID:10.2.2.2:0
     Originator: 10.100.1.2, Cluster list: 10.100.1.5
     Connector Attribute: count=1
      type 1 len 12 value 1:2:10.100.1.2
     mpls labels in/out nolabel/29
     rx pathid: 0, tx pathid: 0

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 1050
Paths: (2 available, best #2, table one)
 Advertised to update-groups:
    2
 Refresh Epoch 1
 Local, 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 11, localpref 100, valid, internal
     Extended Community: RT:1:1 OSPF DOMAIN ID:0x0005:0x000000640200
       OSPF RT:0.0.0.0:2:0 OSPF ROUTER ID:10.2.4.1:0
     Originator: 10.100.1.1, Cluster list: 10.100.1.5
     Connector Attribute: count=1
      type 1 len 12 value 1:1:10.100.1.1
     mpls labels in/out nolabel/32
     rx pathid: 0, tx pathid: 0
 Refresh Epoch 1
```

```
 Local, 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 11, localpref 100, valid, internal, best
     Extended Community: RT:1:1 OSPF DOMAIN ID:0x0005:0x000000640200
       OSPF RT:0.0.0.0:2:0 OSPF ROUTER ID:10.2.2.2:0
     Originator: 10.100.1.2, Cluster list: 10.100.1.5
     Connector Attribute: count=1
      type 1 len 12 value 1:2:10.100.1.2
     mpls labels in/out nolabel/29
     rx pathid: 0, tx pathid: 0x0
```

Der beste von PE3 gewählte Pfad ist der von PE1 angegebene Pfad, da dieser die niedrigsten IGP-Kosten (11) im Vergleich zu IGP-Kosten (21) für PE2 aufweist. Für PE4 ist dies das Gegenteil. Die Topologie zeigt, dass von PE3 zu PE1 nur ein Hop, von PE3 zu PE2 zwei Hops vorhanden sind. Da alle Verbindungen dieselben IGP-Kosten haben, wählt PE3 den Pfad von PE1 als den besten.

Die Multicast Routing Information Base (MRIB) für (10.100.1.6,232.1.1.1) sieht auf PE1 und PE2 wie folgt aus, wenn noch kein Multicast-Verkehr vorhanden ist:

```
PE1#show ip mroute vrf one 232.1.1.1 10.100.1.6
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), 00:00:12/00:03:17, flags: sT
 Incoming interface: Ethernet0/0, RPF nbr 10.2.1.6
 Outgoing interface list:
   Tunnel0, Forward/Sparse, 00:00:12/00:03:17

PE2#show ip mroute vrf one 232.1.1.1 10.100.1.6
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), 00:00:47/00:02:55, flags: sT
 Incoming interface: Ethernet1/0, RPF nbr 10.2.2.6
 Outgoing interface list:
   Tunnel0, Forward/Sparse, 00:00:47/00:02:55
```

PE1 und PE2 erhielten jeweils eine PIM-Join-Nachricht für (10.100.1.6,232.1.1.1). Die Tunnel0-Schnittstelle befindet sich in der OIL (Outgoing Interface List) für den Multicast-Eintrag auf beiden Routern.

Der Multicast-Datenverkehr beginnt für (10.100.1.6,232.1.1.1) zu fließen. "Debug ip pim vrf one 232.1.1.1" und "debug ip mrouting vrf one 232.1.1.1" zeigen uns, dass das Eintreffen von Multicast-Datenverkehr auf Tunnel0 (im OIL) beider Ingress-PE-Router dazu führt, dass der Assert-Mechanismus ausgeführt wird.

PE1

```
PIM(1): Send v2 Assert on Tunnel0 for 232.1.1.1, source 10.100.1.6, metric [110/11]
PIM(1): Assert metric to source 10.100.1.6 is [110/11]
MRT(1): not RPF interface, source address 10.100.1.6, group address 232.1.1.1
PIM(1): Received v2 Assert on Tunnel0 from 10.100.1.2
PIM(1): Assert metric to source 10.100.1.6 is [110/11]
PIM(1): We lose, our metric [110/11]
PIM(1): Prune Tunnel0/232.10.10.10 from (10.100.1.6/32, 232.1.1.1)
MRT(1): Delete Tunnel0/232.10.10.10 from the olist of (10.100.1.6, 232.1.1.1)
MRT(1): Reset the PIM interest flag for (10.100.1.6, 232.1.1.1)
MRT(1): set min mtu for (10.100.1.6, 232.1.1.1) 1500->18010 - deleted
PIM(1): Received v2 Join/Prune on Tunnel0 from 10.100.1.3, not to us
PIM(1): Join-list: (10.100.1.6/32, 232.1.1.1), S-bit set
```

PE2

```
PIM(1): Received v2 Assert on Tunnel0 from 10.100.1.1
PIM(1): Assert metric to source 10.100.1.6 is [110/11]
PIM(1): We win, our metric [110/11]
PIM(1): (10.100.1.6/32, 232.1.1.1) oif Tunnel0 in Forward state
PIM(1): Send v2 Assert on Tunnel0 for 232.1.1.1, source 10.100.1.6, metric [110/11]
PIM(1): Assert metric to source 10.100.1.6 is [110/11]
PIM(1): Received v2 Join/Prune on Tunnel0 from 10.100.1.3, to us
PIM(1): Join-list: (10.100.1.6/32, 232.1.1.1), S-bit set
PIM(1): Update Tunnel0/10.100.1.3 to (10.100.1.6, 232.1.1.1), Forward state, by PIM SG Join
```

Wenn die Metrik und die Entfernung für beide Router zum Source 10.100.1.6 identisch sind, gibt es einen Timer-Breaker, um den Gewinner des Assert-Tests zu ermitteln. Der Zeitbegrenzer ist die höchste IP-Adresse des PIM-Nachbarn auf Tunnel0 (Standard-MDT). In diesem Fall ist dies PE2:

```
PE1#show ip pim vrf one neighbor
PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      P - Proxy Capable, S - State Refresh Capable, G - GenID Capable,
      L - DR Load-balancing Capable
Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.100.1.4        Tunnel0                  06:27:57/00:01:29 v2    1 / DR S P G
10.100.1.3        Tunnel0                  06:28:56/00:01:24 v2    1 / S P G
10.100.1.2        Tunnel0                  06:29:00/00:01:41 v2    1 / S P G

PE1#show ip pim vrf one interface
```

```
Address          Interface                Ver/   Nbr     Query  DR         DR
                                          Mode   Count   Intvl  Prior
10.2.1.1         Ethernet0/0              v2/S   0       30     1          10.2.1.1
10.2.4.1         Ethernet1/0              v2/S   0       30     1          10.2.4.1
10.100.1.1       Lspvif1                  v2/S   0       30     1          10.100.1.1
10.100.1.1       Tunnel0                  v2/S   3       30     1          10.100.1.4
```

PE1 hat Tunnel0 aus dem OIL des Multicast-Eintrags wegen der Asserts entfernt. Da die OIL leer wurde, wird der Multicast-Eintrag gelöscht.

```
PE1#show ip mroute vrf one 232.1.1.1 10.100.1.6
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), 00:17:24/00:00:01, flags: sPT
 Incoming interface: Ethernet0/0, RPF nbr 10.2.1.6
 Outgoing interface list: Null
```

Für PE2 ist das A-Flag auf der Schnittstelle Tunnel0 festgelegt, da es der Gewinner des Asserts ist.

```
PE2#show ip mroute vrf one 232.1.1.1 10.100.1.6
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), 00:17:20/00:02:54, flags: sT
 Incoming interface: Ethernet1/0, RPF nbr 10.2.2.6
 Outgoing interface list:
   Tunnel0, Forward/Sparse, 00:17:20/00:02:54, A
```

PE2 sendet regelmäßig eine Anfrage an Tunnel0 (Standard-MDT), kurz bevor der Assert-Timer abläuft. Als solcher bleibt PE2 der Gewinner.

```
PE2#
PIM(1): Send v2 Assert on Tunnel0 for 232.1.1.1, source 10.100.1.6, metric [110/11]
PIM(1): Assert metric to source 10.100.1.6 is [110/11]
```

Schlussfolgerung

Der Assert-Mechanismus funktioniert auch mit einer Tunnelschnittstelle im OIL. Die Asserten werden über den Standard-MDT ausgetauscht, wenn die Eingangs-PE-Router C-(S,G)-Multicast-Datenverkehr auf der zugehörigen Tunnelschnittstelle empfangen, die sich im OIL befindet.

Assert-Mechanismus mit Daten-MDTs

In den meisten Fällen, in denen Daten-MDTs konfiguriert werden, wird der Assertionsmechanismus weiterhin auf dem Standard-MDT ausgeführt, da der C-(S,G)-Datenverkehr erst nach drei Sekunden vom Standard-MDT auf die Daten-MDTs umgeschaltet wird. Das Gleiche geschieht dann wie oben beschrieben. Beachten Sie, dass **pro Multicast-fähiges VRF nur eine Tunnelschnittstelle vorhanden ist**: Der Standard-MDT und alle Daten-MDTs verwenden nur eine Tunnelschnittstelle. Diese Tunnelschnittstelle wird in der OIL der Eingangs-PE-Router oder als RPF-Schnittstelle der Egress-PE-Router verwendet.

In einigen Fällen ist es möglich, dass der Assert-Mechanismus nicht ausgelöst wird, bevor die Daten-MDTs signalisiert werden. Dann ist es möglich, dass der C-(S,G)-Multicast-Datenverkehr auf einem Daten-MDT auf den Eingangs-PE-Routern PE1 und PE2 weitergeleitet wird. In solchen Fällen kann dies zu einem permanenten Duplikat des C-(S,G)-Multicast-Datenverkehrs im gesamten MPLS-Core-Netzwerk führen. Um dies zu vermeiden, wurde diese Lösung implementiert: Wenn ein Eingangs-PE-Router einen anderen Eingangs-PE-Router erkennt, kündigt er einen Daten-MDT an, für den der PE-Router auch ein Eingangs-PE-Router ist, er schließt sich diesem Daten-MDT an. Grundsätzlich werden dem Daten-MDT nur Egress-PE-Router (die über einen Downstream-Empfänger verfügen) hinzugefügt. Da Eingangs-PE-Router dem von anderen Eingangs-PE-Routern angekündigten Daten-MDT beitreten, führt er zum Eingangs-PE-Router, der Multicast-Datenverkehr von der Tunnelschnittstelle empfängt, die in der OIL vorhanden ist. Dies löst den Assertionsmechanismus aus und führt zu einem der Eingangs-PE-Router, der die Weiterleitung des C-(S,G)-Multicast-Datenverkehrs an seinen Daten-MDT (mit der Tunnelschnittstelle) beendet. der andere Ingress-PE (der Gewinner des Asserts) kann den C-(S,G)-Multicast-Datenverkehr weiterhin an seinen Daten-MDT weiterleiten.

Gehen Sie im nächsten Beispiel davon aus, dass die PE-Router für den Eingehend PE1 und PE2 den Multicast-Datenverkehr für C-(S,G) auf dem Standard-MDT nie voneinander gesehen haben. Der Datenverkehr befindet sich nur drei Sekunden lang auf dem Standard-MDT, und es ist nicht schwer zu verstehen, dass dieser Datenverkehr auftreten kann, wenn z. B. im Core-Netzwerk ein vorübergehender Datenverkehrsverlust auftritt.

Die Konfiguration für den Daten-MDT wird allen PE-Routern hinzugefügt. Die Konfiguration aller PE-Router (der RD kann auf den PE-Routern unterschiedlich sein) ist wie folgt:

```
vrf definition one
 rd 1:1
 !
 address-family ipv4
 mdt default 232.10.10.10
mdt data 232.11.11.0 0.0.0.0
 route-target export 1:1
 route-target import 1:1
```

```
 exit-address-family
!
```

Sobald PE1 und PE2 den Datenverkehr von der Quelle sehen, erstellen sie einen C-(S,G)-Eintrag. Beide Eingangs-PE-Router leiten den C-(S,G)-Multicast-Datenverkehr an den Standard-MDT weiter. Ausgangs-PE-Router PE3 und PE4 empfangen den Multicast-Datenverkehr und leiten ihn weiter. Aufgrund eines temporären Problems wird der Datenverkehr von PE1 für PE2 und umgekehrt für den Standard-MDT nicht angezeigt. Beide senden einen Daten-MDT-Join-Type Length Value (TLV) an den Standard-MDT.

Wenn kein C-(S,G)-Datenverkehr vorhanden ist, wird dieser Multicast-Status auf den Eingangs-PE-Routern angezeigt:

```
PE1#show ip mroute vrf one 232.1.1.1 10.100.1.6
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), 00:00:45/00:02:44, flags: sT
 Incoming interface: Ethernet0/0, RPF nbr 10.2.1.6
 Outgoing interface list:
   Tunnel0, Forward/Sparse, 00:00:45/00:02:42

PE2#show ip mroute vrf one 232.1.1.1 10.100.1.6
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), 00:02:18/00:03:28, flags: sT
 Incoming interface: Ethernet1/0, RPF nbr 10.2.2.6
 Outgoing interface list:
   Tunnel0, Forward/Sparse, 00:02:18/00:03:28
```

Das y-Flag ist noch nicht festgelegt. Beide Eingangs-PE-Router haben die Tunnel0-Schnittstelle im OIL. Dies liegt daran, dass PE3 über RPF für PE1 und PE4 über RPF für PE2 für C-(S,G)

verfügt.

Wenn der Multicast-Datenverkehr für C-(S,G) beginnt, leiten sowohl PE1 als auch PE2 den Datenverkehr weiter. Der Grenzwert für den Daten-MDT wird auf beiden Eingangs-PE-Routern überschritten. Beide Router senden einen Daten-MDT-Join-TLV, und nach drei Sekunden beginnt die Weiterleitung an den Daten-MDT. Beachten Sie, dass PE1 mit dem vom PE2 und PE2 bezogenen Daten-MDT verbunden ist, der vom PE1 bezogen wird.

```
PE1#show ip mroute vrf one 232.1.1.1 10.100.1.6
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), 00:01:26/00:03:02, flags: sTy
 Incoming interface: Ethernet0/0, RPF nbr 10.2.1.6
 Outgoing interface list:
   Tunnel0, Forward/Sparse, 00:01:26/00:03:02

PE2#show ip mroute vrf one 232.1.1.1 10.100.1.6
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), 00:00:41/00:02:48, flags: sTy
 Incoming interface: Ethernet1/0, RPF nbr 10.2.2.6
 Outgoing interface list:
   Tunnel0, Forward/Sparse, 00:00:41/00:02:48
```

Sowohl PE1 als auch PE empfangen Datenverkehr für C-(S,G) auf der Tunnel0-Schnittstelle (jetzt jedoch vom Daten-MDT, nicht vom Standard-MDT), und der Assertionsmechanismus wird aktiviert. Nur PE2 leitet weiterhin den C-(S,G)-Datenverkehr über seinen Daten-MDT weiter:

```
PE1#
PIM(1): Send v2 Assert on Tunnel0 for 232.1.1.1, source 10.100.1.6, metric [110/11]
```

```
PIM(1): Assert metric to source 10.100.1.6 is [110/11]
MRT(1): not RPF interface, source address 10.100.1.6, group address 232.1.1.1
PIM(1): Received v2 Assert on Tunnel0 from 10.100.1.2
PIM(1): Assert metric to source 10.100.1.6 is [110/11]
PIM(1): We lose, our metric [110/11]
PIM(1): Prune Tunnel0/232.11.11.0 from (10.100.1.6/32, 232.1.1.1)
MRT(1): Delete Tunnel0/232.11.11.0 from the olist of (10.100.1.6, 232.1.1.1)
MRT(1): Reset the PIM interest flag for (10.100.1.6, 232.1.1.1)
PIM(1): MDT Tunnel0 removed from (10.100.1.6,232.1.1.1)
MRT(1): Reset the y-flag for (10.100.1.6,232.1.1.1)
PIM(1): MDT next_hop change from: 232.11.11.0 to 232.10.10.10 for (10.100.1.6, 232.1.1.1)
Tunnel0
MRT(1): set min mtu for (10.100.1.6, 232.1.1.1) 1500->18010 - deleted
PIM(1): MDT threshold dropped for (10.100.1.6,232.1.1.1)
PIM(1): Receive MDT Packet (9889) from 10.100.1.2 (Tunnel0), length (ip: 44, udp: 24), ttl: 1
PIM(1): TLV type: 1 length: 16 MDT Packet length: 16

PE2#
PIM(1): Received v2 Assert on Tunnel0 from 10.100.1.1
PIM(1): Assert metric to source 10.100.1.6 is [110/11]
PIM(1): We win, our metric [110/11]
PIM(1): (10.100.1.6/32, 232.1.1.1) oif Tunnel0 in Forward state
PIM(1): Send v2 Assert on Tunnel0 for 232.1.1.1, source 10.100.1.6, metric [110/11]
PIM(1): Assert metric to source 10.100.1.6 is [110/11]
PE2#
PIM(1): Received v2 Join/Prune on Tunnel0 from 10.100.1.3, to us
PIM(1): Join-list: (10.100.1.6/32, 232.1.1.1), S-bit set
PIM(1): Update Tunnel0/10.100.1.3 to (10.100.1.6, 232.1.1.1), Forward state, by PIM SG Join
MRT(1): Update Tunnel0/232.10.10.10 in the olist of (10.100.1.6, 232.1.1.1), Forward state - MAC
built
MRT(1): Set the y-flag for (10.100.1.6,232.1.1.1)
PIM(1): MDT next_hop change from: 232.10.10.10 to 232.11.11.0 for (10.100.1.6, 232.1.1.1)
Tunnel0
```

PE1 verfügt nicht mehr über die Tunnelschnittstelle im OIL.

```
PE1#show ip mroute vrf one 232.1.1.1 10.100.1.6
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), 00:10:23/00:00:04, flags: sPT
 Incoming interface: Ethernet0/0, RPF nbr 10.2.1.6
 Outgoing interface list: Null
```

Für PE2 ist auf der Tunnel0-Schnittstelle das A-Flag festgelegt:

```
PE2#show ip mroute vrf one 232.1.1.1 10.100.1.6
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), 00:10:00/00:02:48, flags: sTy
 Incoming interface: Ethernet1/0, RPF nbr 10.2.2.6
 Outgoing interface list:
   Tunnel0, Forward/Sparse, 00:08:40/00:02:48, A
```

Schlussfolgerung

Der Assert-Mechanismus funktioniert auch, wenn Daten-MDTs verwendet werden. Die Asserten werden über den Standard-MDT ausgetauscht, wenn die Eingangs-PE-Router C-(S,G)-Multicast-Datenverkehr auf der zugehörigen Tunnelschnittstelle empfangen, die sich im OIL befindet.