

# Dual-Homed Source and Data MDT in mVPN

## Inhoud

[Inleiding](#)

[Het probleem](#)

[Mechanisme instellen op de standaard-MDT](#)

[Conclusie](#)

[Mechanisme voor controle met MDT's](#)

[Conclusie](#)

## Inleiding

Dit document beschrijft mVPN (Multicast Virtual Provider Network) met dubbele homed Source and Data MDT (Multicast Distribution Tree). Een voorbeeld in Cisco IOS<sup>®</sup> wordt gebruikt om het gedrag te illustreren.

## Het probleem

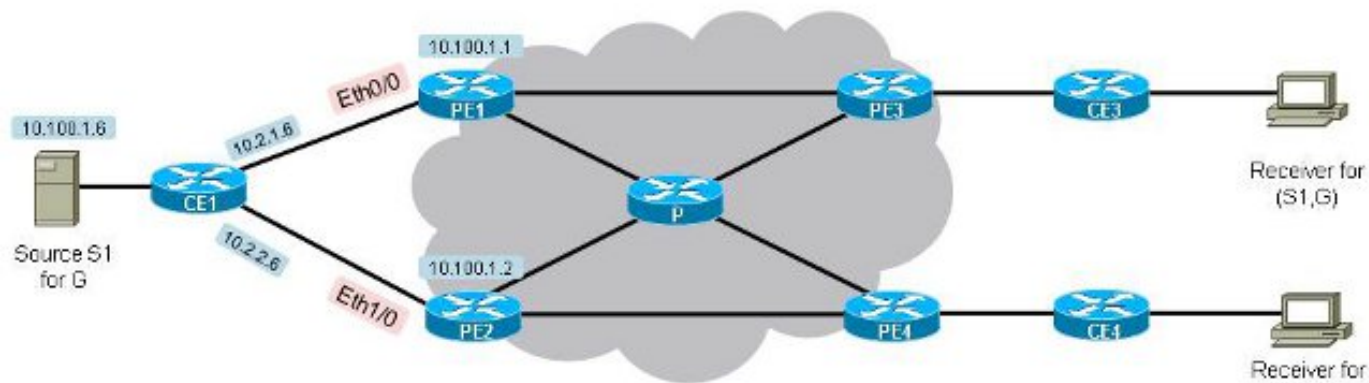
Als een bron in de mVPN-wereld twee-gecalibreerd is naar twee PE-routers (Ingress Provider Edge), is het voor de twee Ingoers mogelijk om beide IP-routers zowel voorwaarts verkeer voor één (S,G) naar de MPLS-cloud (Multiprotocol Label Switching) te verzenden. Dit is mogelijk als, bijvoorbeeld, er twee druk-PE routers en elk omgekeerd pad doorsturen (RPF) naar een verschillende Ingratie-router zijn. Als beide Ingress PE-routers naar de standaard MDT doorsturen, start het activeringsmechanisme in en de ene Ingress PE wint het opslagmechanisme en de andere verliest zodat één en slechts één Ingoers-PE de klant (C-) (S,G) naar de MDT blijft doorsturen. Als het activeringsmechanisme om welke reden dan ook niet op de standaard MDT is gestart, is het echter mogelijk dat beide Ingress PE-routers beginnen met het verzenden van het multicast-verkeer naar één Data-MDT dat zij in werking stellen. Omdat het verkeer niet meer op de standaard MDT is, maar op Data MDTs, ontvangen beide Ingress PE-routers niet het C-(S,G) verkeer van elkaar op de MDT/Tunnel-interface. Dit kan leiden tot een aanhoudend dubbel verkeer stroomafwaarts. Dit document legt de oplossing voor dit probleem uit.

## Mechanisme instellen op de standaard-MDT

De informatie in deze sectie geldt voor de standaard MDT, ongeacht het kernboomprotocol. Het gekozen kernboomprotocol is Protocol Independent Multicast (PIM).

Cisco IOS wordt gebruikt voor de voorbeelden, maar alles wat wordt vermeld is zowel van toepassing op Cisco IOS-XR. Alle multicast groepen die worden gebruikt, zijn Source Specific Multicast (SSM)-groepen.

Kijk naar afbeelding 1. Dual-Homed-Source-1. Er zijn twee Ingress PE-routers (PE1 en PE2) en twee uitgaande PE-routers (PE3 en PE4). De bron is op CE1 met IP-adres 10.100.1.6. CE1 is tweeledig naar PE1 en PE2.



Afbeelding 1. Tweevoudige bron-1

De configuratie op alle PE-routers (Routeswitchmachine (RD) kan verschillen van de PE-routers) is:

```
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
 !
```

Om beide Ingress-routers in staat te stellen de multicast-stream (10.10.1.6.232.1.1.1) naar de standaard MDT te sturen, moeten ze allebei een account op een uitgaande PE ontvangen. Kijk naar de topologie in figuur 1. Dual-Homed-Source-1. U kunt zien dat, als alle kosten van de randverbindingen gelijk zijn en alle kosten van de kernverbindingen gelijk zijn, PE3 naar PE1 en PE4 RPF naar PE2 zal gaan voor (10.10.1.6.232.1.1.1). Ze zijn allebei RPF voor hun dichtstbijzijnde Ingress PE. Deze output bevestigt dit:

```
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 heeft RPF naar 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 heeft RPF tot PE2. PE3 beschouwt PE1 als RPF-buurman omdat de eenmakingsroute naar 10.100.1.6/32 in Virtual Routing/Forwarding (VRF) één de beste is via PE1. PE3 ontvangt de route 10.100.1.6/32 daadwerkelijk van zowel PE1 als PE2. Alle criteria in het Border Gateway Protocol (BGP) Best Path Calculator zijn dezelfde, behalve: voor de kosten voor het BGP-adres van de volgende hop.

```
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

```

Het beste door PE3 gekozen pad is het pad dat door PE1 wordt geadverteerd omdat dat de laagste IGP-kosten (11) heeft, tegen de IGP-kosten (21) naar PE2. Voor PE4 is het de omgekeerde weg. De topologie toont aan dat er van PE3 naar PE1 slechts één hop is, terwijl van PE3 naar PE2 twee hop bestaat. Aangezien alle links dezelfde IGP-kosten hebben, wordt het pad van PE1 als het beste gekozen.

Multicast Routing Information Base (MRIB) voor (10.100.1.6.232.1.1.1) ziet er als volgt uit op PE1 en PE2 wanneer er nog geen multicast verkeer is:

```

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 en PE2 kregen beide een PIM-aansluiting voor (10.100.1.6.232.1.1.1). De interface Tunnel0 is in de vertrekkende interfacekaart (OIL) voor de multicast-ingang op beide routers.

Het multicast verkeer begint te stromen voor (10.100.1.6.232.1.1.1). "Debug ip pim vrf one 232.1.1.1" en "debug ip routing vrf one 232.1.1.1" laten zien dat het multicast verkeer naar Tunnel (in de OIL) toeneemt van beide Ingeer-PE-routers veroorzaakt dat het activeringsmechanisme draait.

## 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
```

Als de metriek en de afstand van beide routers gelijk zijn aan Bron 10.100.1.6, dan is er een trekbreker om de trekwinnaar te bepalen. De werkbalk is het hoogste IP-adres van de PIM buurman op de Tunnel0 (standaard MDT). In dit geval is dit 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 Address	Interface	Uptime/Expires	Ver	DR 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/ Mode	Nbr Count	Query Intvl	DR Prior	DR
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 heeft Tunnel0 uit de OIL van de multicast ingang verwijderd vanwege de berichten. Aangezien de OLIE leeg werd, wordt de multicast-ingang gesnoeid.

```
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
```

PE2 heeft de A-vlag op de interface Tunnel0 gezet, omdat het de belangrijkste winnaar is.

```
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 stuurt periodiek een waarschuwing op Tunnel0 (Standaard MDT), vlak voordat de timer verstrijkt. Als zodanig blijft PE2 de winnaar van de aanval.

```
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]
```

## Conclusie

Het aanhoudingsmechanisme werkt ook met een tunnelinterface in de OIL. De berichten worden uitgewisseld via de standaard MDT wanneer de Ingress PE-routers multicast verkeer op de gekoppelde tunnelinterface C-C-(S,G) ontvangen die zich in de OIL bevindt.

## Mechanisme voor controle met MDT's

Het grootste gedeelte van de tijd dat MDT's worden geconfigureerd, zal het activeringsmechanisme nog draaien op de standaardMDT omdat het C-(S,G)-verkeer pas na drie seconden wordt overgeschakeld van de standaard MDT naar de Data MDT's. Dan gebeurt hetzelfde als eerder beschreven. Merk op dat **er slechts één tunnelinterface is per multicast-enabled VRF**: In het geval van de standaardMDT en alle MDT's worden slechts één tunnelinterface gebruikt. Deze tunnelinterface wordt gebruikt in OIL op de IP-routers van de uitgang of als een RPF-interface op de PE-routers van de uitgang.

In sommige gevallen is het mogelijk dat het activa-mechanisme niet wordt geactiveerd voordat de MDT's worden gesignaleerd. Dan is het mogelijk dat het C-(S,G) multicast verkeer op een Data MDT op zowel Ingress PE-routers PE1 als PE2 begint te worden doorgestuurd. In dergelijke gevallen kan dit leiden tot permanent dubbel C-(S,G) multicast verkeer over het MPLS-kernnetwerk. Om dit te voorkomen, werd deze oplossing ten uitvoer gelegd: Wanneer een Ingress PE-router een andere Ingress PE-router ziet, kondigt deze een MDT aan waarvoor de PE-router ook een Ingress PE-router is, wordt deze ook gebruikt om MDT in gegevens te verwerken. In beginsel zouden alleen Groot PE-routers (die een downstreamontvanger hebben) zich bij de MDT aansluiten. Omdat Ingress PE-routers zich aansluiten bij de Data MDT die is aangekondigd door andere Ingress PE-routers, leidt dit tot de Ingress PE-router die multicast verkeer ontvangt van de Tunnelinterface die aanwezig is in de OIL. Dit zet daarom het opslagmechanisme in werking en leidt tot een van de Ingress PE-routers om te stoppen met het verzenden van het C-(S,G) multicast-verkeer naar zijn MDT (met tunnelinterface), terwijl het andere Ingress-verkeer (de winnaar van het beroep) kan het multicastverkeer C-(S,G) naar zijn MDT blijven doorsturen.

Ga er bij het volgende voorbeeld van uit dat de IP-routers PE1 en PE2 het multicast-verkeer C-(S,G) van elkaar op de standaard MDT nooit hebben gezien. Het verkeer is slechts drie seconden op de standaard MDT en het is niet moeilijk te begrijpen dat dit kan gebeuren als er bijvoorbeeld tijdelijk verkeersverlies op het kernnetwerk is.

De configuratie voor Data MDT wordt toegevoegd aan alle PE-routers. De configuratie op alle PE-routers (RD kan verschillend zijn op de PE-routers) is:

```
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
 !
```

Zodra PE1 en PE2 het verkeer van de bron zien, creëren zij een C-(S,G)-ingang. Beide IP-routers sturen het multicast verkeer C-D(S,G) naar de standaard MDT. PE-routers PE3 en PE4 ontvangen het multicast verkeer en verzenden dit. Als gevolg van een tijdelijk probleem ziet PE2 het verkeer

van PE1 niet en omgekeerd op de standaard MDT. Ze verzenden allebei een Data MDT Samenvoegen Type Lengte Waarde (TLV) uit op de Standaard MDT.

Als er geen C-(S,G) verkeer is, ziet u deze multicast toestand op de Ingoers PE-routers:

```
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
```

De y-flag is nog niet ingesteld. Beide IP-routers hebben de Tunnel0-interface in de OLIE. Dit is te wijten aan het feit dat PE3 RPF naar PE1 heeft en PE4 RPF naar PE2 voor C-(S,G).

Wanneer multicast verkeer voor C-(S,G) begint te stromen, zowel PE1 als PE2 vooruit het verkeer. De drempel voor Data MDT wordt overschreden op zowel Ingress PE-routers en beide een Data MDT Join TLV verzenden en na drie seconden beginnen met verzenden naar hun Data MDT. Merk op dat PE1 zich aansluit bij de door PE2 en PE2 verkregen gegevens MDT die door PE1 zijn verkregen.



```

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

```

Zowel PE1 als PE ontvangen verkeer voor C-(S,G) op de Tunnel0-interface (maar nu van de Data MDT, niet de Default MDT) en het assert mechanism knikt in. Alleen PE2 blijft het C-(S,G)-verkeer op zijn MDT doorsturen:

```

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 heeft niet langer de tunnelinterface in de 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
```

**PE2 heeft de A-vlag op de Tunnel0-interface ingesteld:**

```
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

## Conclusie

Het activeringsmechanisme werkt ook wanneer MDT's worden gebruikt. De berichten worden uitgewisseld via de standaard MDT wanneer de Ingress PE-routers multicast verkeer op de gekoppelde tunnelinterface C-C-(S,G) ontvangen die zich in de OIL bevindt.