

MDT de origem e dados dual-homed em mVPN

Contents

[Introduction](#)

[O problema](#)

[Mecanismo de asserção no MDT padrão](#)

[Conclusão](#)

[Mecanismo de asserção com MDTs de dados](#)

[Conclusão](#)

Introduction

Este documento descreve o mVPN (Multicast Virtual Provider Network) com origem dual-homed e MDT de dados (Multicast Distribution Tree). Um exemplo no Cisco IOS[®] é usado para ilustrar o comportamento.

O problema

Se uma origem no mundo mVPN é dual-homed para dois roteadores de Borda do Provedor de Entrada (PE - Ingress Provider Edge), pode ser possível que os dois roteadores de PE de entrada encaminhem o tráfego para uma (S,G) na nuvem Multiprotocol Label Switching (MPLS - Multiprotocol Label Switching). Isso é possível se, por exemplo, houver dois roteadores PE de saída e cada RPF (Reverse Path Forwarding) para um roteador PE de entrada diferente. Se ambos os roteadores de PE de entrada forem encaminhados para o MDT padrão, o mecanismo de asserção será ativado e um PE de entrada ganhará o mecanismo de asserção e o outro perderá, de modo que um e apenas um PE de entrada continue a encaminhar o cliente (C-) (S,G) para o MDT. No entanto, se por algum motivo o mecanismo de asserção não iniciou no MDT padrão, então é possível que ambos os roteadores de PE de entrada comecem a transmitir o tráfego multicast C-(S,G) em um Data-MDT que iniciam. Como o tráfego não está mais no MDT padrão, mas nos MDTs de dados, ambos os roteadores de PE de entrada não recebem o tráfego C-(S,G) um do outro na interface MDT/Tunnel. Isso pode causar tráfego duplicado persistente em downstream. Este documento explica a solução para esse problema.

Mecanismo de asserção no MDT padrão

As informações nesta seção são verdadeiras para o MDT padrão, independentemente do protocolo da árvore central. O protocolo de árvore principal escolhido é o Protocol Independent Multicast (PIM).

O Cisco IOS é usado para os exemplos, mas tudo o que é mencionado se aplica igualmente ao Cisco IOS-XR. Todos os grupos multicast usados são grupos de Multicast Específico de Origem (SSM - Source Specific Multicast).

Veja a Figura 1. Dual-Homed-Source-1. Há dois roteadores PE de entrada (PE1 e PE2) e dois roteadores PE de saída (PE3 e PE4). A origem está no CE1 com o endereço IP 10.100.1.6. CE1 é dual-homed para PE1 e PE2.

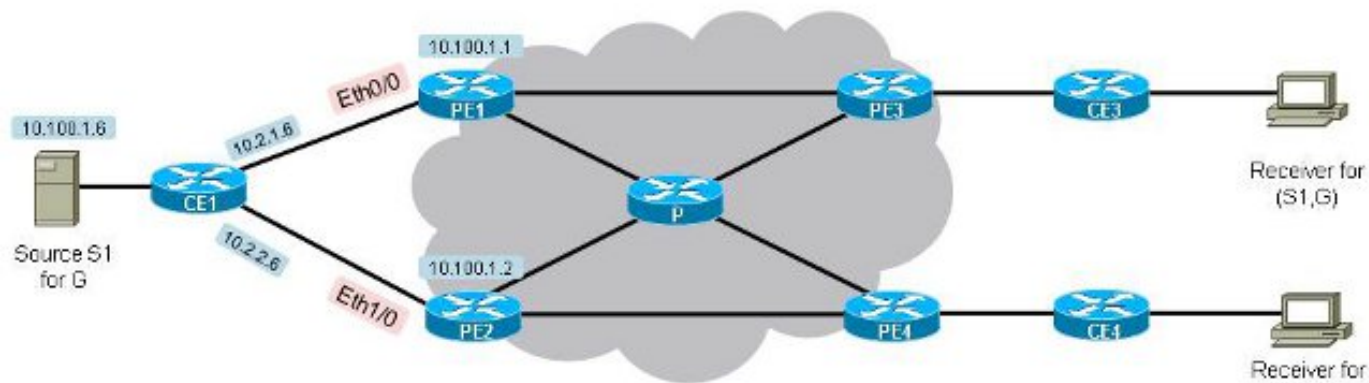


Figura 1. Fonte-1 dual-homed

A configuração em todos os roteadores PE (o Distinguisher de Rota (RD) pode ser diferente nos roteadores PE) é:

```
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
 !
```

Para que ambos os roteadores de PE de entrada comecem a encaminhar o fluxo multicast (10.100.1.6,232.1.1.1) para o MDT padrão, ambos devem receber um Join de um PE de saída. Examine a topologia na Figura 1. Dual-Homed-Source-1. Você pode ver que, por padrão, se todos os custos dos links de borda forem os mesmos e todos os custos dos links de núcleo forem os mesmos, então o PE3 irá RPF em direção ao PE1 e o PE4 irá RPF em direção ao PE2 para (10.100.1.6,232.1.1.1). Ambos são RPF para o PE de entrada mais próximo. Esta saída confirma:

```
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 tem RPF para 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 tem RPF para PE2. O motivo pelo qual PE3 escolhe PE1 como o vizinho RPF é que a rota unicast em direção a 10.100.1.6/32 em Virtual Routing/Forwarding (VRF) one é a melhor via PE1. O PE3 realmente recebe a rota 10.100.1.6/32 de PE1 e PE2. Todos os critérios no algoritmo de cálculo do melhor caminho do protocolo BGP (Border Gateway Protocol) são os mesmos, exceto o custo para o endereço do próximo salto BGP.

```
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:0x0005:0x0000000640200
  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 nlabel/29
rx pathid: 0, tx pathid: 0x0

```

O melhor caminho escolhido pelo PE3 é o caminho anunciado pelo PE1 porque ele tem o menor custo do Interior Gateway Protocol (IGP) (11), em comparação com o custo IGP (21) para o PE2. Para PE4 é o contrário. A topologia revela que de PE3 a PE1 há apenas um salto, enquanto de PE3 a PE2 há dois saltos. Como todos os links têm o mesmo custo IGP, PE3 escolhe o caminho de PE1 como o melhor.

A Base de Informações de Roteamento Multicast (MRIB - Multicast Routing Information Base) para (10.100.1.6,232.1.1.1) é semelhante a esta em PE1 e PE2 quando ainda não há tráfego multicast:

```

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 e PE2 recebem um PIM Join para (10.100.1.6,232.1.1.1). A interface Tunnel0 está na Lista de Interface de Saída (OIL) para a entrada multicast em ambos os roteadores.

O tráfego multicast começa a fluir para (10.100.1.6,232.1.1.1). "Debug ip pim vrf one 232.1.1.1" e "debug ip mrouting vrf one 232.1.1.1" mostram que a chegada do tráfego multicast ao Tunnel0 (no OIL) de ambos os roteadores de entrada PE, faz com que o mecanismo assert seja executado.

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
```

Se a métrica e a distância forem iguais para ambos os roteadores em direção à Origem 10.100.1.6, então haverá um empate-breaker para determinar o vencedor do assert. O empate-breaker é o endereço IP mais alto do vizinho PIM no Tunnel0 (MDT padrão). Nesse caso, é o 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 removeu Tunnel0 do OIL da entrada multicast devido às asserções. Como o OIL ficou vazio, a entrada multicast é podada.

```
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
```

O PE2 tem o sinalizador A definido na interface Tunnel0, porque é o vencedor do assert.

```
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
```

O PE2 envia periodicamente uma asserção no Tunnel0 (MDT padrão), logo antes do temporizador de asserção expirar. Como tal, o PE2 continua sendo o vencedor da asserção.

```
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]
```

Conclusão

O mecanismo assert também funciona com uma interface Tunnel no OIL. As asserções são trocadas pelo MDT padrão quando os roteadores do PE de entrada recebem tráfego multicast C-(S,G) na interface de túnel associada que está no OIL.

Mecanismo de asserção com MDTs de dados

Na maioria das vezes em que os MDTs de dados são configurados, o mecanismo de asserção ainda será executado no MDT padrão, pois o tráfego C-(S,G) é comutado somente do MDT padrão para os MDTs de dados após três segundos. O mesmo ocorre como descrito anteriormente. Observe que **há apenas uma interface de túnel por VRF habilitado para multicast**: o MDT padrão e todos os MDTs de dados usam apenas uma interface de túnel. Essa interface de túnel é usada no OIL nos roteadores PE de entrada ou como uma interface RPF nos roteadores PE de saída.

Em alguns casos, é possível que o mecanismo de asserção não seja acionado antes que os MDTs de dados sejam sinalizados. Em seguida, é possível que o tráfego multicast C-(S,G) comece a ser encaminhado em um MDT de dados nos roteadores PE de entrada PE1 e PE2. Nesses casos, isso pode levar ao tráfego multicast C-(S,G) duplicado permanente através da rede central MPLS. Para evitar isso, essa solução foi implementada: quando um roteador de PE de entrada vê outro roteador de PE de entrada anunciar um MDT de dados para o qual o roteador PE também é um roteador de PE de entrada, ele junta esse MDT de dados. Em princípio, somente os roteadores PE de saída (que têm um receptor downstream) se uniram ao MDT de dados. Como os roteadores de PE de entrada se juntam ao MDT de dados anunciado por outros roteadores de PE de entrada, ele leva ao roteador de PE de entrada que recebe tráfego multicast da interface de túnel que está presente no OIL e, portanto, aciona o mecanismo de asserção e leva a um dos roteadores de PE de entrada para parar de encaminhar o tráfego multicast C-(S,G) para seu MDT de dados (com a interface de túnel), enquanto os outros PE (o vencedor da asserção) pode continuar a encaminhar o tráfego multicast C-(S,G) para seu MDT de dados.

Para o próximo exemplo, suponha que os roteadores PE de entrada PE PE PE1 e PE2 nunca viram o tráfego multicast C-(S,G) um do outro no MDT padrão. O tráfego está no MDT padrão por apenas três segundos e não é difícil entender que isso pode ocorrer se houver, por exemplo, perda temporária de tráfego na rede central.

A configuração para Data MDT é adicionada a todos os roteadores PE. A configuração em todos os roteadores PE (o RD pode ser diferente nos roteadores PE) é:

```
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
 !
```

Assim que o PE1 e o PE2 veem o tráfego da origem, eles criam uma entrada C-(S,G). Ambos os roteadores de entrada PE encaminham o tráfego multicast C-(S,G) para o MDT padrão. Os roteadores PE3 e PE4 de saída recebem o tráfego multicast e o encaminham. Devido a um

problema temporário, o PE2 não vê o tráfego do PE1 e vice-versa no MDT padrão. Ambos enviam um TLV (Data MDT Join Type Length Value, valor de comprimento do tipo de junção de dados) para fora no MDT padrão.

Se não houver tráfego C-(S,G), você verá esse estado de multicast nos roteadores PE de entrada:

```
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
```

O y-flag ainda não está definido. Ambos os roteadores de entrada PE têm a interface Tunnel0 no OIL. Isso se deve ao fato de que PE3 tem RPF para PE1 e PE4 ter RPF para PE2 para C-(S,G).

Quando o tráfego multicast para C-(S,G) começa a fluir, tanto PE1 como PE2 encaminham o tráfego. O limite de MDT de dados é cruzado em ambos os roteadores de PE de entrada e ambos enviam um TLV de junção de MDT de dados e, após três segundos, começam a encaminhar para o MDT de dados. Observe que PE1 junta o MDT de Dados originado por PE2 e PE2 ao MDT de

Dados originado por PE1.

```
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
```

O PE1 e o PE recebem tráfego para C-(S,G) na interface Tunnel0 (mas agora do MDT de dados, não o MDT padrão) e o mecanismo assert é iniciado. Somente o PE2 continua a encaminhar o tráfego C-(S,G) em seu MDT de dados:

```
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
```

O PE1 não tem mais a interface de túnel no 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
```

O PE2 tem o sinalizador A definido na interface Tunnel0:

```
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

Conclusão

O mecanismo assert também funciona quando MDTs de dados são usados. As asserções são trocadas pelo MDT padrão quando os roteadores do PE de entrada recebem tráfego multicast C-(S,G) na interface de túnel associada que está no OIL.