

MDT source et données à double résidence dans mVPN

Contenu

[Introduction](#)

[Le problème](#)

[Mécanisme d'affirmation sur le MDT par défaut](#)

[Conclusion](#)

[Mécanisme d'affirmation avec les MDT de données](#)

[Conclusion](#)

Introduction

Ce document décrit mVPN (Multicast Virtual Provider Network) avec source double et MDT de données (Multicast Distribution Tree). Un exemple de Cisco IOS[®] est utilisé pour illustrer le comportement.

Le problème

Si une source dans le monde mVPN est à double résidence vers deux routeurs PE (Inbound Provider Edge), il peut être possible pour les deux routeurs PE d'entrée de transférer le trafic d'un (S, G) dans le cloud MPLS (Multiprotocol Label Switching). Cela est possible si, par exemple, il existe deux routeurs PE de sortie et chaque RPF (Reverse Path Forwarding) vers un routeur PE de sortie différent. Si les deux routeurs PE d'entrée se dirigent vers le MDT par défaut, le mécanisme d'assertion s'enclenchera et un PE d'entrée gagnera le mécanisme d'assertion et l'autre perdra de sorte qu'un seul PE d'entrée continue à transférer le Client (C-) (S, G) sur le MDT. Cependant, si, pour une raison quelconque, le mécanisme d'assertion n'a pas démarré sur le MDT par défaut, il est alors possible pour les deux routeurs PE d'entrée de commencer à transmettre le trafic de multidiffusion C-(S, G) sur un Data-MDT qu'ils lancent. Comme le trafic n'est plus sur le MDT par défaut, mais sur les MDT de données, les deux routeurs PE d'entrée ne reçoivent pas le trafic C-(S, G) l'un de l'autre sur l'interface MDT/Tunnel. Cela peut entraîner un trafic dupliqué persistant en aval. Ce document explique la solution à ce problème.

Mécanisme d'affirmation sur le MDT par défaut

Les informations de cette section sont vraies pour le MDT par défaut, quel que soit le protocole de l'arborescence principale. Le protocole d'arborescence principale choisi est le protocole PIM (Protocol Independent Multicast).

Cisco IOS est utilisé pour les exemples, mais tout ce qui est mentionné s'applique également à Cisco IOS-XR. Tous les groupes de multidiffusion utilisés sont des groupes SSM (Source Specific Multicast).

Regardez la figure 1. Double-Home-Source-1. Il existe deux routeurs PE d'entrée (PE1 et PE2) et deux routeurs PE de sortie (PE3 et PE4). La source se trouve sur CE1 avec l'adresse IP

10.100.1.6. CE1 est à double résidence vers PE1 et PE2.

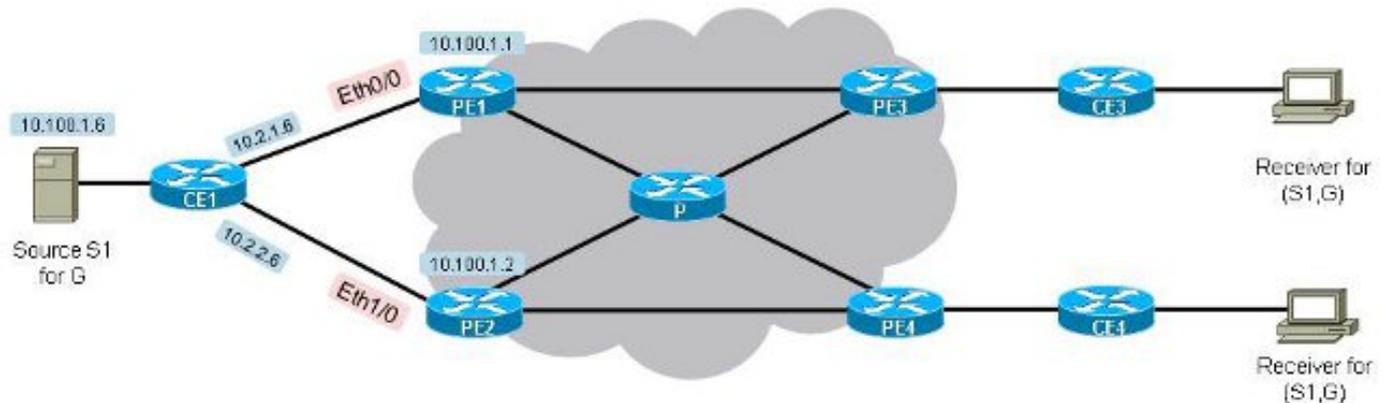


Figure 1. Source à double résidence-1

La configuration de tous les routeurs PE (le séparateur de route (RD) peut être différent sur les routeurs PE) est la suivante :

```
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
!
```

Pour que les deux routeurs PE d'entrée commencent à transférer le flux de multidiffusion (10.100.1.6,232.1.1.1) vers le MDT par défaut, ils doivent tous deux recevoir une jointure d'un PE de sortie. Examinez la topologie de la Figure 1. Double-Home-Source-1. Vous pouvez voir que par défaut, si tous les coûts des liaisons de périphérie sont identiques et que tous les coûts des liaisons de coeur de réseau sont identiques, alors PE3 RPF vers PE1 et PE4 RPF vers PE2 pour (10.100.1.6,232.1.1.1). Ils sont tous deux RPF à leur PE d'entrée le plus proche. Ce résultat confirme ceci :

```
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 dispose de RPF vers 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 dispose de RPF à PE2. La raison pour laquelle PE3 choisit PE1 comme voisin RPF est que la route de monodiffusion vers 10.100.1.6/32 dans le VRF (Virtual Routing/Forwarding) 1 est la meilleure via PE1. PE3 reçoit la route 10.100.1.6/32 à la fois de PE1 et de PE2. Tous les critères de l'algorithme BGP (Border Gateway Protocol) Best Path Calculation Algorithm sont identiques, à l'exception du coût vers l'adresse de tronçon suivant 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:0x0005:0x000000640200
  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

```

Le meilleur chemin choisi par PE3 est le chemin annoncé par PE1, car il présente le coût IGP (Interior Gateway Protocol) le plus bas (11), par rapport au coût IGP (21) vers PE2. Pour PE4, c'est l'inverse. La topologie indique que de PE3 à PE1 il n'y a qu'un seul saut, tandis que de PE3 à PE2 il y a deux sauts. Puisque toutes les liaisons ont le même coût IGP, PE3 choisit le chemin de PE1 comme étant le meilleur.

La base d'informations de routage multidiffusion (MRIB) pour (10.100.1.6,232.1.1.1) ressemble à ceci sur PE1 et PE2 lorsqu'il n'y a pas encore de trafic multidiffusion :

```

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 et PE2 ont tous deux reçu une adhésion PIM pour (10.100.1.6,232.1.1.1). L'interface Tunnel0 figure dans la liste OIL (Outgoing Interface List) pour l'entrée de multidiffusion sur les deux routeurs.

Le trafic de multidiffusion commence à circuler pour (10.100.1.6,232.1.1.1). « Debug ip pim vrf one 232.1.1.1 » et « debug ip mrouting vrf one 232.1.1.1 » nous montrent que l'arrivée du trafic de multidiffusion sur Tunnel0 (dans l'OIL) des deux routeurs PE d'entrée entraîne l'exécution du mécanisme d'assertion.

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
```

Si la métrique et la distance sont identiques des deux routeurs vers la source 10.100.1.6, alors il y a un séparateur pour déterminer le gagnant de l'assertion. Le séparateur est l'adresse IP la plus élevée du voisin PIM sur le tunnel0 (MDT par défaut). Dans ce cas, il s'agit de 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 a supprimé Tunnel0 de l'OIL de l'entrée de multidiffusion en raison des asserts. Depuis que l'OIL est devenu vide, l'entrée de multidiffusion est élaguée.

```
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
```

L'indicateur A de PE2 est défini sur l'interface Tunnel0, car il s'agit du gagnant de l'assertion.

```
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 envoie régulièrement un assertion sur Tunnel0 (MDT par défaut), juste avant l'expiration du minuteur d'assertion. En tant que tel, PE2 reste le gagnant.

```
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]
```

Conclusion

Le mécanisme d'assertion fonctionne également avec une interface de tunnel dans l'OIL. Les assertions sont échangées sur le MDT par défaut lorsque les routeurs PE d'entrée reçoivent du trafic de multidiffusion C-(S, G) sur l'interface de tunnel associée qui se trouve dans l'OIL.

Mécanisme d'affirmation avec les MDT de données

La plupart du temps, lorsque des MDT de données sont configurés, le mécanisme d'assertion s'exécute toujours sur le MDT par défaut, car le trafic C-(S, G) est commuté du MDT par défaut aux MDT de données après trois secondes. Il en va de même pour ce qui est de la description précédente. Notez qu'il **n'existe qu'une seule interface de tunnel par VRF multidiffusion** : le MDT par défaut et tous les MDT de données utilisent une seule interface de tunnel. Cette interface de tunnel est utilisée dans l'OIL sur les routeurs PE d'entrée ou comme interface RPF sur les routeurs PE de sortie.

Dans certains cas, il est possible que le mécanisme d'assertion ne soit pas déclenché avant que les MDT de données ne soient signalés. Il est alors possible que le trafic de multidiffusion C-(S, G) commence à être transféré sur un MDT de données sur les routeurs PE1 et PE2 d'entrée. Dans de tels cas, cela pourrait conduire à un trafic de multidiffusion C-(S, G) en double permanent sur le réseau principal MPLS. Pour éviter cela, cette solution a été mise en oeuvre : lorsqu'un routeur PE d'entrée voit un autre routeur PE d'entrée annoncer un MDT de données pour lequel le routeur PE est également un routeur PE d'entrée, il joint ce MDT de données. En principe, seuls les routeurs PE de sortie (qui ont un récepteur en aval) rejoignent le MDT de données. Comme les routeurs PE d'entrée rejoignent le MDT de données annoncé par d'autres routeurs PE d'entrée, il conduit le routeur PE d'entrée à recevoir le trafic de multidiffusion de l'interface de tunnel qui est présente dans l'OIL, ce qui déclenche le mécanisme d'assertion et conduit l'un des routeurs PE d'entrée à arrêter de transférer le trafic de multidiffusion C-(S, G) sur son MDT de données (avec l'interface de tunnel), tandis que l'autre (le gagnant de l'assertion) peut continuer à transférer le trafic de multidiffusion C-(S, G) sur son MDT de données.

Pour l'exemple suivant, supposons que les routeurs PE d'entrée PE PE1 et PE2 n'ont jamais vu le trafic de multidiffusion C-(S, G) l'un de l'autre sur le MDT par défaut. Le trafic est sur le MDT par défaut pendant trois secondes seulement et il n'est pas difficile de comprendre que cela peut se produire si, par exemple, il y a une perte de trafic temporaire sur le réseau principal.

La configuration de Data MDT est ajoutée à tous les routeurs PE. La configuration de tous les routeurs PE (la distance peut être différente sur les routeurs PE) est la suivante :

```
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
 !
```

Dès que PE1 et PE2 voient le trafic de la source, ils créent une entrée C-(S, G). Les deux routeurs PE d'entrée transmettent le trafic de multidiffusion C-(S, G) sur le MDT par défaut. Les routeurs PE de sortie PE3 et PE4 reçoivent le trafic de multidiffusion et le transmettent. En raison d'un problème temporaire, PE2 ne voit pas le trafic en provenance de PE1 et inversement sur le MDT par défaut. Ils envoient tous deux une valeur TLV (Data MDT Join Type Length Value) sur le MDT par défaut.

S'il n'y a pas de trafic C-(S, G), vous voyez cet état de multidiffusion sur les routeurs PE d'entrée :

```
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
```

L'indicateur y n'est pas encore défini. Les deux routeurs PE d'entrée ont l'interface Tunnel0 dans l'OIL. Ceci est dû au fait que PE3 a RPF vers PE1 et PE4 a RPF vers PE2 pour C-(S, G).

Lorsque le trafic de multidiffusion pour C-(S, G) commence à circuler, PE1 et PE2 transmettent le trafic. Le seuil de MDT de données est franchi sur les deux routeurs PE d'entrée et les deux

envoient un TLV de jointure MDT de données et après trois secondes commencent à transmettre sur leur MDT de données. Notez que PE1 rejoint le MDT de données fourni par PE2 et PE2 rejoint le MDT de données fourni par 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
```

PE1 et PE reçoivent le trafic pour C-(S, G) sur l'interface Tunnel0 (mais maintenant à partir du MDT de données, et non du MDT par défaut) et le mécanisme d'assertion se déclenche. Seul PE2 continue à transférer le trafic C-(S, G) sur son MDT de données :

```
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 n'a plus l'interface de tunnel dans l'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
```

L'indicateur A de PE2 est défini sur l'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

Conclusion

Le mécanisme d'assertion fonctionne également lorsque des MDT de données sont utilisés. Les assertions sont échangées sur le MDT par défaut lorsque les routeurs PE d'entrée reçoivent du trafic de multidiffusion C-(S, G) sur l'interface de tunnel associée qui se trouve dans l'OIL.