

Source et données dual-homed MDT dans le mVPN

Contenu

[Introduction](#)

[Le problème](#)

[Affirmez le mécanisme sur le par défaut MDT](#)

[Conclusion](#)

[Affirmez le mécanisme avec des données MDTs](#)

[Conclusion](#)

Introduction

Ce document décrit le mVPN (réseau du fournisseur virtuel de Multidiffusion) avec la doubles source et données autoguidées MDT (arbre de distribution de Multidiffusion). Un exemple dans le Cisco IOS® est utilisé afin de montrer le comportement.

Le problème

Si un source in le monde de mVPN est double autoguidé à deux Routeurs de Provider Edge d'entrée (PE), il pourrait être possible aux deux Routeurs de PE d'entrée aux les deux le trafic en avant pour un (S, G) dans le nuage de Commutation multiprotocole par étiquette (MPLS). C'est possible si, par exemple, il y a deux Routeurs de PE de sortie et chaque Reverse Path Forwarding (RPF) à un différent routeur PE d'entrée. Si les deux Routeurs de PE d'entrée en avant sur le par défaut MDT, alors le mécanisme d'affirmation donneront un coup de pied dedans et un PE d'entrée gagne le mécanisme d'affirmation et l'autre perd de sorte qu'un et seulement un PE d'entrée continue à expédier le client (c) (S, G) sur le MDT. Cependant, si pour une raison quelconque, le mécanisme d'affirmation ne commençait pas sur le par défaut MDT, puis il est possible que les deux Routeurs de PE d'entrée commencent à transmettre le c (S, G) le trafic de multidiffusion sur une données-MDT QU'ils initient. Puisque le trafic est non sur le par défaut MDT plus, mais sur des données MDTs, les deux Routeurs de PE d'entrée ne reçoivent pas le c (S, G) le trafic entre eux sur l'interface MDT/Tunnel. Ceci peut entraîner l'en aval en double persistant du trafic. Ce document explique la solution au problème.

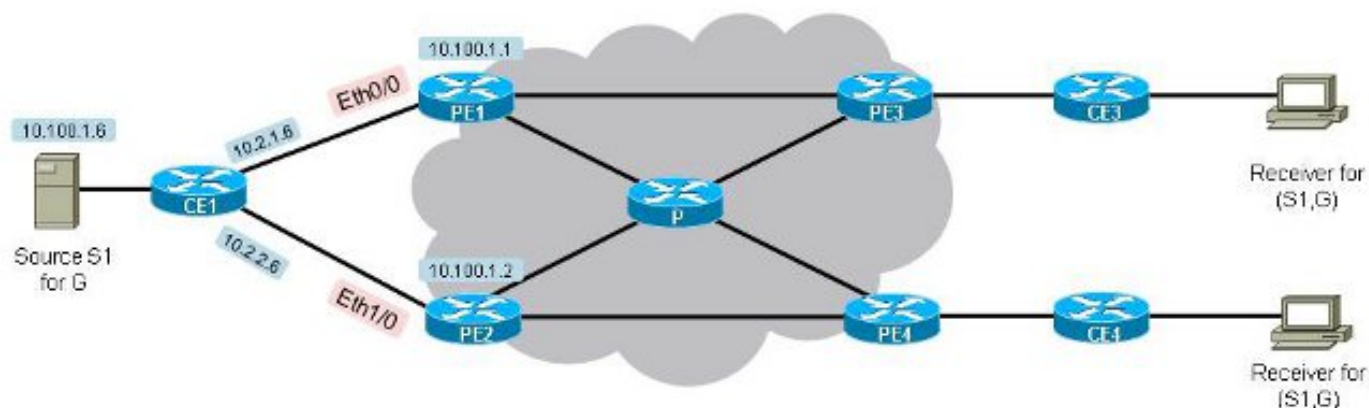
Affirmez le mécanisme sur le par défaut MDT

Les informations dans cette section jugent vrai pour le par défaut MDT, indépendamment du principal protocole d'arborescence. Le principal protocole choisi d'arborescence est le Protocol Independent Multicast (PIM).

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

Regardez le schéma 1. Dual-Homed-Source-1. Il y a deux Routeurs de PE d'entrée (PE1 et PE2)

et deux Routeurs de PE de sortie (PE3 et PE4). La source est à CE1 avec l'adresse IP 10.100.1.6. CE1 est double autoguidé à PE1 et à PE2.



Le schéma 1. Dual-Homed-Source-1

La configuration sur tous les Routeurs de PE (le moteur de distinction de route (RD) peut être différent sur les Routeurs de PE) est :

```
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
 !
```

Afin d'obliger les deux Routeurs de PE d'entrée pour commencer à expédier le flot de Multidiffusion (10.100.1.6,232.1.1.1) sur le par défaut MDT, ils doivent chacun des deux recevoir un jointre d'un PE de sortie. Regardez la topologie dans Figure1. Dual-Homed-Source-1. Vous pouvez voir cela par défaut, si tous les coûts des liens de périphérie sont identiques et tous les coûts des principaux liens sont identiques, alors PE3 RPF vers PE1 et PE4 RPF vers PE2 pour (10.100.1.6,232.1.1.1). Ils les deux RPF à leur PE d'entrée plus étroit. Cette sortie 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 a le RPF à 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 a le RPF à PE2. La raison pour laquelle PE3 sélectionne PE1 car le voisin RPF est que l'artère d'unicast vers 10.100.1.6/32 dans le routage virtuel/expédition (VRF) un est le meilleur par l'intermédiaire de PE1. PE3 reçoit réellement l'artère 10.100.1.6/32 de PE1 et de PE2. Tous les critères dans le meilleur algorithme de calcul de chemin de Protocole BGP (Border Gateway Protocol) sont identiques, excepté le coût vers l'adresse de bgp next-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:0x0005:0x0000000640200
  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 nolabel/29
rx pathid: 0, tx pathid: 0x0

```

Le meilleur chemin choisi par PE3 est le chemin annoncé par PE1 parce que cela a le plus bas coût de Protocole IGP (Interior Gateway Protocol) (11), contre l'IGP coûté (21) vers PE2. Pour PE4 c'est l'inverse. La topologie indique que de PE3 à PE1 il y a seulement un saut, alors que de PE3 à PE2 il y a deux sauts. Puisque tous les liens ont le même coût d'IGP, PE3 sélectionne le chemin de PE1 comme meilleur.

Le Routing Information Base de Multidiffusion (MRIB) pour (10.100.1.6,232.1.1.1) ressemble à ceci sur PE1 et PE2 quand il n'y a aucun trafic de multidiffusion pourtant :

```

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 chacun des deux ont reçu un PIM Join pour (10.100.1.6,232.1.1.1). L'interface Tunnel0 est dans la liste d'interfaces en sortie (HUILE) pour l'entrée multicast sur les deux Routeurs.

Les débuts du trafic de multidiffusion à circuler pour (10.100.1.6,232.1.1.1). Le « vrf un 232.1.1.1 de debug ip pim » et le « vrf un 232.1.1.1 de debug ip mrouting » nous prouvent que l'arrivée du trafic de multidiffusion sur Tunnel0 (dans l'HUILE) des deux Routeurs de PE d'entrée, fait fonctionner le mécanisme d'affirmation.

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 mesure et la distance est les mêmes des deux Routeurs vers la source 10.100.1.6, alors il y a un lien-briseur afin de déterminer le gagnant d'affirmation. Le lien-briseur est l'adresse IP la plus élevée du voisin PIM sur le Tunnel0 (MDT par défaut). Dans ce cas, c'est 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

Tunnel0 retiré par PE1 de l'HUILE de l'entrée multicast en raison du affirme. Puisque l'HUILE est devenue vide, l'entrée multicast est taillée.

```
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 a l'Un-indicateur réglé sur l'interface Tunnel0, parce que c'est le gagnant d'affirmation.

```
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 périodiquement une affirmation sur Tunnel0 (MDT par défaut), juste avant le temporisateur d'affirmation expire. Comme un tel PE2 reste le gagnant d'affirmation.

```
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'affirmation fonctionne également avec une interface de tunnel dans l'HUILE. Affirme sont permutés au-dessus du par défaut MDT quand les Routeurs de PE d'entrée reçoivent c (S, G) le trafic de multidiffusion sur l'interface de tunnel associée qui est dans l'HUILE.

Affirmez le mécanisme avec des données MDTs

Le plus souvent quand des données MDTs sont configurées, le mécanisme d'affirmation fonctionnera toujours sur le par défaut MDT comme c (S, G) le trafic est seulement commuté plus de du par défaut MDT aux données MDTs après trois secondes. Alors le même se produit comme décrit précédemment. Notez qu'il y a **seulement une interface de tunnel par VRF Multidiffusion-activé** : le par défaut MDT et tous données MDTs utilisent une interface de tunnel seulement. Cette interface de tunnel est utilisée dans l'HUILE sur les Routeurs de PE d'entrée ou comme interface RPF sur les Routeurs de PE de sortie.

Dans certains cas il est possible que le mécanisme d'affirmation ne soit pas déclenché avant que les données MDTs soient signalées. Alors il est possible que le c (S, G) des débuts du trafic de multidiffusion à expédier sur des données MDT sur Routeurs PE1 de PE d'entrée et PE2. En pareil cas, ceci pourrait mener au doublon c (S de constante, G) le trafic de multidiffusion à travers le réseau de noyau MPLS. Afin d'éviter ceci, cette solution a été mise en application : quand un routeur PE d'entrée voit un autre routeur PE d'entrée annoncer des données MDT pour lesquelles le routeur PE est également un routeur PE d'entrée, elles joignent ces données MDT. En principe, seulement les Routeurs de PE de sortie (qui ont un récepteur en aval) joindraient les données MDT. Puisque les Routeurs de PE d'entrée joignent les données MDT annoncées par d'autres Routeurs de PE d'entrée, il mène au routeur PE d'entrée recevant le trafic de multidiffusion de l'interface de tunnel qui est présente dans l'HUILE, et par conséquent ceci déclenche le mécanisme d'affirmation et mène à un des Routeurs de PE d'entrée pour cesser d'expédier le c (S, G) le trafic de multidiffusion sur ses données MDT (avec l'interface de tunnel), alors que l'autre PE d'entrée (le gagnant d'affirmation) peut continuer à expédier le c (S, G) le trafic de multidiffusion sur ses données MDT.

Pour l'exemple suivant, supposez que les Routeurs PE1 et PE2 de PE d'entrée n'ont jamais vu le c (S, G) le trafic de multidiffusion entre eux sur le par défaut MDT. Le trafic est sur le par défaut MDT pendant seulement trois secondes et il n'est pas difficile de comprendre que ceci peut se produire s'il y a, par exemple, perte provisoire du trafic sur le principal réseau.

La configuration pour les données MDT est ajoutée à tous les Routeurs de PE. La configuration sur tous les Routeurs de PE (le RD peut être différent sur les Routeurs de PE) est :

```
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 verront le trafic de la source, ils créent le courant alternatif - (S, G) entrée. Les deux Routeurs de PE d'entrée expédient le c (S, G) le trafic de multidiffusion sur le par défaut MDT. Les Routeurs PE3 et PE4 de PE de sortie reçoivent le trafic de multidiffusion et l'expédient. En raison d'une question provisoire, PE2 ne voit pas le trafic de PE1 et vice-versa sur le par défaut MDT. Ils que chacun des deux envoient des données MDT joignent la valeur de longueur de type (TLV) sur le par défaut MDT.

S'il n'y a aucun c (S, G) le trafic, vous voyez cet état de Multidiffusion sur les Routeurs de 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
```

Le y-indicateur n'est pas encore placé. Les deux Routeurs de PE d'entrée ont l'interface Tunnel0 dans l'HUILE. C'est dû au fait que PE3 a le RPF vers PE1 et PE4 a le RPF vers PE2 pour c (S,

G).

Quand le trafic de multidiffusion pour le c (S, G) des débuts à circuler, PE1 et PE2 expédient le trafic. Le seuil pour les données MDT est franchi sur des Routeurs de PE d'entrée et chacun des deux envoient des données MDT joignent la TLV et après l'expédition de début de trois secondes sur leurs données MDT. Notez que PE1 joint les données MDT originaires par PE2 et PE2 joint les données MDT originaires 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 le c (S, G) sur l'interface Tunnel0 (mais maintenant des données MDT, pas du par défaut MDT) et le mécanisme d'affirmation donne un coup de pied dedans. Seulement PE2 continue à expédier le c (S, G) le trafic sur ses données MDT :

```
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'HUILE.

```

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 a l'Un-indicateur réglé 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'affirmation fonctionne également quand des données MDTs sont utilisées.

Affirme sont permutés au-dessus du par défaut MDT quand les Routeurs de PE d'entrée reçoivent c (S, G) le trafic de multidiffusion sur l'interface de tunnel associée qui est dans l'HUILE.