

MDT de multidiffusion de nouvelle génération par défaut : Profil 0

Contenu

[Introduction](#)

[Informations générales](#)

[PIM en tant que signalisation de superposition](#)

[Tâches de configuration](#)

[Configuration](#)

[Vérification](#)

[Dépannage](#)

Introduction

Ce document décrit comment les paquets multidiffusion traversent avec l'utilisation du coeur MPLS (Multiprotocol Label Switching) dans la multidiffusion de nouvelle génération.

Informations générales

MDT par défaut - PIM C - signalisation mcast

Le projet Rosen utilise l'encapsulation de routage générique (GRE) comme protocole de superposition. Cela signifie que tous les paquets de multidiffusion sont encapsulés dans GRE. Un réseau local virtuel est émulé avec tous les routeurs de périphérie du fournisseur (PE) du VPN qui rejoignent un groupe de multidiffusion. Il s'agit de l'arbre de distribution multidiffusion (MDT) par défaut. Le MDT par défaut est utilisé pour la signalisation Protocol Independent Multicast (PIM) Hello et autres PIM, mais également pour le trafic de données. Si la source envoie beaucoup de trafic, il est inefficace d'utiliser le MDT par défaut et un MDT de données peut être créé. Les données MDT incluent uniquement les PE qui ont des récepteurs pour le groupe utilisé.

Le projet Rosen est assez simple à déployer et fonctionne bien, mais il présente quelques inconvénients. Examinons les points suivants :

Surcharge - GRE ajoute 24 octets de surcharge au paquet. Par rapport à MPLS qui ajoute généralement 8 ou 12 octets, 100 % ou plus de surcharge est ajouté à chaque paquet.

PIM dans le coeur - Draft Rosen nécessite que PIM soit activé dans le coeur, car les PE doivent joindre le MDT par défaut et/ou de données qui est effectué par la signalisation PIM. Si le module ASM PIM est utilisé dans le coeur, un RP est également nécessaire. Si PIM SSM est exécuté dans le coeur, aucun RP n'est nécessaire.

État de base : l'état inutile est créé dans le coeur en raison de la signalisation PIM provenant du PE. Le noyau doit avoir le moins d'état possible.

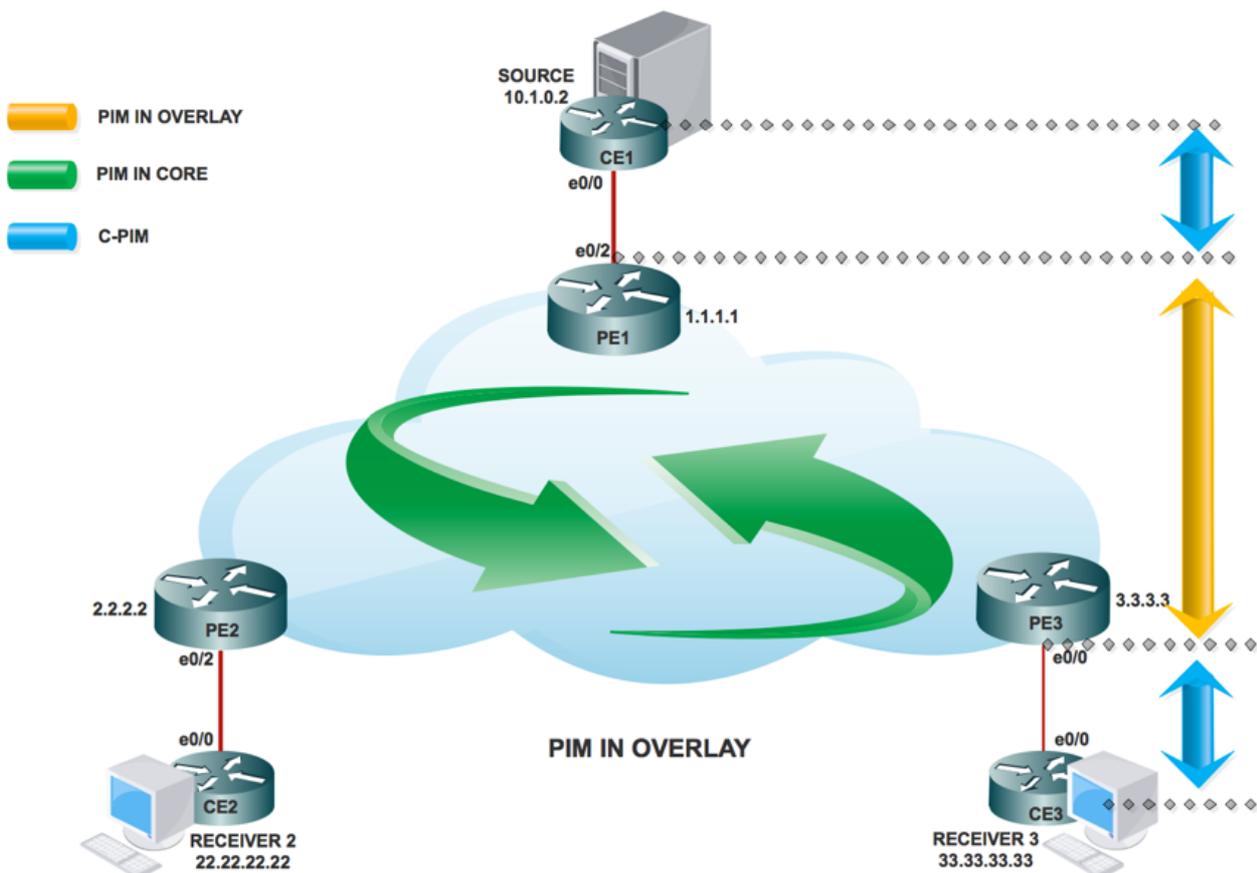
contiguïtés PIM - Les PE deviendront des voisins PIM les uns avec les autres. S'il s'agit d'un grand VPN et d'un grand nombre de PE, beaucoup de contiguïtés PIM seront créées. Cela génère beaucoup de signaux Hello et autres, ce qui ajoute à la charge du routeur.

Monodiffusion contre multidiffusion - Le transfert monodiffusion utilise MPLS, le multicast utilise GRE. Cela ajoute de la complexité et signifie que la monodiffusion utilise un mécanisme de transfert différent de la multidiffusion, ce qui n'est pas la solution optimale.

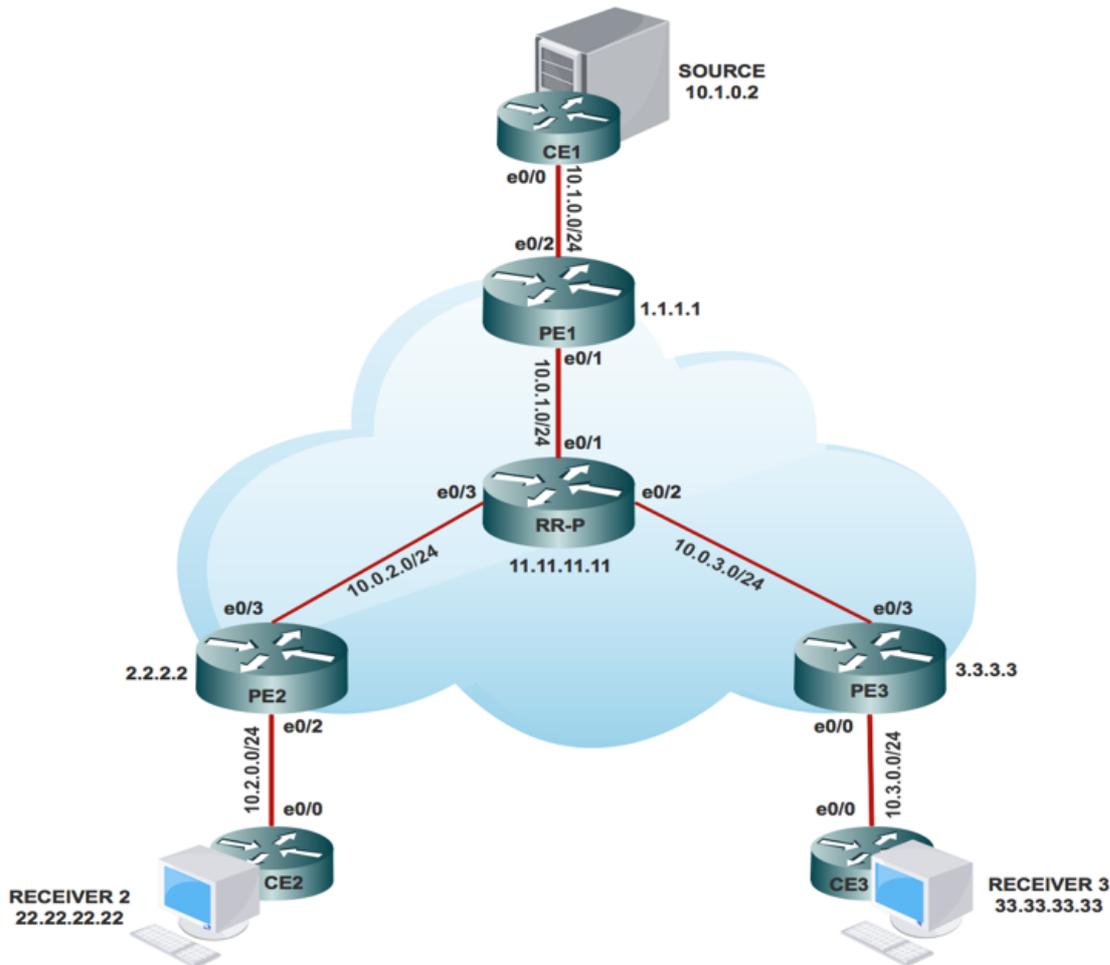
Inefficacité - Le MDT par défaut envoie le trafic à tous les PE du VPN, que le PE ait un récepteur dans le (*, G) ou (S, G) du groupe utilisé.

- Le MDT par défaut sera utilisé pour connecter la multidiffusion à tous les PE dans un VRF.
- Par défaut, il connecte tous les routeurs PE.
- Par défaut, il transporte tout le trafic.
- Tout le trafic de contrôle PIM et le trafic du plan de données. Par exemple, le trafic (*, G) et le trafic (S, G).
- Cela représente multipoint à multipoint.
- N'importe qui peut envoyer et tout le monde peut recevoir de l'arbre.

PIM en tant que signalisation de superposition



Topologie



Tâches de configuration

1. Activez le routage multidiffusion sur tous les noeuds.
2. Activez le mode PIM Sparse dans toutes les interfaces.
3. Avec le VRF existant, configurez le MDT par défaut.
4. Configurez le VRF sur l'interface Ethernet0/x.
5. Activez le routage multidiffusion sur VRF.
6. Configurez PIM SSM Default dans tous les noeuds du coeur de réseau.
7. Configurez le RP BSR dans le noeud CE.
8. Préconfiguré :
 - VRF m-GRE
 - mBGP : Famille d'adresses VPNv4
 - Protocole de routage VRF

Configuration

1. Activez le routage multidiffusion sur tous les noeuds.

```
(config)# ip multicast-routing
```

2. Activez le mode PIM Sparse dans toutes les interfaces.

```
(config)# interface ethernet0/x
```

```
(config-if)#ip pim sparse-mode
```

```
(config)# interface loopback0
```

```
(config-if)#ip pim sparse-mode
```

3. Avec VRF qui existe déjà, configurez le MDT par défaut.

```
(config)#ip vrf m-GRE
```

```
(config-vrf)# mdt default 232.1.1.1
```

4. Configurez le VRF sur l'interface Ethernet0/x.

Sur PE1, PE2 et PE3.

```
(config)# interface ethernet0/x
```

```
(config-if)# ip vrf forwarding m-GRE
```

```
(config-if)# ip address 10.x.0.1 255.255.255.0
```

5. Activez le routage multidiffusion sur VRF.

Sur PE1, PE2 et PE3.

```
(config)# ip multicast-routing vrf m-GRE
```

6. Configurez RP pour le coeur du fournisseur de services.

Sur les noeuds PE1, PE2, PE3 et RR-P.

```
(config)# ip pim rp-address 11.11.11.11
```

7. Configurez le RP BSR dans le noeud CE (récepteur).

Sur le récepteur 2.

```
(config)# ip pim bsr-candidate loopback0
```

```
(config)# ip pim rp-candidate loopback0
```

Vérification

Utilisez cette section pour confirmer que votre configuration fonctionne correctement.

Tâche 1 : Vérifier la connectivité physique

- Vérifiez que toutes les interfaces connectées sont " UP "

Tâche 2 : Vérification de la monodiffusion VPNv4 de la famille d'adresses

- Vérifiez que BGP est activé sur tous les routeurs pour la monodiffusion AF VPNv4 et que les voisins BGP sont “ UP ”
- Vérifiez que la table de monodiffusion VPNv4 BGP contient tous les préfixes du client.

Tâche 3 : Vérifier le trafic multidiffusion de bout en bout

- Vérifiez le voisinage PIM.
- Vérifiez que l'état de multidiffusion est créé de bout en bout.
- Vérifier l'entrée mRIB sur PE1, PE2 et PE3
- Vérifiez que l'entrée (S, G) mFIB, paquet étant incrémenté dans le transfert logiciel.
- Vérifier que les paquets ICMP atteignent l'interface CE à CE.

Verify all the connected interface are “UP”

```
#sh ip interface brief
```

Task 2: Verify Address Family VPNv4 unicast

Address Family VPNv4 unicast and BGP neighbors

```
# show running-config | s r bgp
# show bgp vpnv4 unicast summary all
```

VPNv4 unicast table has all the Customer prefixes

```
PE1#sh bgp vpnv4 unicast all

BGP table version is 31, local router ID is 1.1.1.1

   Network          Next Hop           Metric LocPrf Weight Path
Route Distinguisher: 100:100 (default for vrf m-GRE)

*>i 22.22.22.22/32   2.2.2.2             0   100     0 20 i
*>i 33.33.33.33/32   3.3.3.3             0   100     0 30 i
*>  111.111.111.111/32
                        10.1.0.2            0                   0 10 i
```

Check on all the PE nodes (PE1, PE2 and PE3)

Task 3: Verify Multicast Traffic end to end

Verify that multicast state is created end to end

```
#sh ip mroute vrf m-GRE 230.1.1.1 verbose
IP Multicast Routing Table
Flags: T - SPT-bit set, p - PIM Joins on route

(10.1.0.2, 230.1.1.1), 00:00:35/00:02:24, flags: Tp
Incoming interface: Ethernet0/2, RPF nbr 10.1.0.2
Outgoing interface list:
  Tunnel2, GRE MDT: 232.1.1.1 (default), Forward/Sparse,
  00:00:35/00:02:54, p
```

Check on all the PE nodes (PE1, PE2 and PE3)

Verify that (S,G) mFIB entry, packet getting incremented

```
# sh ip mfib vrf m-GRE 230.1.1.1 verbose
Entry Flags: NS - Negate Signalling, SP - Signal Present,
             A - Accept, F - Forward, RA - MRIB Accept, RF - MRIB Forward,
             MA - MFIB Accept
Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kbits per second
Other counts:      Total/RPF failed/Other drops
I/O Item Counts:  FS Pkt Count/PS Pkt Count
VRF m-GRE

(10.1.0.2,230.1.1.1) Flags: K DDE
SW Forwarding: 5/0/100/0, Other: 0/0/0
Ethernet0/2 Flags: RA A MA
Tunnel2, MDT/232.1.1.1 Flags: RF F NS
CEF: Adjacency with MAC: 4500000000000000FF2FD0CA01010101E801010100000800
Pkts: 4/1
```

Check PIM Neighborship at the core and vrf

```
PE1#sh ip pim 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
Neighbor      Interface      Uptime/Expires  Ver  DR
Address                               Prio/Mode
10.0.1.2      Ethernet0/1    01:34:51/00:01:25 v2   1 / DR S P G

PE1#sh ip pim vrf m-GRE 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
Neighbor      Interface      Uptime/Expires  Ver  DR
Address                               Prio/Mode
10.1.0.2      Ethernet0/2    01:34:32/00:01:42 v2   1 / DR S P G
3.3.3.3       Tunnel2        01:32:32/00:01:41 v2   1 / S P G
2.2.2.2       Tunnel2        01:32:32/00:01:36 v2   1 / S P G
```

Multicast Forwarding Packets

```
#sh ip mfib vrf m-GRE 230.1.1.1 count
Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/
Kilobits per second
Other counts:      Total/RPF failed/Other drops(OIF-null,
rate-limit etc)
VRF m-GRE
 11 routes, 7 (*,G)s, 2 (*,G/m)s
Group: 230.1.1.1
  RP-tree,
  SW Forwarding: 0/0/0/0, Other: 0/0/0
  Source: 10.1.0.2,
  SW Forwarding: 51/0/100/0, Other: 0/0/0
  Totals - Source count: 2, Packet count: 102

Groups: 1, 2.00 average sources per group
```

Verify ICMP packets getting reach from CE to CE

```
SOURCE1#ping 230.1.1.1 repeat 100 timeout 0

Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 230.1.1.1, timeout is 0
seconds:

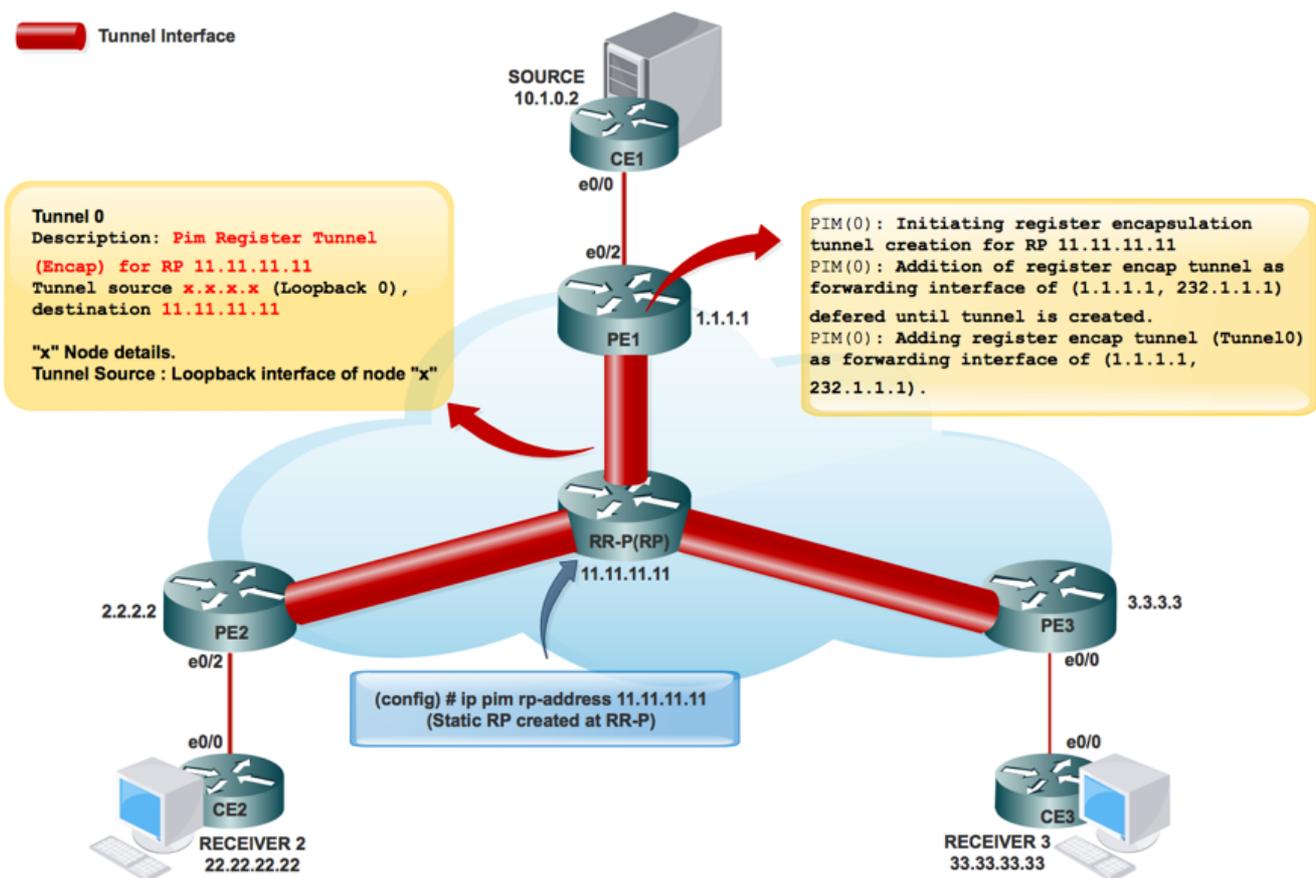
Reply to request 0 from 10.3.0.2, 4 ms
Reply to request 1 from 10.3.0.2, 4 ms
Reply to request 2 from 10.3.0.2, 4 ms
Reply to request 3 from 10.3.0.2, 4 ms
```

Lors de la création des interfaces de tunnel :

Création RP du fournisseur de services :

Une fois que les informations RP sont inondées dans le coeur. Le tunnel d'interface 0 est créé.

 Tunnel Interface



PIM(0) : Initialisation de la création du tunnel d'encapsulation de registre pour RP 11.11.11.11.

PIM(0) : La création initiale du tunnel de registre a réussi pour RP 11.11.11.11.

PIM(0) : Ajout du tunnel d'encapsulation de registre en tant qu'interface de transfert de (1.1.1.1, 232.1.1.1) différée jusqu'à la création du tunnel.

9 mai 17:34:56.155 : PIM(0) : Cochez RP 11.11.11.11 dans le (, 232.1.1.1).

PIM(0) : Ajout du tunnel d'encapsulation de registre (Tunnel0) en tant qu'interface de transfert de (1.1.1.1, 232.1.1.1).

```
PE1#sh int tunnel 0
```

```
Tunnel0 is up, line protocol is up
```

```
Hardware is Tunnel
```

```
Description: Pim Register Tunnel (Encap) for RP 11.11.11.11
```

```
Interface is unnumbered. Using address of Ethernet0/1 (10.0.1.1)
```

```
MTU 17912 bytes, BW 100 Kbit/sec, DLY 50000 usec,
```

```
reliability 255/255, txload 1/255, rxload 1/255
```

```
Encapsulation TUNNEL, loopback not set
```

```
Keepalive not set
```

Tunnel source 10.0.1.1 (Ethernet0/1), destination 11.11.11.11 >>>>>>>>> Tunnel Source and destination

Tunnel Subblocks:

src-track:

Tunnel0 source tracking subblock associated with Ethernet0/1

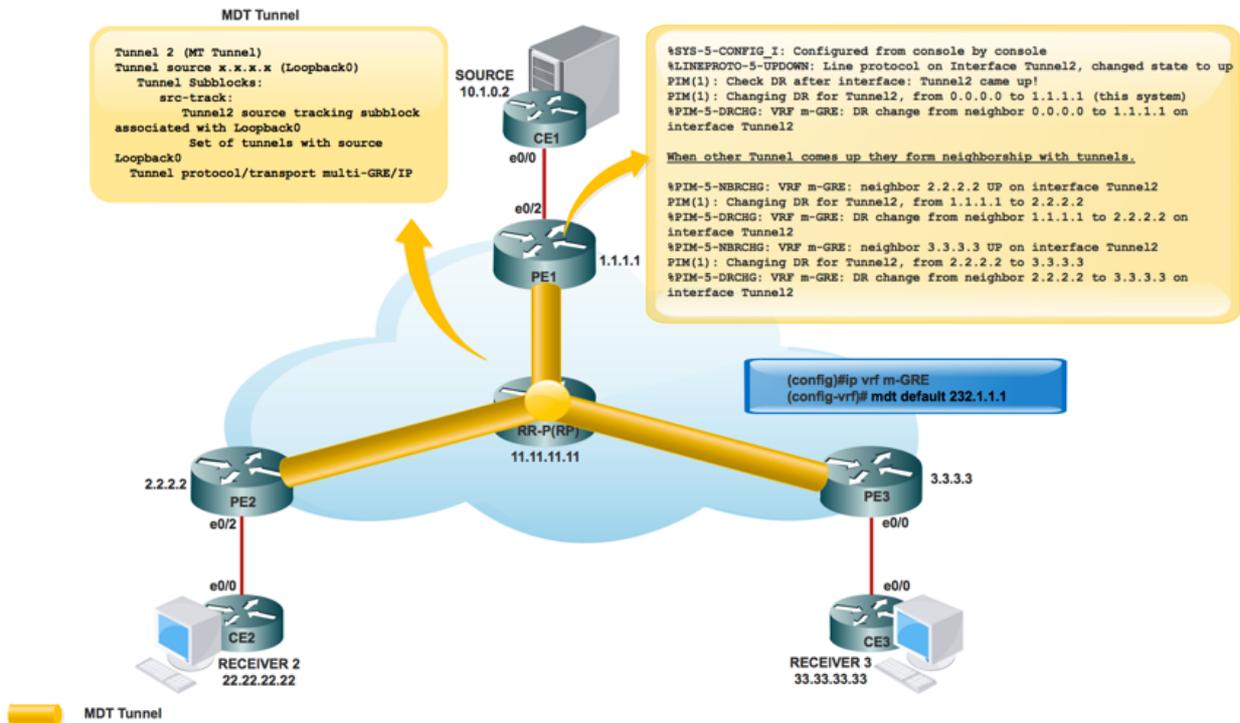
Set of tunnels with source Ethernet0/1, 1 member (includes iterators), on interface <OK>

Tunnel protocol/transport PIM/IPv4

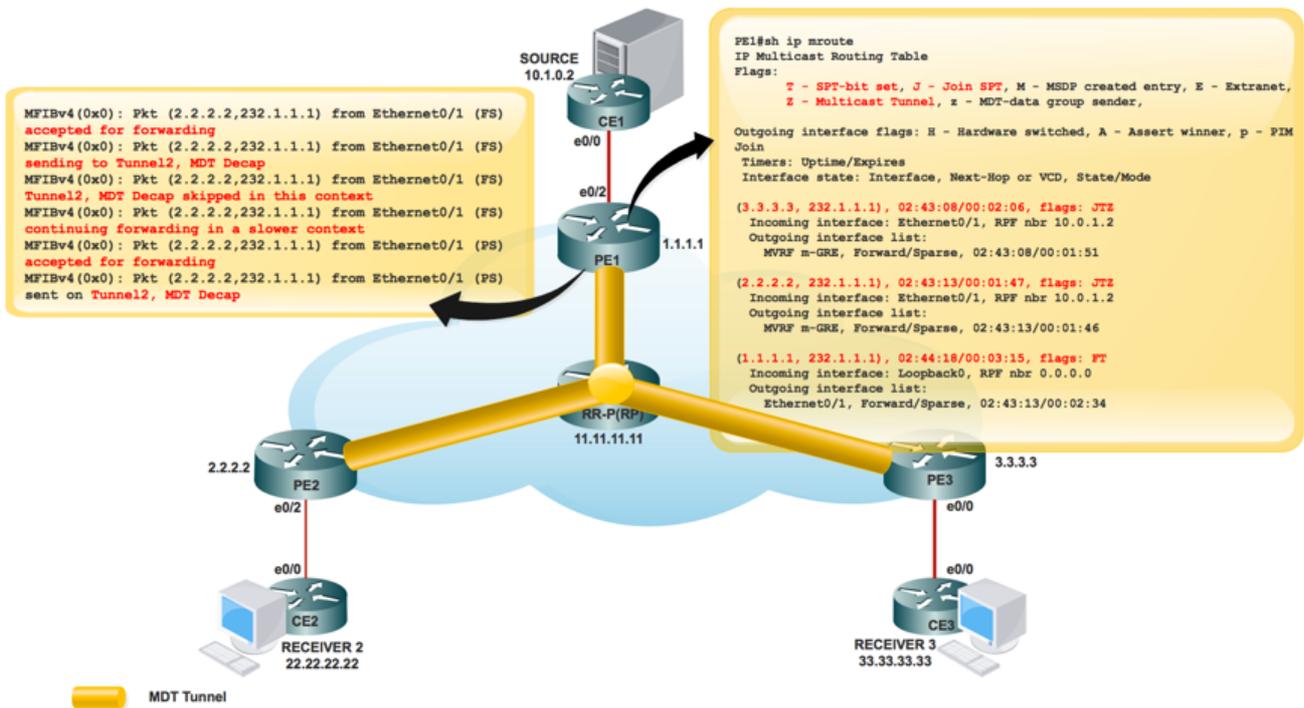
Tunnel TOS/Traffic Class 0xC0, Tunnel TTL 255

Tunnel transport MTU 1472 bytes

Création du tunnel MDT :



Création de MRIB dans le coeur :



PE1#sh ip mroute

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,

(3.3.3.3, 232.1.1.1), 00:10:13/00:01:01, flags: **JTZ**

Incoming interface: Ethernet0/1, RPF nbr 10.0.1.2

Outgoing interface list:

MVRFB m-GRE, Forward/Sparse, 00:10:13/00:01:46

(2.2.2.2, 232.1.1.1), 00:10:14/00:00:57, flags: **JTZ**

Incoming interface: Ethernet0/1, RPF nbr 10.0.1.2

Outgoing interface list:

MVRFB m-GRE, Forward/Sparse, 00:10:14/00:01:45

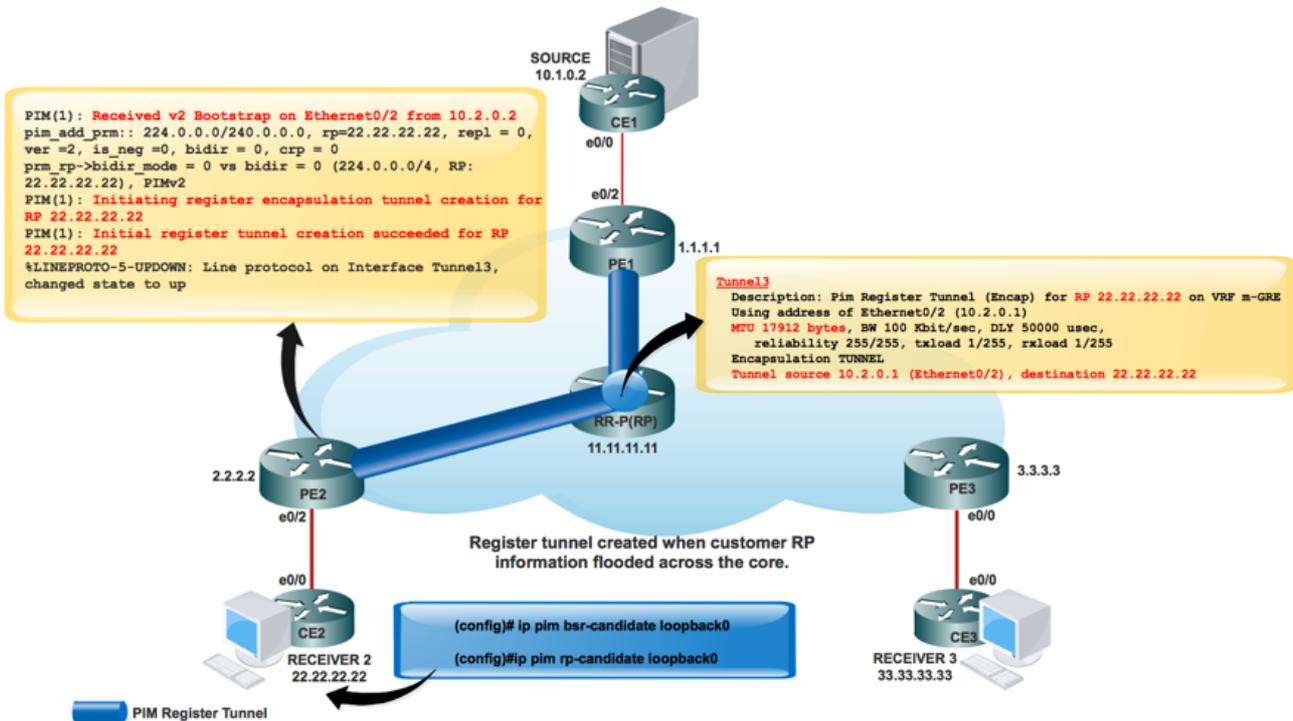
(1.1.1.1, 232.1.1.1), 00:10:15/00:03:20, flags: FT

Incoming interface: Loopback0, RPF nbr 0.0.0.0

Outgoing interface list:

Ethernet0/1, Forward/Sparse, 00:10:15/00:03:04

Une fois le RP créé pour le réseau du client :



*May 9 18:54:42.170: prm_rp->bidir_mode = 0 vs bidir = 0 (224.0.0.0/4, RP:33.33.33.33), PIMv2

*May 9 18:54:42.170: PIM(1): Initiating register encapsulation tunnel creation for RP 33.33.33.33

*May 9 18:54:42.170: PIM(1): Initial register tunnel creation succeeded for RP 33.33.33.33

*May 9 18:54:43.173: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel2, changed state to up

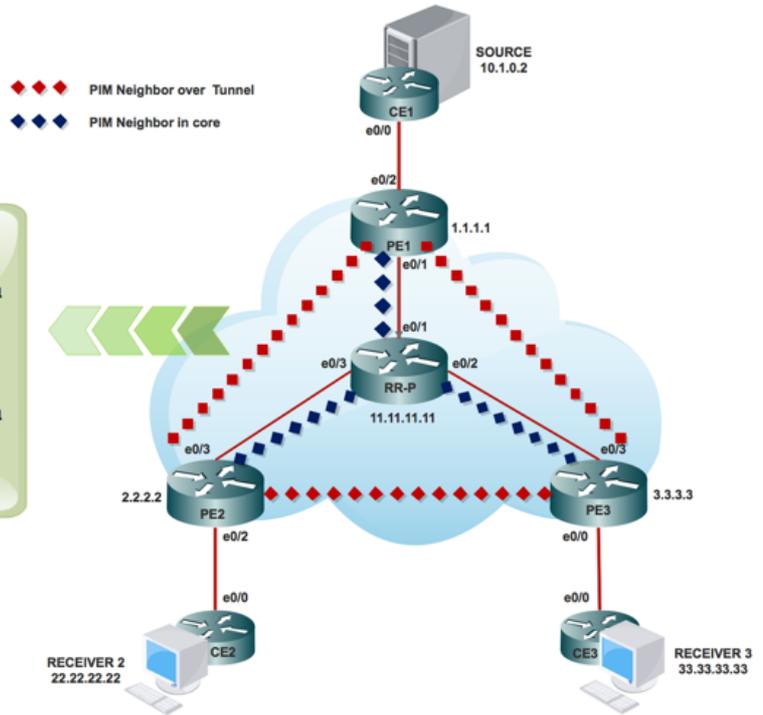
Une interface de tunnel est créée afin de transporter les informations RP du client.

PIM(1) : Initialisation de la création du tunnel d'encapsulation de registre pour RP 22.22.22.22.

Il s'agit du tunnel créé pour effectuer l'encapsulation Register au RP.

Pour chaque RP en mode intermédiaire découvert, un tunnel d'encapsulation Register est créé. Sur le RP en mode clairsemé lui-même, une interface de tunnel de désencapsulation a été créée pour recevoir les paquets Register.

Voisinage PIM :



Control Plane Scalability:

For Example:

- => PE anticipating 100 MVPN services which distributed across 100 PEs.
- => Each PE maintains 9900 (99x100) PIM adjacencies in addition to the adjacency.
- => In order to preserve 9900 PIM adjacencies, the PE would be sending approx 330 PIM adjacencies per second (Using default 30s PIM hello timer)
- => The number will get worse as the number of MVPN services or PEs increases.

```
PE1#sh ip pim interface
```

Address	Interface	Ver/	Nbr	Query	DR	DR
		Mode	Count	Intvl	Prior	
1.1.1.1	Loopback0	v2/S	0	30	1	1.1.1.1
10.0.1.1	Ethernet0/1	v2/S	1	30	1	10.0.1.2

```
PE1#sh ip pim vrf m-GRE 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

Neighbor	Interface	Uptime/Expires	Ver	DR
Address				Prio/Mode
10.1.0.2	Ethernet0/2	03:08:34/00:01:43	v2	1 / DR S P G
3.3.3.3	Tunnell	01:44:24/00:01:41	v2	1 / DR S P G
2.2.2.2	Tunnell	01:44:24/00:01:38	v2	1 / S P G

Flux de paquets :

Le flux de paquets du plan de contrôle se divise en deux parties.

1. Le récepteur est en ligne.

2. La source est active.

Lorsque le récepteur est actif :

```

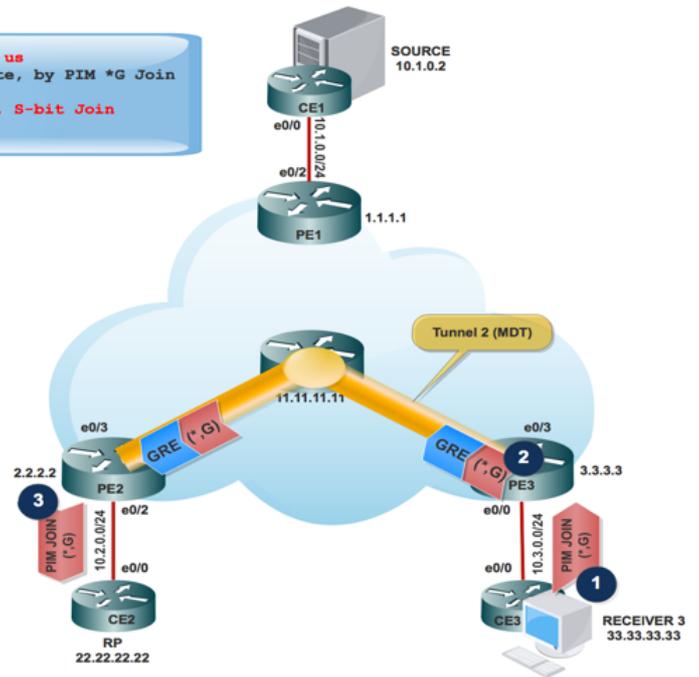
1
PIM(1): Received v2 Join/Prune on Ethernet0/0 from 10.3.0.2, to us
PIM(1): Add Ethernet0/0/10.3.0.2 to (*, 224.1.1.1), Forward state, by PIM *G Join
PIM(1): Upstream mode for (*, 224.1.1.1) changed from 0 to 1
PIM(1): Adding v2 (22.22.22.22/32, 224.1.1.1), WC-bit, RPT-bit, S-bit Join
PIM(1): Send v2 join/prune to 2.2.2.2 (Tunnel2)
    
```

```

2
PE3#sh ip mroute vrf m-GRE
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group,
       C - Connected, J - Join SPT,
(*, 224.1.1.1), 00:00:22/00:02:46, RP 22.22.22.22, flags: SJC
Incoming interface: Tunnel2, RPF nbr 2.2.2.2
Outgoing interface list:
Ethernet0/0, Forward/Sparse, 00:00:22/00:03:07
    
```

```

3
PE2# sh ip mroute vrf m-GRE
(*, 224.1.1.1), 01:03:26/00:02:57, RP 22.22.22.22, flags: Sp
Incoming interface: Ethernet0/2, RPF nbr 10.2.0.2
Outgoing interface list:
Tunnel2, GRE MDT: 232.1.1.1 (default), Forward/Sparse,
01:03:26/00:02:57, p
    
```



1. Le récepteur est en ligne, envoie PIM JOIN (*, G) vers PE3.

2. PE3 encapsule le PIM JOIN (*, G) dans le paquet GRE et envoie par le tunnel 2 (tunnel MDT), qui est vérifié à partir de l'interface entrante de **show ip mroute vrf m-GRE**.

```

42 26.584402 3.3.3.3 224.0.0.13 PIMv2 92 Join/Prune
▶ Frame 42: 92 bytes on wire (736 bits), 92 bytes captured (736 bits) on interface 0
▶ Ethernet II, Src: aa:bb:cc:00:50:30 (aa:bb:cc:00:50:30), Dst: IPv4mcast_01:01:01 (01:00:5e:01:01:01)
▶ Internet Protocol Version 4, Src: 3.3.3.3, Dst: 232.1.1.1
▶ Generic Routing Encapsulation (IP)
▶ Internet Protocol Version 4, Src: 3.3.3.3, Dst: 224.0.0.13
▼ Protocol Independent Multicast
  0010 .... = Version: 2
  .... 0011 = Type: Join/Prune (3)
  Reserved byte(s): 00
  Checksum: 0xc0b8 [correct]
  ▼ PIM Options
    Upstream-neighbor: 2.2.2.2
    Reserved byte(s): 00
    Num Groups: 1
    Holdtime: 210
  ▼ Group 0: 224.1.1.1/32
    ▼ Num Joins: 1
      IP address: 22.22.22.22/32 (SWR)
    Num Prunes: 0
    
```

PE3#sh ip mroute

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

```
(3.3.3.3, 232.1.1.1), 10:20:04/00:02:56, flags: FT
```

```
Incoming interface: Loopback0, RPF nbr 0.0.0.0
```

```
Outgoing interface list:
```

```
Ethernet0/3, Forward/Sparse, 10:20:04/00:02:40
```

1. PE2 a reçu le paquet GRE dont la source est 3.3.3.3 et la destination 232.1.1.1 et le transfère à MVRF m-GRE en fonction de l'OIL.

```
PE2#sh ip mroute
```

```
IP Multicast Routing Table
```

```
Flags:
```

```
T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
```

```
Z - Multicast Tunnel, z - MDT-data group sender,
```

```
(3.3.3.3, 232.1.1.1), 11:47:30/00:01:01, flags: JTZ
```

```
Incoming interface: Ethernet0/3, RPF nbr 10.0.2.2
```

```
Outgoing interface list:
```

```
MVRF m-GRE, Forward/Sparse, 11:47:30/00:00:29
```

Le paquet GRE est décapsulé et PIM JOIN envoie vers le RP.

Note: Le voisin RPF est 2.2.2.2 car la jointure PIM est destinée à l'adresse RP pour former le RPT via le coeur de réseau.

Note: Bit WC et Bit RPT : déclenché par l'état (*, G), le DR crée un message Join/Prune avec l'adresse RP dans sa liste de jointure et le bit générique (bit WC) et le bit RP-tree (bit RPT) définis sur 1. Le bit WC indique que toute source peut correspondre et être transmise conformément à cette rubrique s'il n'y a plus de correspondance ; le bit RPT indique que cette jointure est envoyée vers le haut de l'arborescence RP partagée. La liste des pruneaux reste vide. Lorsque le bit RPT est défini sur 1, il indique que la jointure est associée à l'arborescence RP partagée et que, par conséquent, le message Join/Prune est propagé le long de l'arborescence RP. Lorsque le bit WC est défini sur 1, il indique que l'adresse est un RP et que les récepteurs en aval s'attendent à recevoir des paquets de toutes les sources via ce chemin (arborescence partagée).

```
PE2#sh ip mroute verbose
```

```
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 -

V - RD & Vector, v - Vector, p - PIM Joins on route

(2.2.2.2, 232.1.1.1), 22:48:12/00:02:04, flags: FTp

Incoming interface: Loopback0, RPF nbr 0.0.0.0

Outgoing interface list:Ethernet0/3, Forward/Sparse, 22:48:12/00:03:12, p

1. GRE a encapsulé la portée du paquet au niveau de la source PE1.

PE1#sh ip mroute verbose

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,

(2.2.2.2, 232.1.1.1), 22:55:50/00:02:45, flags: JTZ

Incoming interface: Ethernet0/1, RPF nbr 10.0.1.2

Outgoing interface list:MVRF m-GRE, Forward/Sparse, 22:55:50/00:01:09

PIM(1): Received v2 Join/Prune on Tunnel2 from 2.2.2.2, to us

PIM(1): Join-list: (10.1.0.2/32, 224.1.1.1), S-bit set

2. PIM JOIN (S, G) atteint à la source CE.

3. La source a maintenant obtenu les informations du récepteur intéressé et le trafic commence à être envoyé à la source PE1.

4. À la source PE1 :

PIM(1): Add Tunnel2/2.2.2.2 to (10.1.0.2, 224.1.1.1), Forward state, by PIM SG Join

MFIBv4(0x1): Pkt (10.1.0.2,224.1.1.1) from Ethernet0/2 (PS) accepted for forwarding

MFIBv4(0x1): Pkt (10.1.0.2,224.1.1.1) from Ethernet0/2 (PS) sending to Tunnel2, MDT/232.1.1.1

MFIBv4(0x1): Pkt (10.1.0.2,224.1.1.1) from Ethernet0/2 (PS) sent on Tunnel2, MDT/232.1.1.1

À PE2 (RP PE) :

```
PIM(1): Prune-list: (10.1.0.2/32, 224.1.1.1) RPT-bit set
```

```
PIM(1): Cancel sending Join for (10.1.0.2/32, 224.1.1.1) on Tunnel2
```

```
PE2#sh ip mroute vrf m-GRE
```

```
IP Multicast Routing Table
```

```
Flags: L - Local, P - Pruned, R - RP-bit set, F - Register flag,
```

```
(10.1.0.2, 224.1.1.1), 00:03:52/00:01:29, flags: R
```

```
Incoming interface: Ethernet0/2, RPF nbr 10.2.0.2
```

```
Outgoing interface list:
```

```
Tunnel2, Forward/Sparse, 00:00:52/00:02:58
```

Capture PCAP du paquet de multidiffusion à partir de PE1. Tunnel dans le tunnel par défaut MDT. Encapsulé avec GRE.

5. Au niveau du récepteur PE3, le paquet est reçu.

```
PE3#sh ip mroute verbose
```

```
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,
```

```
Z - Multicast Tunnel, z - MDT-data group sender,
```

```
(1.1.1.1, 232.1.1.1), 23:12:51/00:02:50, flags: JTZ
```

```
Incoming interface: Ethernet0/3, RPF nbr 10.0.3.2
```

```
Outgoing interface list:
```

```
MVRF m-GRE, Forward/Sparse, 23:12:51/stopped
```

```
PIM(1): Building Join/Prune packet for nbr 2.2.2.2
```

```
PIM(1): Adding v2 (10.1.0.2/32, 224.1.1.1), RPT-bit, S-bit Prune
```

```
PIM(1): Send v2 join/prune to 2.2.2.2 (Tunnel2)
```

```
PIM(1): Building Join/Prune packet for nbr 1.1.1.1
```

```
MFIBv4(0x1): Pkt (10.1.0.2,224.1.1.1) from Tunnel2, MDT/232.1.1.1 (PS) accepted for forwarding
```

```
MFIBv4(0x1): Pkt (10.1.0.2,224.1.1.1) from Tunnel2, MDT/232.1.1.1 (PS) sent on Ethernet0/0
MFIBv4(0x1): Pkt (10.1.0.2,224.1.1.1) from Tunnel2, MDT/232.1.1.1 (PS) accepted for forwarding
MFIBv4(0x1): Pkt (10.1.0.2,224.1.1.1) from Tunnel2, MDT/232.1.1.1 (PS) sent on Ethernet0/0
*Jun  2 20:09:11.817: PIM(1): Received v2 Join/Prune on Ethernet0/0 from 10.3.0.2, to us
```

```
PE3#sh ip mroute vrf m-GRE verbose
```

```
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,
```

```
      V - RD & Vector, v - Vector, p - PIM Joins on route
```

```
(10.1.0.2, 224.1.1.1), 00:00:07/00:02:52, flags: Tp
```

```
Incoming interface: Tunnel2, RPF nbr 1.1.1.1
```

```
Outgoing interface list:
```

```
  Ethernet0/0, Forward/Sparse, 00:00:07/00:03:22, p
```

```
RPF Change at PE3 (Receiver PE)
```

```
MRT(1): (10.1.0.2,224.1.1.1), RPF change from /2.2.2.2 to Tunnel1/1.1.1.1
```

```
MRT(1): Create (10.1.0.2 ,224.1.1.1), RPF (Tunnel2, 1.1.1.1, 200/0)
```

```
MRT(1): Set the T-flag for (10.1.0.2, 224.1.1.1)
```

```
MRT(1): WAVL Insert interface: Tunnel1 in (10.1.0.2,224.1.1.1) Successful
```

```
MRT(1): set min mtu for (10.1.0.2, 224.1.1.1) 18010->1500
```

Remarque : le voisin RPF est modifié une fois qu'un paquet multidiffusion de PE1 est reçu. Auparavant, il s'agissait de PE2 comme RP hébergé derrière lui. Après avoir reçu le premier paquet de multidiffusion, il modifie le RPF et définit le bit SPT.

Flux de trafic sur le tunnel MDT par défaut :

- Le transfert sur le MDT utilise GRE, le paquet C devient un paquet P.
- Adresse P-Packet S = adresse d'appairage BGP du PE
 Adresse G = adresse du groupe MDT (par défaut ou données)
- C-Packet IP TOS sera copié dans P-Packet.
- Les étiquettes MPLS NE sont PAS utilisées dans le coeur, uniquement la multidiffusion native.

Flux de paquets :

1. Un paquet C arrive sur une interface PE configurée VRF, mVRF est implicitement identifié.

Contrôle RPF normal sur la source C.

L'interface C-Packet a été répliquée dans l'OIL. À ce stade, il s'agit de l'interface PE dans le même VRF.

```
PE1#sh ip mroute vrf m-GRE verbose
```

```
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, v - Vector, p - PIM Joins on route
```

```
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
```

```
Timers: Uptime/Expires
```

```
Interface state: Interface, Next-Hop or VCD, State/Mode
```

```
(10.1.0.2, 224.1.1.1), 00:00:03/00:02:56, flags: Tp
```

```
Incoming interface: Ethernet0/2, RPF nbr 10.1.0.2
```

```
Outgoing interface list:
```

```
  Tunnel2, GRE MDT: 232.1.1.1 (default), Forward/Sparse, 00:00:03/00:03:26, p (Small "p"  
indicates downstream PIM join)
```

Si l'OIL contient une MTI, le paquet C est encapsulé dans un paquet P. Si l'indicateur "y" est défini sur la destination d'entrée utilisée est le groupe DATA-MDT sinon le groupe MDT par défaut. La source est l'adresse de l'homologue BGP PE et la destination est l'adresse du groupe MDT.

2. Le paquet P est transmis via le réseau P conformément à la multidiffusion normale.

Le paquet arrive à l'interface globale. Entrée globale (S, G) ou (*, G) pour le groupe MDT référencé. Contrôle RPF normal sur P-Source (homologue PE).

3. Le paquet P est répliqué en dehors de l'interface OIL. À ce stade, il s'agit de P/PE dans la table de routage globale.

4. Si l'indicateur "Z" définit le paquet est décapsulé pour révéler le paquet C. L'interface mVRF cible et l'interface entrante dérivée du groupe MDT sont la destination de l'en-tête encapsulé.

Vérification RPF du paquet C dans mVRF effectuée, C Packet a répliqué l'OIL dans mVRF.

```
PE3#sh ip mroute verbose
```

```
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,

Z - Multicast Tunnel, **z** - MDT-data group sender,

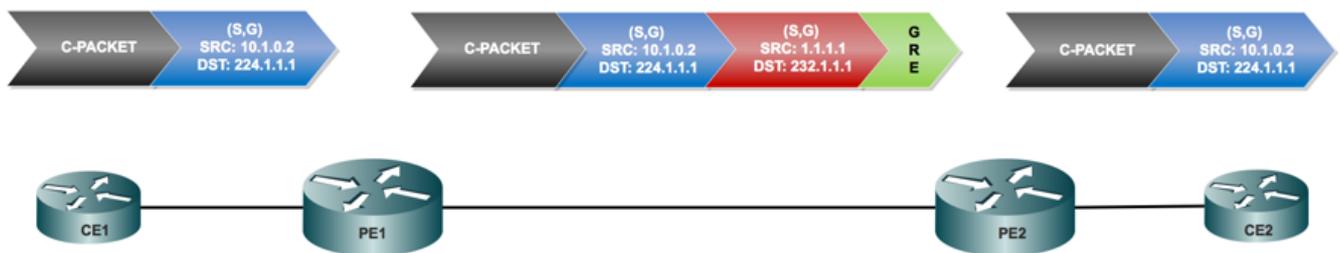
(1.1.1.1, 232.1.1.1), 1d01h/00:02:47, flags: **JTZ**

Incoming interface: Ethernet0/3, RPF nbr 10.0.3.2

Outgoing interface list: **MVRF m-GRE**, Forward/Sparse, 1d01h/stopped

5. Paquet C natif au niveau du récepteur 3.

Encapsulation de paquets :

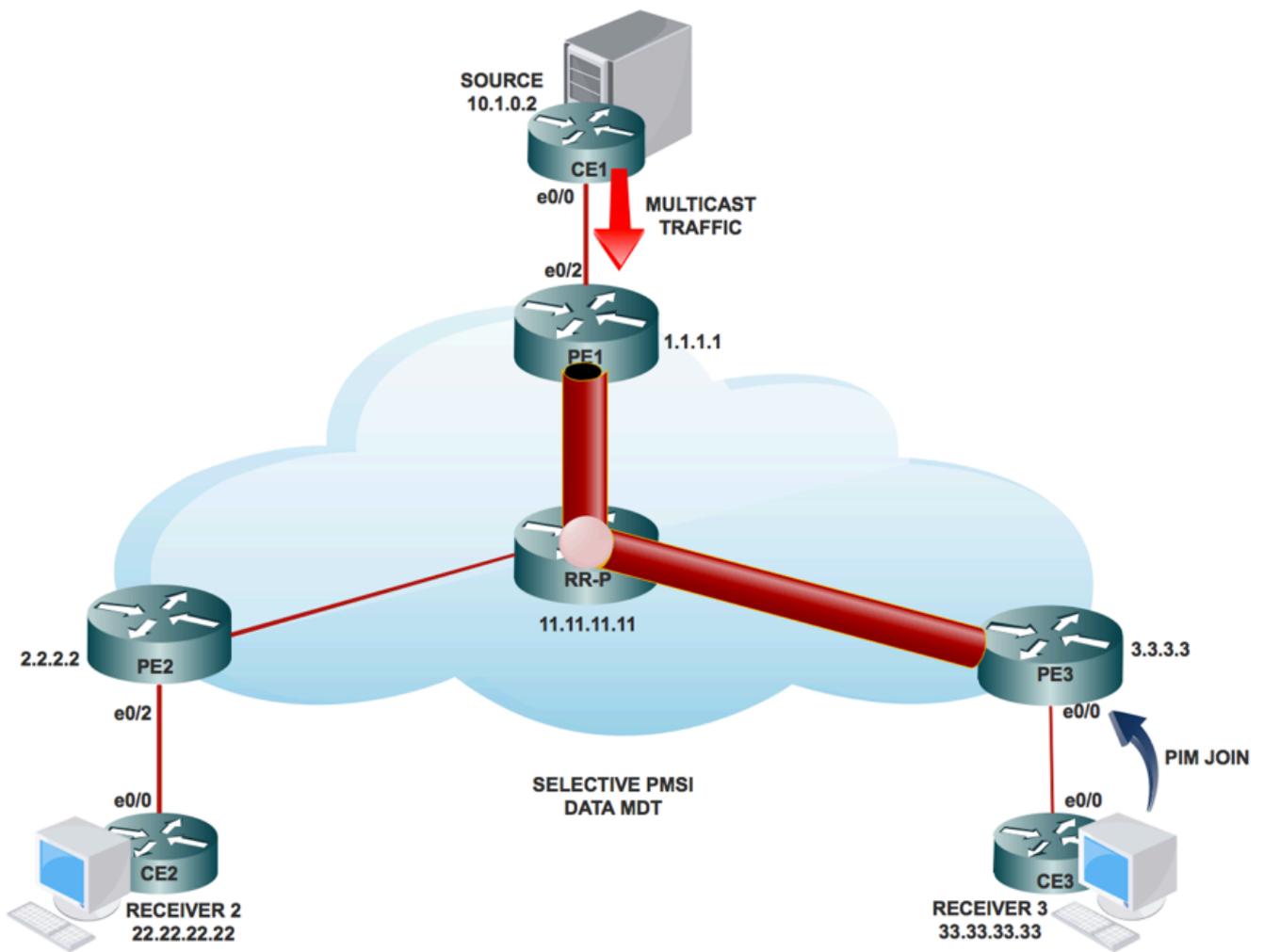


MDT de données :

Qu'est-ce que Data MDT ?

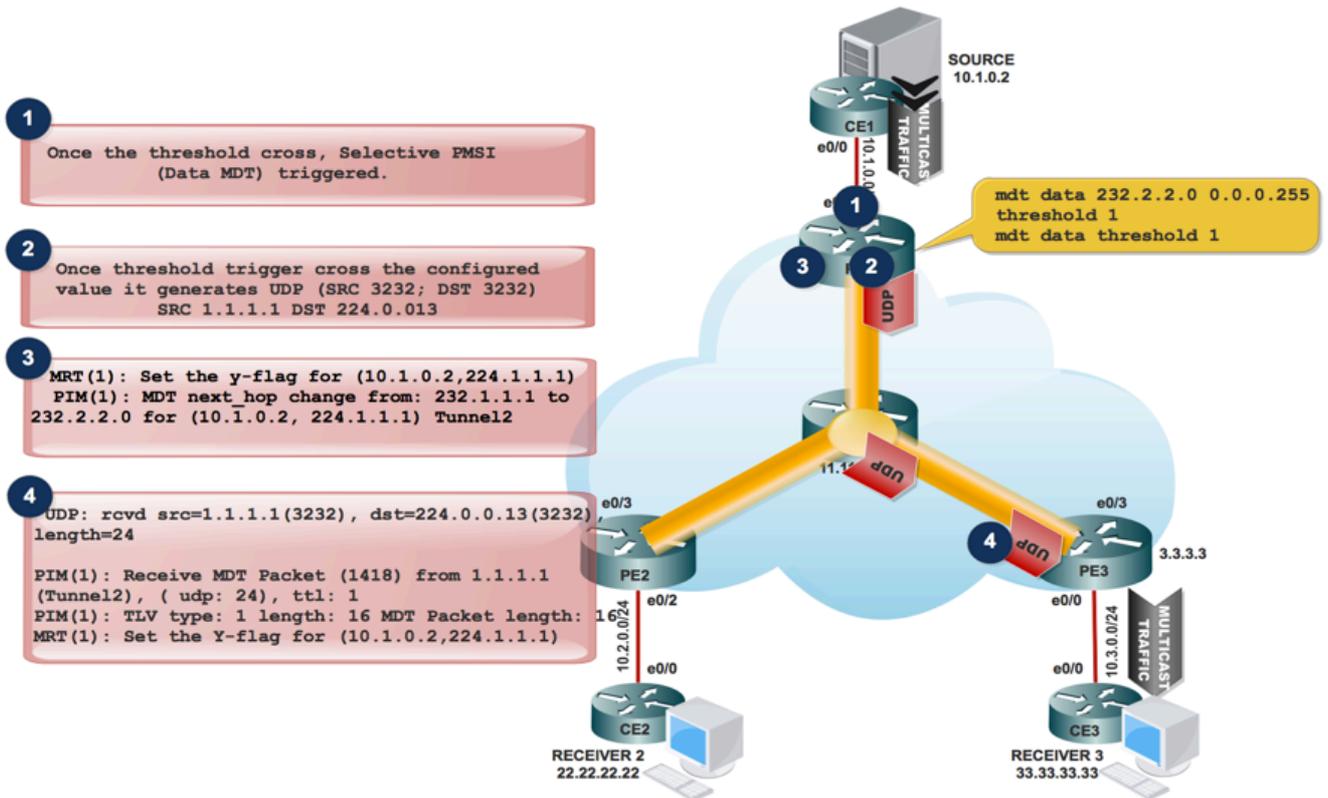
Il est facultatif. Il est créé en demande, il transporte un trafic spécifique (S, G). Dans la dernière version IOS®, le seuil configuré est " 0 " et " " infini. À chaque fois qu'un premier paquet atteint le VRF, le MDT de données s'initialise, et si l'infini, le MDT de données ne sera jamais créé, et le trafic se déplace vers l'avant dans le MDT par défaut. Le MDT de données est toujours l'arbre de réception, il n'envoie jamais de trafic. Le MDT de données est uniquement destiné au trafic (S, G).

PMSI sélectif :



- Elle est facultative. Il est créé en demande, il transporte un trafic spécifique (S, G).
- À chaque fois qu'un premier paquet atteint le VRF, le MDT de données s'initialise, et si l'infini, le MDT de données n'est jamais créé, et le trafic se déplace vers l'avant dans le MDT par défaut.
- Le MDT de données est toujours l'arbre de réception, il n'envoie jamais de trafic. Le MDT de données est uniquement destiné au trafic (S, G).
- Le message PIM porte les groupes C- (S, G) et P.

Création de DATA MDT :



1. Lorsque le trafic de multidiffusion entre dans le VRF et lorsque le débit de trafic atteint le seuil. Il génère un paquet MDT.

2. Le paquet MDT est encapsulé dans UDP avec Source et Destination 3232. Et l'envoyer au destinataire intéressé.

252	23.108432	1.1.1.1	224.0.0.13	UDP	82	3232 → 3232	Len=16
<ul style="list-style-type: none"> Frame 252: 82 bytes on wire (656 bits), 82 bytes captured (656 bits) on interface 0 Ethernet II, Src: aa:bb:cc:00:30:20 (aa:bb:cc:00:30:20), Dst: IPv4mcast_01:01:01 (01:00:5e:01:01:01) Internet Protocol Version 4, Src: 1.1.1.1, Dst: 232.1.1.1 Generic Routing Encapsulation (IP) Internet Protocol Version 4, Src: 1.1.1.1, Dst: 224.0.0.13 User Datagram Protocol, Src Port: 3232 (3232), Dst Port: 3232 (3232) Data (16 bytes) 							

3. Après avoir envoyé le paquet UDP au destinataire intéressé, il définit "indicateur y" et modifie le saut suivant MDT en nouvelle adresse de groupe MDT.

À la source PE1 :

```
MRT(1): Set the y-flag for (10.1.0.2,224.1.1.1)
```

```
PIM(1): MDT next_hop change from: 232.1.1.1 to 232.2.2.0 for (10.1.0.2, 224.1.1.1) Tunnel2
```

```
PE1#sh ip mroute vrf m-GRE verbose
```

```
IP Multicast Routing Table
```

```
Flags:
```

T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
Y - Joined MDT-data group, y - Sending to MDT-data group,
p - PIM Joins on route

(10.1.0.2, 224.1.1.1), 00:08:09/00:02:46, flags: T~~y~~p

Incoming interface: Ethernet0/2, RPF nbr 10.1.0.2

Outgoing interface list:

Tunnel2, GRE MDT: 232.2.2.0 (data), Forward/Sparse, 00:08:09/00:03:27, A, p (Small "p"
indicates downstream PIM join)

Note: Le tronçon suivant de l'OIL passe à 232.2.2.0.

2. Sur PE3, lorsqu'il reçoit le paquet MDT encapsulé dans le port UDP SRC 3232 et le port DST 3232.

UDP: rcvd src=1.1.1.1(3232), dst=224.0.0.13(3232), length=24

PIM(1): Receive MDT Packet (1418) from 1.1.1.1 (Tunnel2), length (ip: 44, udp: 24), ttl: 1

PIM(1): TLV type: 1 length: 16 MDT Packet length: 16

MRT(1): Set the Y-flag for (10.1.0.2,224.1.1.1)

PE3#sh ip mroute vrf m-GRE verbose

IP Multicast Routing Table

Flags:

T - SPT-bit set, Y - Joined MDT-data group, y - Sending to MDT-data

p - PIM Joins on route

(10.1.0.2, 224.1.1.1), 00:08:27/00:00:20, flags: T~~Y~~p

Incoming interface: Tunnel1, RPF nbr 1.1.1.1, MDT:232.2.2.0/00:02:15

Outgoing interface list:

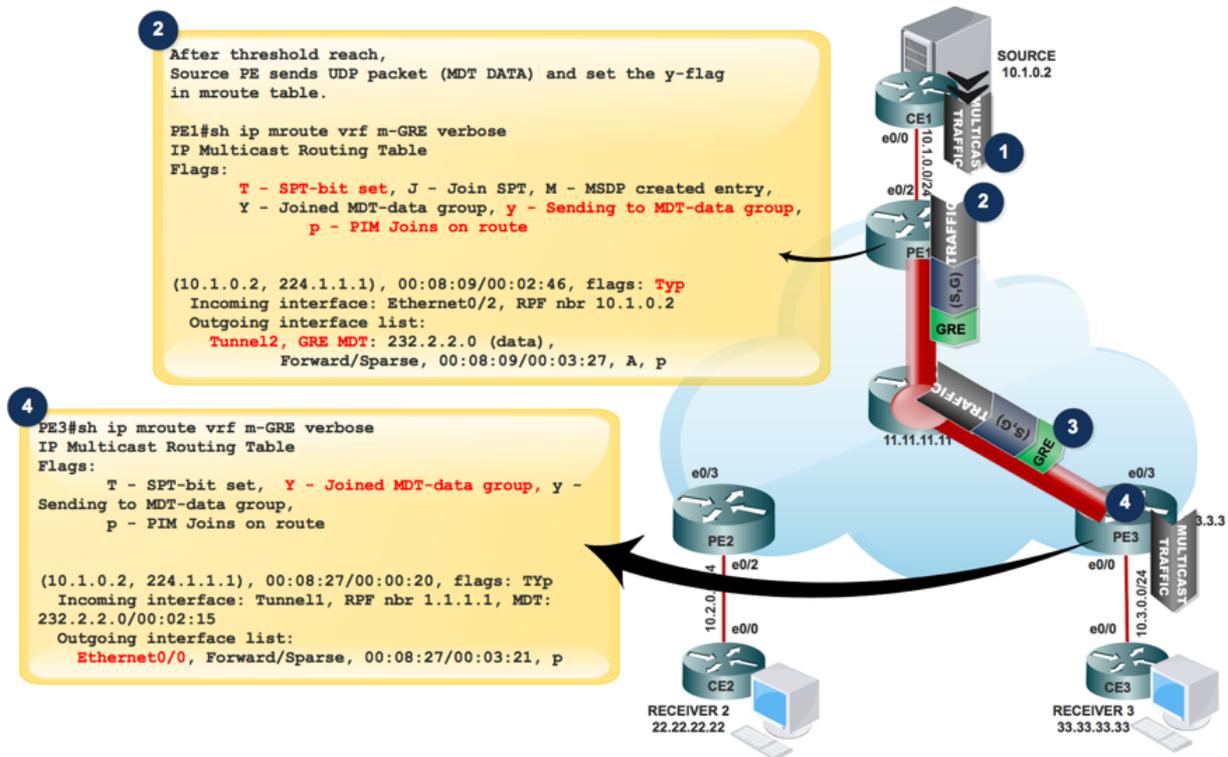
Ethernet0/0, Forward/Sparse, 00:08:27/00:03:21, p

Le message S-PMSI Join est un message encapsulé UDP dont l'adresse de destination est ALL-PIM-ROUTERS (224.0.0.13) et dont le port de destination est 3232.

Le message S-PMSI Join contient ces informations : un identificateur pour le flux de multidiffusion spécifique qui doit être lié au tunnel P. Il peut être représenté en tant que paire (S, G).
Identificateur du tunnel P auquel le flux doit être lié. Cet identificateur est un champ structuré qui

inclut les informations suivantes :

Flux de trafic de multidiffusion dans le tunnel de données MDT :



```
PE1#sh ip pim mdt send
```

```
MDT-data send list for VRF: m-GRE
```

(source, group)	MDT-data group/num	ref_count
(10.1.0.2, 224.1.1.1)	232.2.2.0	1

```
PE3#sh ip pim mdt receive
```

```
Joined MDT-data [group/mdt number : source] uptime/expires for VRF: m-GRE
```

```
[232.2.2.0 : 1.1.1.1] 00:00:41/00:02:18
```

- Si l'OIL contient une interface de tunnel, alors le paquet est encapsulé avec l'utilisation de GRE, la source étant l'adresse d'appairage BGP du routeur PE local et la destination étant l'adresse du groupe MDT.
- La décision de sélection du groupe Data-MDT dépend de la définition de l'indicateur y sur l'entrée (S, G) dans le mVRF.
- Si l'indicateur Z est défini pour l'entrée (S, G) ou (*, G), il s'agit d'une valeur par défaut ou Data-MDT associée à un mVRF.

- Le paquet P doit être désencapsulé pour révéler le paquet C.

Flag	Name	Description
Z	Multicast Tunnel	Indicates that this entry is an IP multicast group that belongs to the Default or Data MDT tunnel. All packets received for this IP multicast state are sent to the MDT tunnel for decapsulation . Set on <u>receiving</u> PE. Global multicast routing table
Y	Joined MDT-data group	Indicates that the traffic was received through a Data MDT tunnel that was set up specifically for this source and group. MVRF multicast routing table
Z	MDT-data group sender	Set on sending PE. Global multicast routing table
y	Sending to MDT-data group	Indicates that the traffic was sent through a Data MDT tunnel that was set up specifically for this source and group. MVRF multicast routing table
V	RD & Vector	
v	Vecor	
E	Extranet source mroute entry	Indicates that a (*, G) or (S, G) entry in the VRF routing table is a source Multicast VRF (MVRF) entry and has extranet receiver MVRF entries linked to it

- Étant donné qu'il n'existe qu'une seule MTI dans le domaine mVRF par domaine de multidiffusion, Data-MDT et Default-MDT utilisent la même interface de tunnel pour le trafic client.
- Les indicateurs Y/y sont nécessaires pour distinguer le trafic MDT par défaut du trafic MDT de données et pour s'assurer que les entrées de routage multicast du client utilisent le groupe MDT-Data correct et se réfèrent à une table interne qui contient les mappages (S, G, Data-MDT).

Dépannage

Il n'existe actuellement aucune information de dépannage spécifique pour cette configuration.