

MDT predefinito multicast di nuova generazione: Profilo 0

Sommario

[Introduzione](#)

[Premesse](#)

[PIM come segnalazione overlay](#)

[Task di configurazione](#)

[Configurazione](#)

[Verifica](#)

[Risoluzione dei problemi](#)

Introduzione

In questo documento viene descritto come il pacchetto Multicast viene attraversato con l'uso del core Multiprotocol Label Switching (MPLS) nel multicast di nuova generazione.

Premesse

MDT predefinito - PIM C - Segnalazione mcast

Draft Rosen usa il Generic Routing Encapsulation (GRE) come protocollo di sovrapposizione. Ciò significa che tutti i pacchetti multicast sono incapsulati nel GRE. Una LAN virtuale viene emulata con tutti i router Provider Edge (PE) della VPN che si uniscono a un gruppo multicast. Questo processo è noto come MDT (Multicast Distribution Tree) predefinito. L'MDT predefinito viene utilizzato per la segnalazione di salve con Protocol Independent Multicast (PIM) e altre segnalazioni PIM, ma anche per il traffico di dati. Se l'origine invia molto traffico, non è efficiente utilizzare l'MDT predefinito ed è possibile creare un MDT dati. I dati MDT includeranno solo i PE che dispongono di ricevitori per il gruppo in uso.

Draft Rosen è abbastanza semplice da distribuire e funziona bene, ma ha alcuni svantaggi. Diamo un'occhiata a questi:

Overhead - GRE aggiunge 24 byte di sovraccarico al pacchetto. Rispetto a MPLS, che in genere aggiunge 8 o 12 byte, ciascun pacchetto contiene il 100% o più di sovraccarico.

PIM nel core - Draft Rosen richiede che PIM sia abilitato nel core in quanto i PE devono unire l'MDT predefinito e/o dei dati che viene eseguito tramite la segnalazione PIM. Se PIM ASM viene utilizzato nel core, è necessario anche un RP. Se PIM SSM viene eseguito nel core, non è necessario alcun RP.

Stato del core - Nel core viene creato uno stato non necessario a causa della segnalazione PIM dai PE. Il nucleo dovrebbe avere il minor stato possibile.

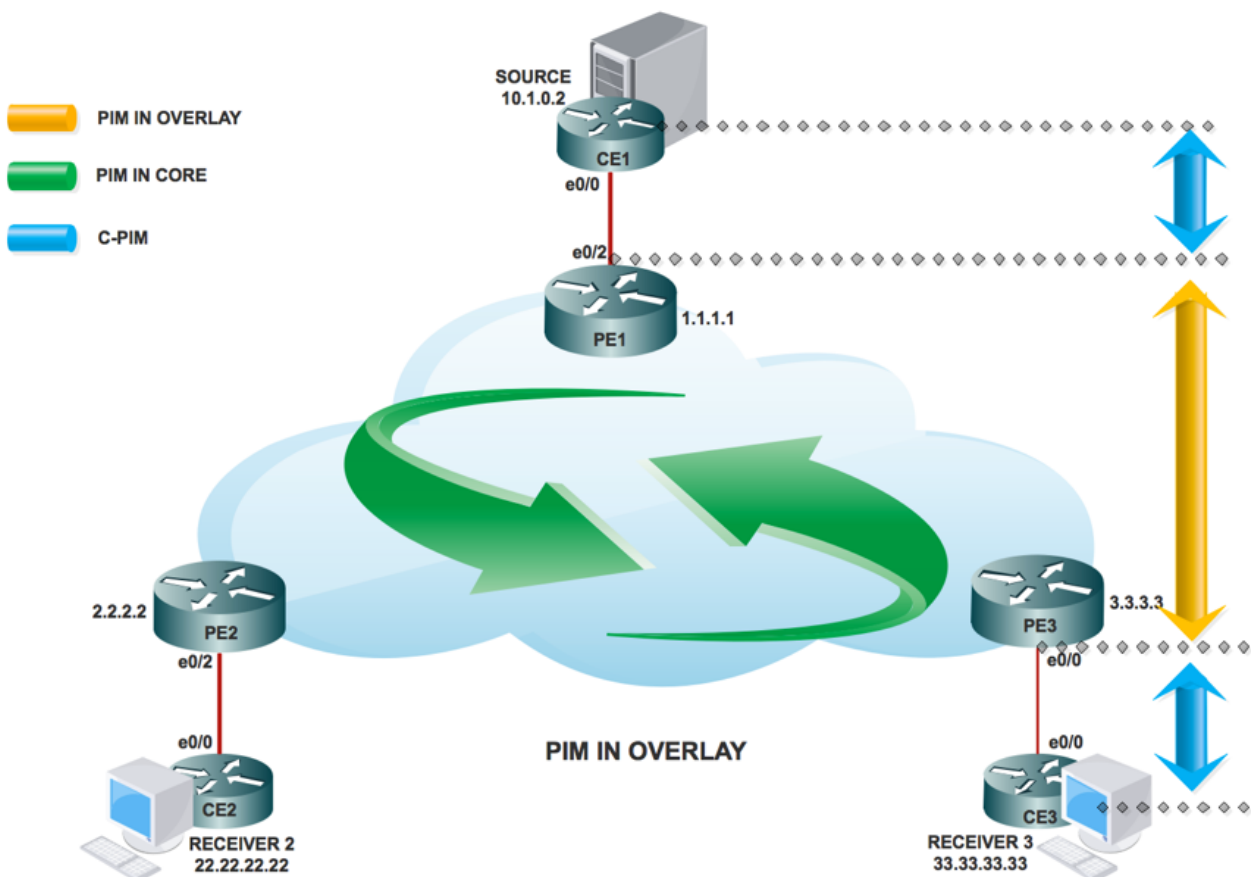
Adiacenze PIM - I PE diventeranno vicini PIM tra loro. Se è una VPN di grandi dimensioni e molti PE, verranno create molte adiacenze PIM. Ciò genera molti segnali di benvenuto e di altro tipo, che aggiungono ulteriore carico al router.

Unicast e multicast: l'inoltro unicast utilizza MPLS, il multicast utilizza GRE. Ciò aggiunge complessità e significa che unicast utilizza un meccanismo di inoltro diverso rispetto al multicast, che non è la soluzione ottimale.

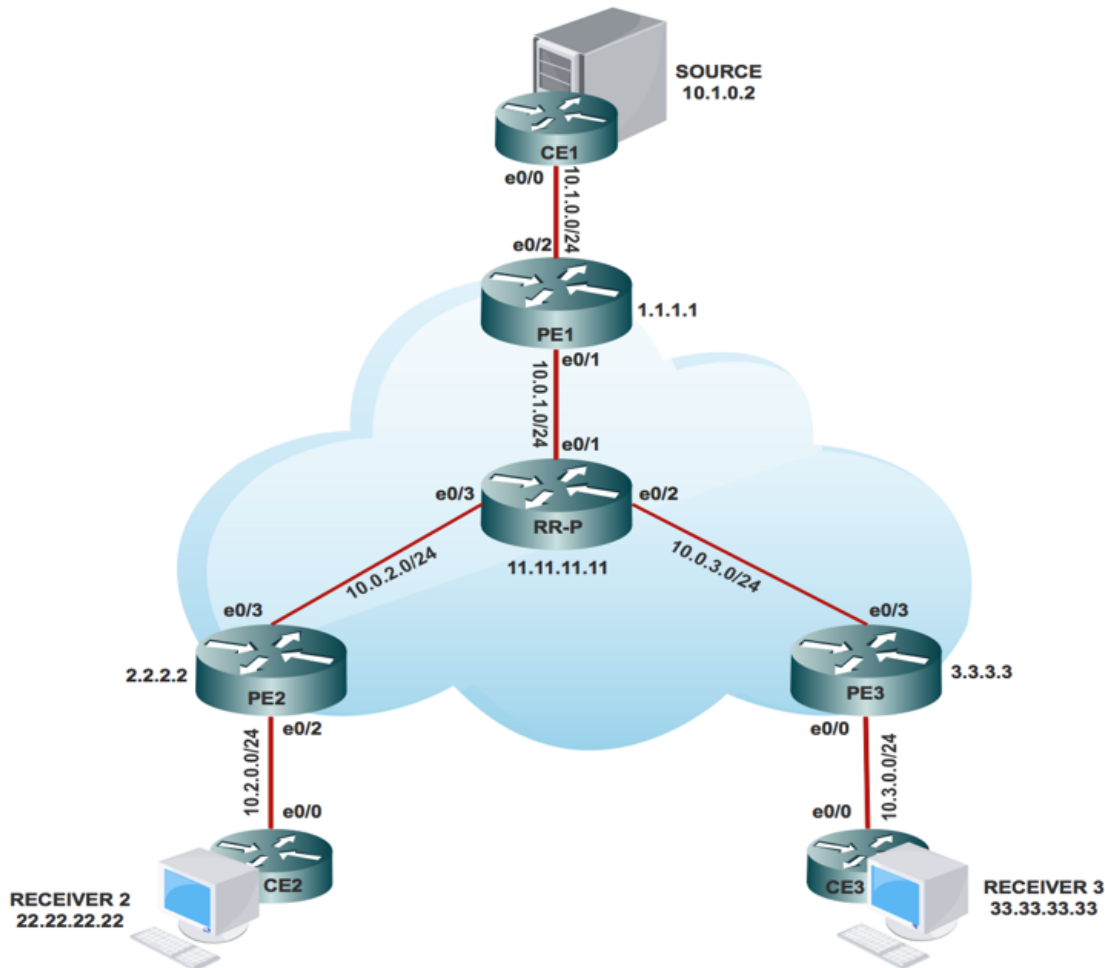
Inefficienza: l'MDT predefinito invia il traffico a tutti i PE della VPN, indipendentemente dal fatto che il PE disponga di un ricevitore nel gruppo (*,G) o (S,G) per il gruppo in uso.

- Per la connessione multicast a tutto il PE in un VRF verrà utilizzato il MDT predefinito.
- Per impostazione predefinita, viene eseguito il collegamento di tutti i router PE.
- Per impostazione predefinita, trasferisce tutto il traffico.
- Tutto il traffico di controllo PIM e il traffico del piano dati. Ad esempio (*,G) Traffico e traffico (S,G).
- Rappresenta da multipunto a multipoint.
- Chiunque può inviare e tutti possono ricevere dall'albero.

PIM come segnalazione overlay



Topologia



Task di configurazione

1. Abilitare il routing multicast su tutti i nodi.
2. Attivare la modalità PIM Sparse in tutta l'interfaccia.
3. Con il VRF esistente, configurare MDT predefinito.
4. Configurare il VRF sull'interfaccia Ethernet0/x.
5. Abilitare il routing multicast su VRF.
6. Configurare PIM SSM Default in tutti i nodi all'interno del core.
7. Configurare BSR RP nel nodo CE.
8. Preconfigurato:
 - VRF m-GRE
 - mBGP: Famiglia di indirizzi VPNv4
 - Protocollo di routing VRF

Configurazione

1. Abilitare il routing multicast su tutti i nodi.

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

2. Abilitare la modalità di sparsità PIM in tutta l'interfaccia.

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

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

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

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

3. Con VRF già esistente, configurare Default MDT.

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

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

4. Configurare il VRF sull'interfaccia Ethernet0/x.

In PE1, PE2 e 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. Abilitare il routing multicast su VRF.

In PE1, PE2 e PE3.

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

6. Configurare RP per la base di Service Provider.

In PE1, PE2, PE3 e Nodo RR-P.

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

7. Configurare BSR RP nel nodo CE (ricevitore).

Sul ricevitore 2.

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

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

Verifica

Fare riferimento a questa sezione per verificare che la configurazione funzioni correttamente.

Attività 1: Verifica connettività fisica

- Verificare che tutte le interfacce collegate siano attive

Attività 2: Verifica unicast VPNv4 per famiglia di indirizzi

- Verificare che BGP sia abilitato in tutti i router per i router AF VPNv4 unicast e i router BGP adiacenti siano "attivi"
- Verificare che la tabella unicast BGP VPNv4 contenga tutti i prefissi del cliente.

Attività 3: Verificare il traffico multicast da un'estremità all'altra.

- Controlla il vicinato di PIM.
- Verificare che lo stato multicast sia stato creato end-to-end.
- Verificare la voce mRIB in PE1, PE2 e PE3
- Verificare che la voce (S,G) mFIB, packet venga incrementata nell'inoltro software.
- Verificare che i pacchetti ICMP vengano raggiunti da CE a 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

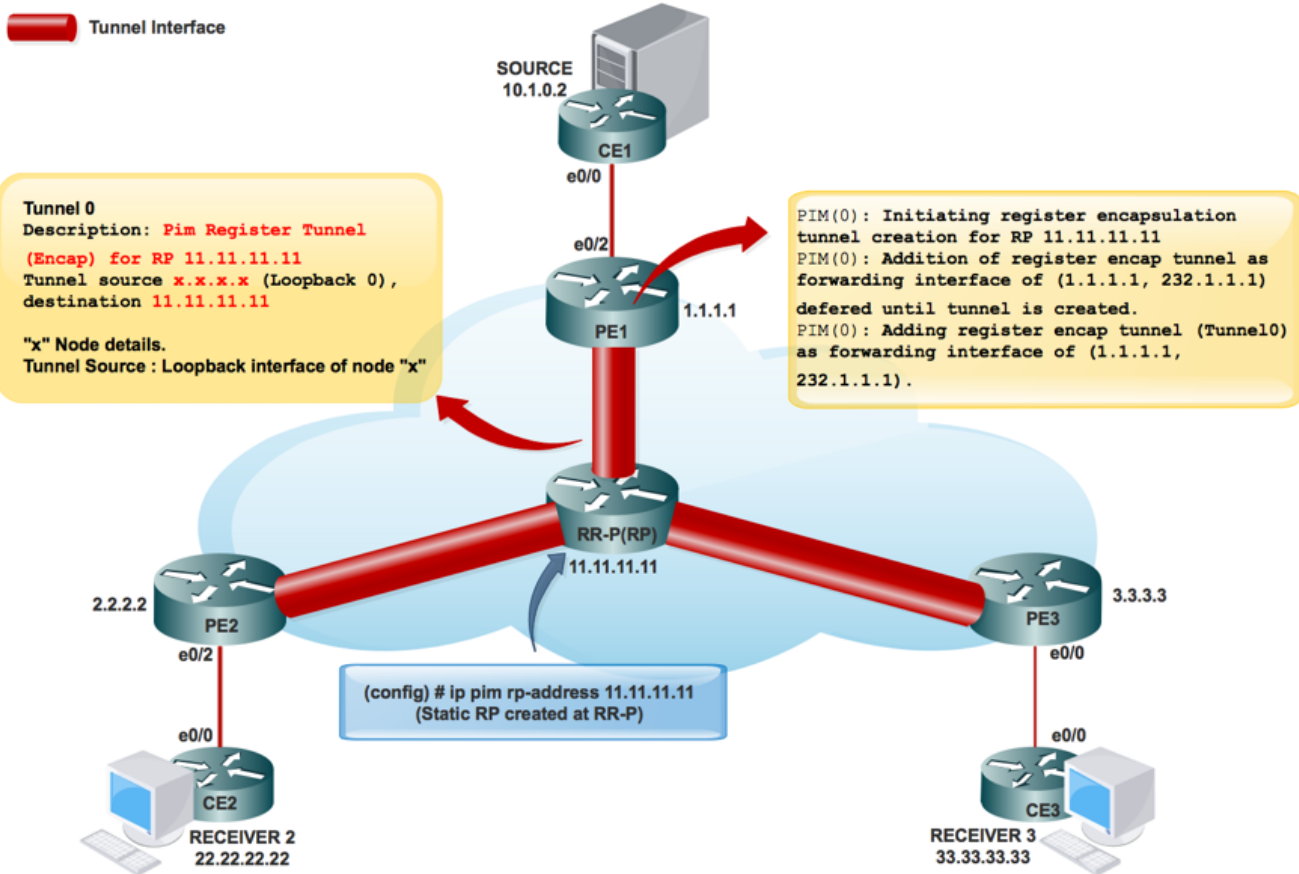
```
SOURCE1#ping 230.1.1.1 repeat 100 timeout 0
Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 230.1.1.1, timeout is 0
seconds:

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
```

Quando si creano le interfacce tunnel:

Creazione RP provider di servizi:

Una volta le informazioni RP si sono inondate nel nucleo. Viene creato il tunnel di interfaccia 0.



PIM(0): Avvio della creazione del tunnel di incapsulamento del registro per RP 11.11.11.11.

PIM(0): Creazione iniziale del tunnel del registro completata per RP 11.11.11.11.

PIM(0): L'aggiunta del tunnel register encap come interfaccia di inoltro di (1.1.1.1, 232.1.1.1) è stata rinviata fino alla creazione del tunnel.

9 maggio 17:34:56.155: PIM(0): Controllare RP 11.11.11.11 nel (, 232.1.1.1).

PIM(0): Aggiunta del tunnel register encap (Tunnel0) come interfaccia di inoltro di (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

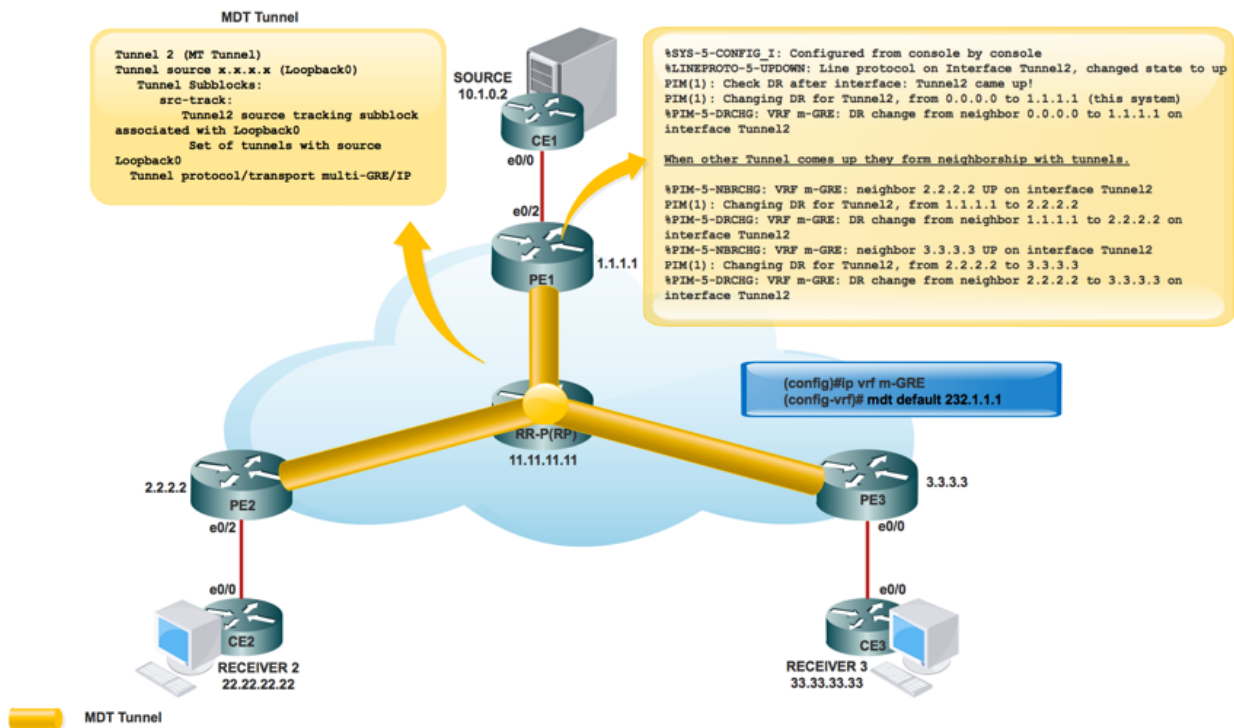
Set of tunnels with source Ethernet0/1, 1 member (includes iterators), on interface <OK>

Tunnel protocol/transport PIM/IPv4

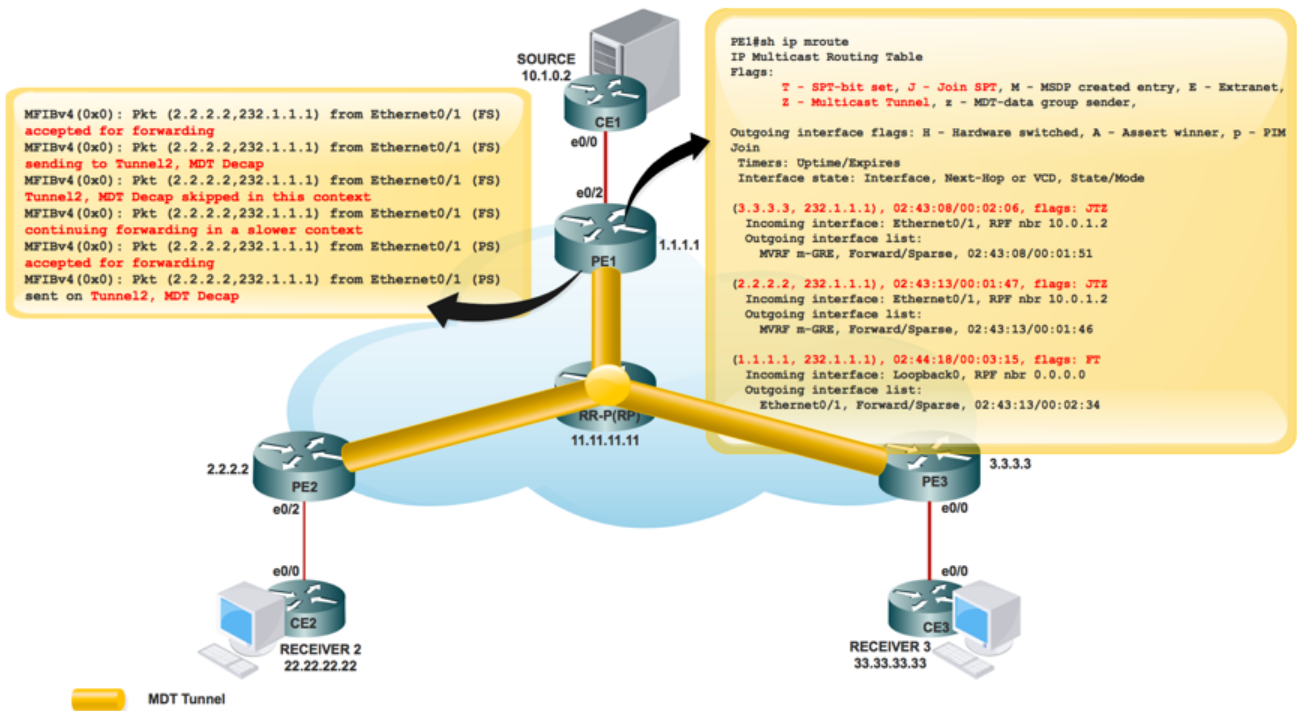
Tunnel TOS/Traffic Class 0xC0, Tunnel TTL 255

Tunnel transport MTU 1472 bytes

Creazione tunnel MDT:



Creazione MRIB nel core:



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:

MVRFP 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:

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

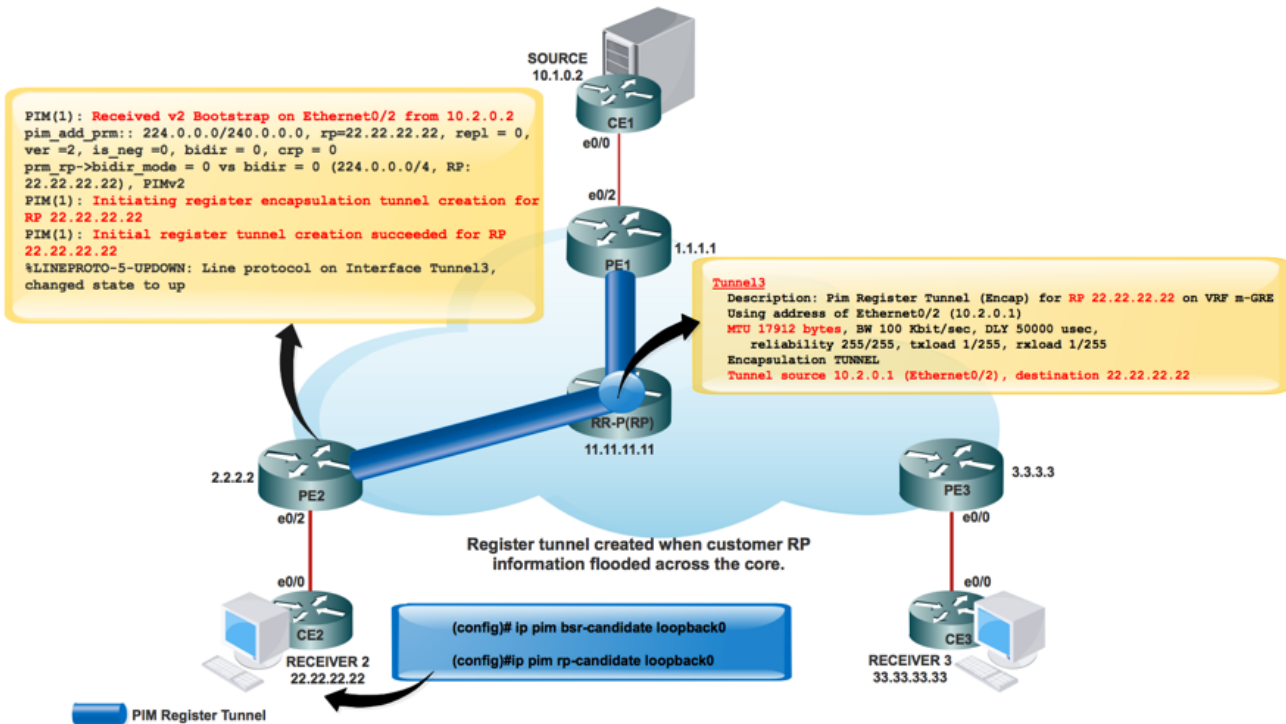
(1.1.1.1, 232.1.1.1), 00:10:15/00:03:20, flags: FT

Incoming interface: Loopback0, RPF nbr 0.0.0.0

Outgoing interface list:

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

Una volta creato l'RP per la rete del cliente:



*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

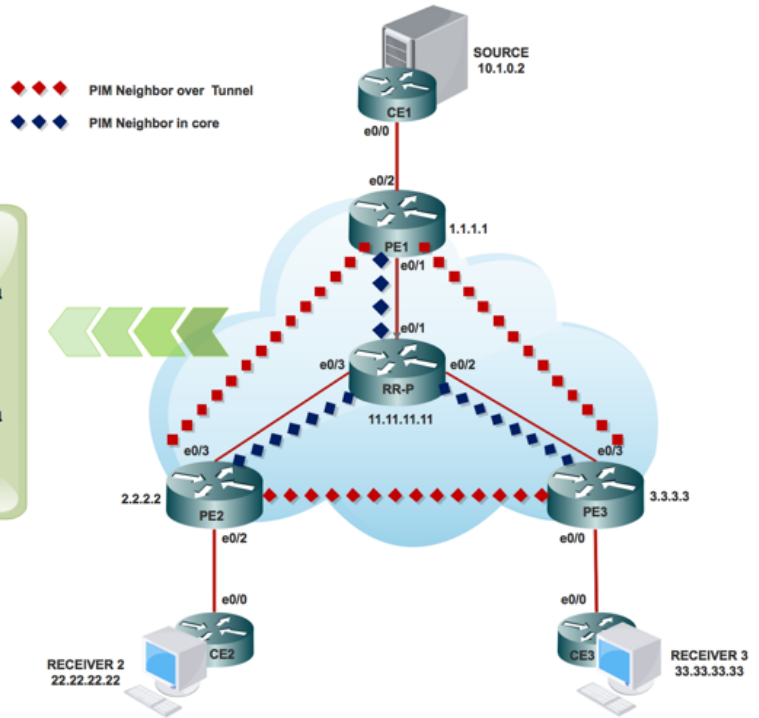
L'interfaccia tunnel viene creata per trasportare le informazioni RP del cliente.

PIM(1): Avvio della creazione del tunnel di incapsulamento del registro per RP 22.22.22.22.

È il tunnel creato per eseguire l'incapsulamento del registro al protocollo RP.

Per ogni RP in modalità sparsa individuato, viene creato un tunnel di incapsulamento del registro. Sul router RP in modalità sparsa, è presente un'interfaccia del tunnel di decapsulamento creata per ricevere i pacchetti Register.

Villaggio 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

Flusso dei pacchetti:

Il flusso del pacchetto del piano di controllo si divide in due parti.

1. Il ricevitore è in linea.

2. Origine attiva.

Quando il ricevitore è attivo:

```

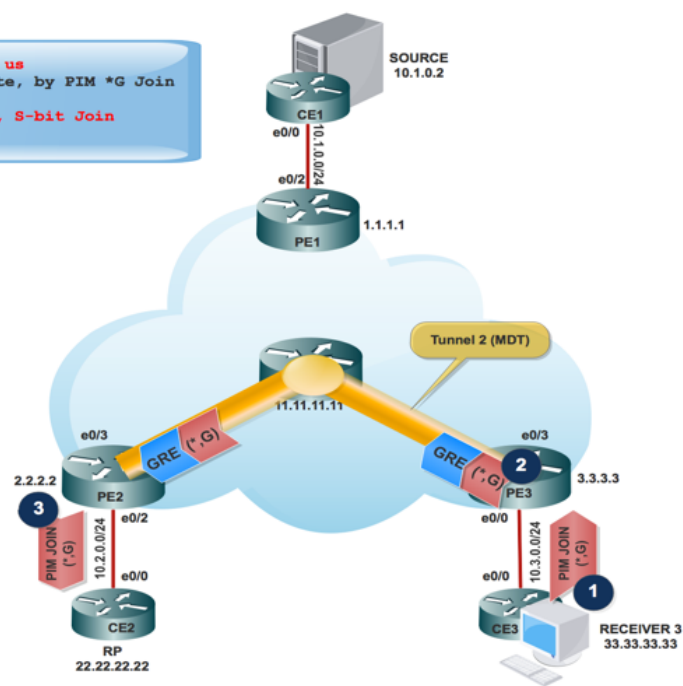
1
PIM(1): Received v2 Join/Prune on Ethernet0/0 from 10.3.0.2, to us
PIM(1): Add Ethernet0/0/10.3.0.2 to (*, 224.1.1.1), Forward state, by PIM *G Join
PIM(1): Upstream mode for (*, 224.1.1.1) changed from 0 to 1
PIM(1): Adding v2 (22.22.22.22/32, 224.1.1.1), WC-bit, RPT-bit, S-bit Join
PIM(1): Send v2 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. Il ricevitore viene connesso e invia PIM JOIN (*,G) a PE3.

2. PE3 incapsula il PIM JOIN (*,G) nel pacchetto GRE e lo invia tramite il tunnel 2 (tunnel MDT), verificato dall'interfaccia in arrivo di **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 ha ricevuto il pacchetto GRE con Source impostato su 3.3.3.3 e Destination 232.1.1.1 e lo inoltra al MVRF m-GRE basato su 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
```

Il pacchetto GRE viene decapsulato e PIM JOIN lo invia all'RP.

Nota: RPF Neighbor è 2.2.2.2 perché l'aggiunta PIM è destinata all'indirizzo RP per formare l'RPT attraverso il nucleo.

Nota: Bit WC e bit RPT: Attivato dallo stato (*,G), il DR crea un messaggio Join/Prune con l'indirizzo RP nel proprio elenco di join e il bit jolly (WC-bit) e il bit RP-tree (RPT-bit) impostato su 1. Il bit WC indica che qualsiasi origine può corrispondere e essere inoltrato in base a questa voce se non c'è più corrispondenza; il bit RPT indica che questo join viene inviato all'albero RP condiviso. L'elenco di prugne è vuoto. Quando il bit RPT è impostato su 1, indica che il join è associato all'albero RP condiviso e pertanto il messaggio Join/Prune viene propagato lungo l'albero RP. Quando il bit WCF è impostato su 1, l'indirizzo è un RP e i ricevitori a valle si aspettano di ricevere pacchetti da tutte le fonti tramite questo percorso (struttura condivisa).

```
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. Il pacchetto incapsulato GRE raggiunge l'origine 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) raggiunge l'origine CE.

3. A questo punto l'origine riceve le informazioni del ricevitore interessato e il traffico inizia a essere inviato all'origine PE1.

4. All'origine 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
```

In 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
```

Acquisizione PCAP di pacchetti multicast da PE1. Tunneling eseguito nel tunnel predefinito MDT. Incapsulato con GRE.

5. Al ricevitore PE3, il pacchetto viene ricevuto.

```
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
```

Nota: il router adiacente RPF viene modificato dopo la ricezione di un pacchetto multicast da PE1. In precedenza era PE2 come RP ospitato dietro di esso. Dopo aver ricevuto il primo pacchetto multicast, modifica l'RPF e imposta il bit SPT.

Flusso del traffico sul tunnel MDT predefinito:

- L'inoltro sull'MDT utilizza il GRE, il pacchetto C diventa un pacchetto P.
- Indirizzo IP-Packet S = indirizzo peer BGP di PE
G address = MDT-Group address (predefinito o dati)
- C-Packet IP TOS verrà copiato in P-Packet.
- Le etichette MPLS NON vengono utilizzate nel core, ma solo nel multicast nativo.

Flusso dei pacchetti:

1. Un pacchetto C arriva su un'interfaccia PE configurata VRF, mVRF viene identificato in modo implicito. Normale controllo RPF su origine C.

Il C-Packet ha replicato l'interfaccia in uscita nell'OIL. A questo punto si tratterebbe di un'interfaccia PE nello stesso 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)
```

Se l'OLIO contiene un MTI, il pacchetto C incapsula in un pacchetto P. Se il flag "y" è impostato sulla voce, la destinazione utilizzata è il gruppo DATA-MDT, altrimenti il gruppo MDT predefinito. L'origine è l'indirizzo peer BGP PE e la destinazione è l'indirizzo del gruppo MDT.

2. Il pacchetto IP viene inoltrato attraverso la rete IP in base al normale multicast.

Il pacchetto arriva all'interfaccia globale. Voce globale (S,G) o (*,G) per il gruppo MDT a cui si fa riferimento. Normale controllo RPF su P-Source (Peer PE).

3. Il pacchetto IP viene replicato dall'interfaccia in OIL. A questo punto si tratta di P/PE nella tabella di route globale.

4. Se il flag "Z" imposta il pacchetto, il pacchetto viene decapsulato per rivelare il pacchetto C. L'mVRF di destinazione e l'interfaccia in arrivo derivata dal gruppo MDT è la destinazione dell'intestazione incapsulata.

Controllo RPF del pacchetto C in mVRF eseguito, pacchetto C replicato in olio in 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,

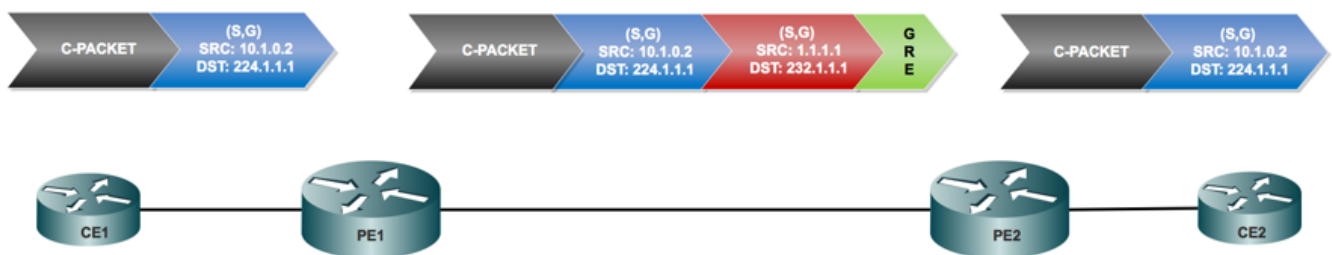
(1.1.1.1, 232.1.1.1), 1d01h/00:02:47, flags: **JTZ**

Incoming interface: Ethernet0/3, RPF nbr 10.0.3.2

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

5. Il pacchetto C nativo raggiunge il ricevitore 3.

Incapsulamento pacchetti:

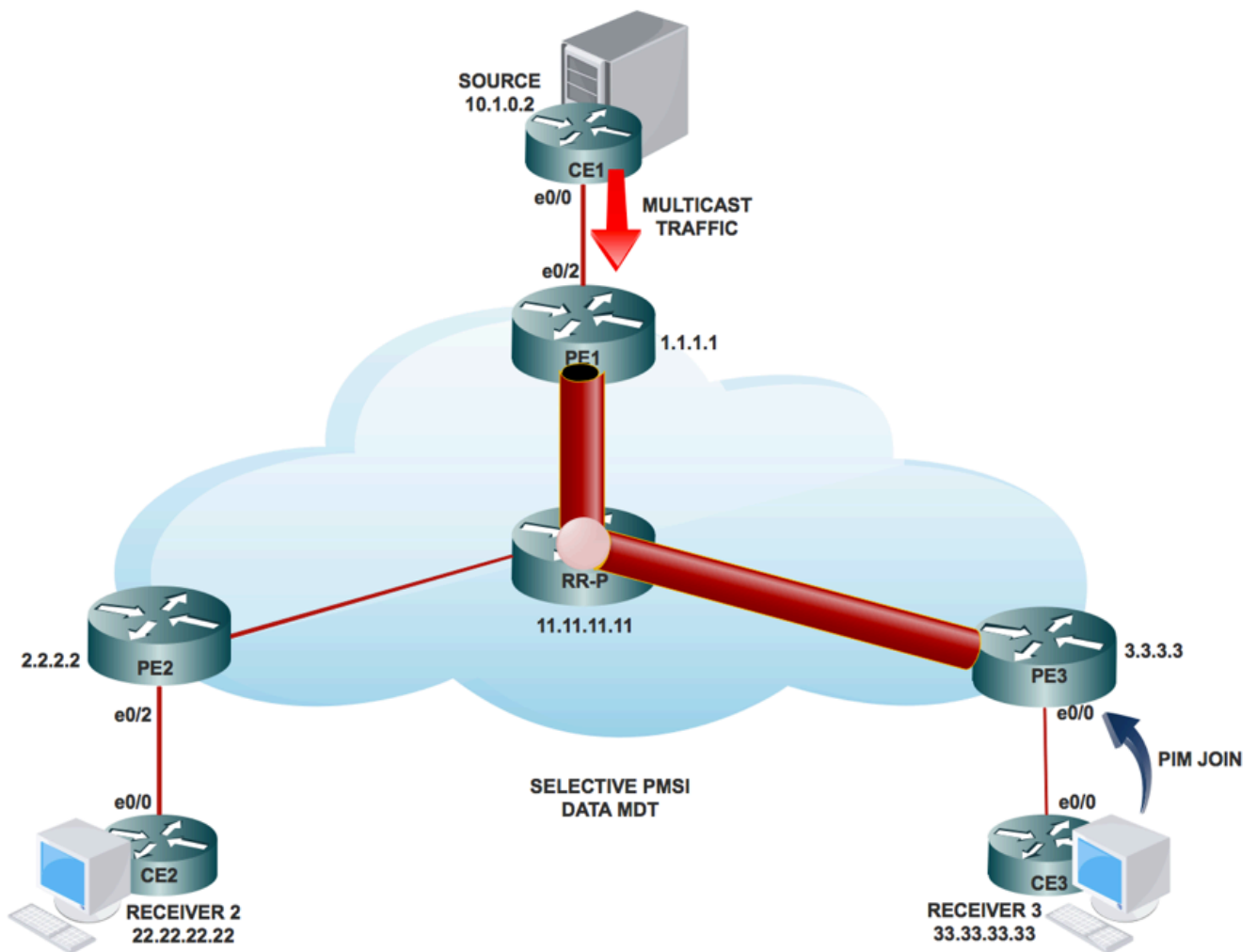


MDT dati:

Cos'è Data MDT?

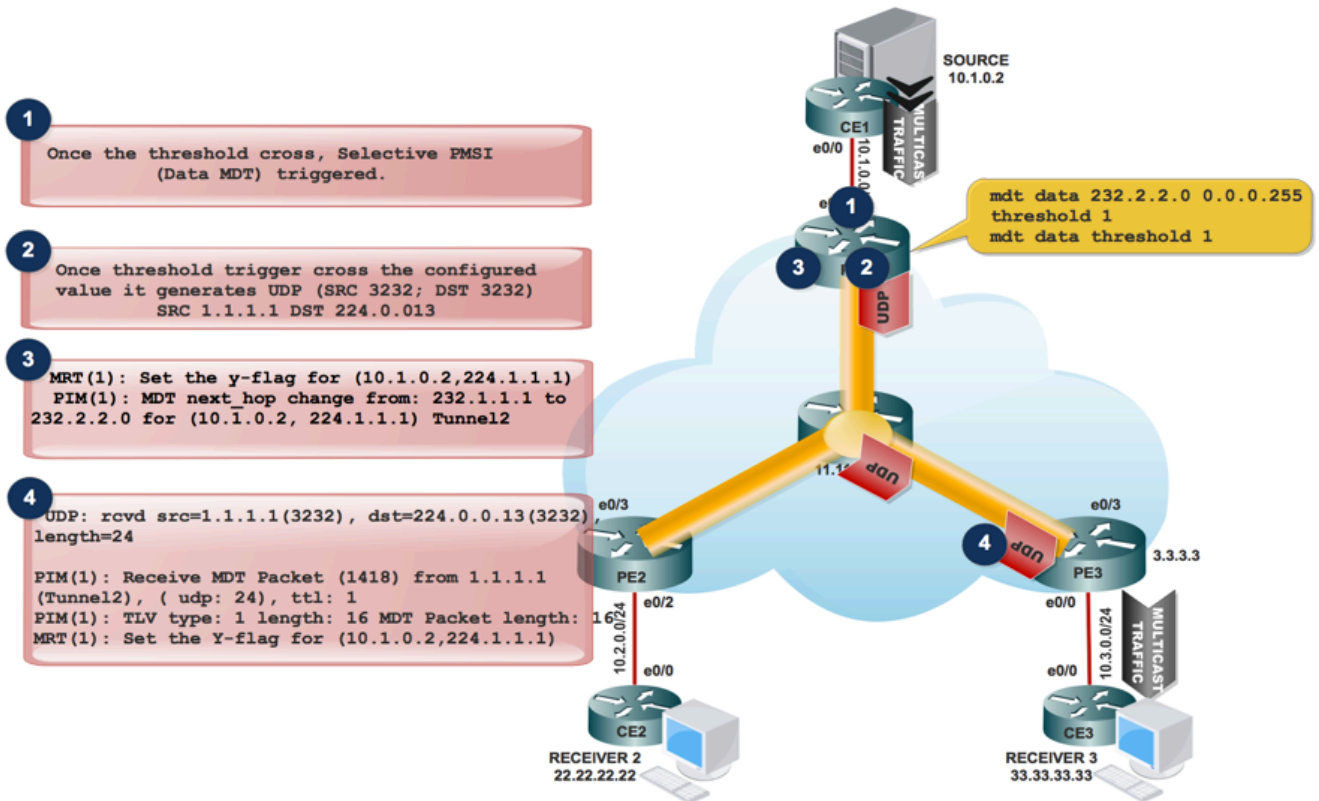
È facoltativo. Viene creato su richiesta, trasporta traffico specifico (S,G). Nell'ultima release di IOS®, la soglia configurata è "0" e "infinite". Ogni volta che un primo pacchetto raggiunge il VRF, il MDT dati viene inizializzato e se il valore è infinito il MDT dati non verrà mai creato e il traffico si sposterà in avanti nel MDT predefinito. Data MDT è sempre l'albero di ricezione e non invia mai traffico. MDT dati è solo per il traffico (S,G).

PMSI selettivo:



- È facoltativo. Viene creato su richiesta, trasporta traffico specifico (S, G).
- Ogni volta che un primo pacchetto raggiunge il VRF, il MDT dati viene inizializzato e, se il valore è infinito, il MDT dati non viene mai creato e il traffico si sposta in avanti nel MDT predefinito.
- L'MDT dei dati è sempre l'albero ricevente e non invia mai traffico. MDT dati è solo per il traffico (S, G).
- Il messaggio PIM trasporta il gruppo C- (S, G) & P.

Modalità di creazione di DATA MDT:



1. Quando il traffico multicast entra nel VRF e quando la velocità del traffico raggiunge la soglia. Genera un pacchetto MDT.
2. Il pacchetto MDT viene incapsulato in UDP con origine e destinazione 3232. E inviarlo al destinatario interessato.

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. Dopo aver inviato il pacchetto UDP al destinatario interessato, imposta il flag "y" e cambia l'hop successivo MDT al nuovo indirizzo del gruppo MDT.

All'origine 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)

Nota: L'hop successivo OIL diventa 232.2.2.0.

2. in PE3, quando riceve il pacchetto MDT incapsulato nella porta UDP SRC 3232 e nella porta 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:

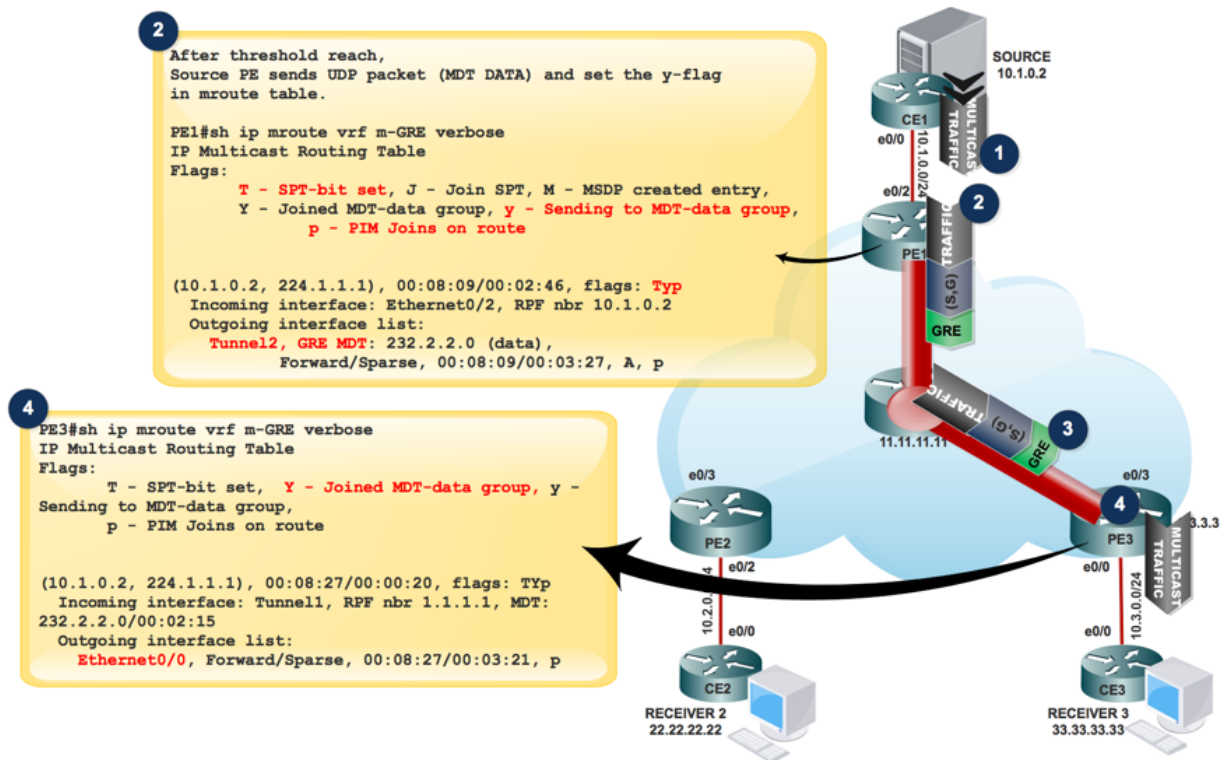
Ethernet0/0, Forward/Sparse, 00:08:27/00:03:21, p

Il messaggio Join S-PMSI è un messaggio incapsulato da UDP il cui indirizzo di destinazione è ALL-PIM-ROUTERS (224.0.0.13) e la cui porta di destinazione è 3232.

Il messaggio Join S-PMSI contiene le seguenti informazioni: Un identificatore per il particolare flusso multicast da associare al tunnel P. Può essere rappresentata come una coppia (S,G). Identificatore del tunnel IP specifico a cui deve essere associato il flusso. Questo identificatore è

un campo strutturato che include le informazioni seguenti:

Flusso del traffico multicast nel tunnel DATI 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
```

- Se l'interfaccia del tunnel OIL contiene un'interfaccia tunnel, il pacchetto viene incapsulato con l'uso del GRE, dove l'origine è l'indirizzo di peer BGP del router PE locale e la destinazione è l'indirizzo del gruppo MDT.
- La decisione di selezionare il gruppo Data-MDT dipende dall'impostazione o meno del flag y sulla voce (S, G) nell'mVRF.
- Se per la voce (S, G) o (*, G) è impostato il flag Z, si tratta di un MDT di default o di dati con un mVRF associato.

- Il pacchetto P deve essere decapsulato per rivelare il pacchetto 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 mulitcast 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 mulitcast 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

- Poiché nel mVRF esiste solo un MTI per dominio multicast, sia Data-MDT che Default-MDT utilizzano la stessa interfaccia tunnel per il traffico del cliente.
- I flag Y/y sono necessari per distinguere il traffico Default-MDT dal traffico Data-MDT e garantire che le voci di routing multicast del cliente utilizzino il gruppo di dati MDT corretto e facciano riferimento a una tabella interna che contiene i mapping (S, G, Data-MDT).

Risoluzione dei problemi

Al momento non sono disponibili informazioni specifiche per la risoluzione dei problemi di questa configurazione.