

MDT padrão multicast de próxima geração: Perfil 0

Contents

[Introduction](#)

[Informações de Apoio](#)

[PIM como sinalização de sobreposição](#)

[Tarefas de configuração](#)

[Configurar](#)

[Verificar](#)

[Troubleshoot](#)

Introduction

Este documento descreve como o pacote Multicast atravessa com o uso do núcleo Multiprotocol Label Switching (MPLS) no Multicast de Próxima Geração.

Informações de Apoio

MDT padrão - PIM C - sinalização mcast

O Draft Rosen usa o Generic Routing Encapsulation (GRE) como um protocolo de sobreposição. Isso significa que todos os pacotes multicast são encapsulados dentro do GRE. Uma LAN virtual é emulada com todos os roteadores de Borda do Provedor (PE - Provider Edge) no grupo de junção de VPN. Isso é conhecido como Árvore de distribuição multicast (MDT) padrão. O MDT padrão é usado para a saudação do Protocol Independent Multicast (PIM) e outra sinalização de PIM, mas também para o tráfego de dados. Se a origem envia muito tráfego, é ineficiente usar o MDT padrão e um MDT de dados pode ser criado. O MDT de dados incluirá apenas PEs que tenham receptores para o grupo em uso.

Rascunho Rosen é bem simples de implantar e funciona bem, mas tem algumas desvantagens. Vamos dar uma olhada nestes itens:

Overhead - O GRE adiciona 24 bytes de sobrecarga ao pacote. Comparado ao MPLS, que normalmente adiciona 8 ou 12 bytes, há 100% ou mais de carga adicional adicionada a cada pacote.

PIM no núcleo - Rascunho Rosen requer que o PIM esteja ativado no núcleo porque os PEs devem se unir ao MDT padrão e/ou de dados, o que é feito através da sinalização PIM. Se o PIM ASM for usado no núcleo, um RP também será necessário. Se o PIM SSM for executado no núcleo, nenhum RP será necessário.

Estado principal - O estado desnecessário é criado no núcleo devido à sinalização PIM do PE. O núcleo deve ter o menor estado possível.

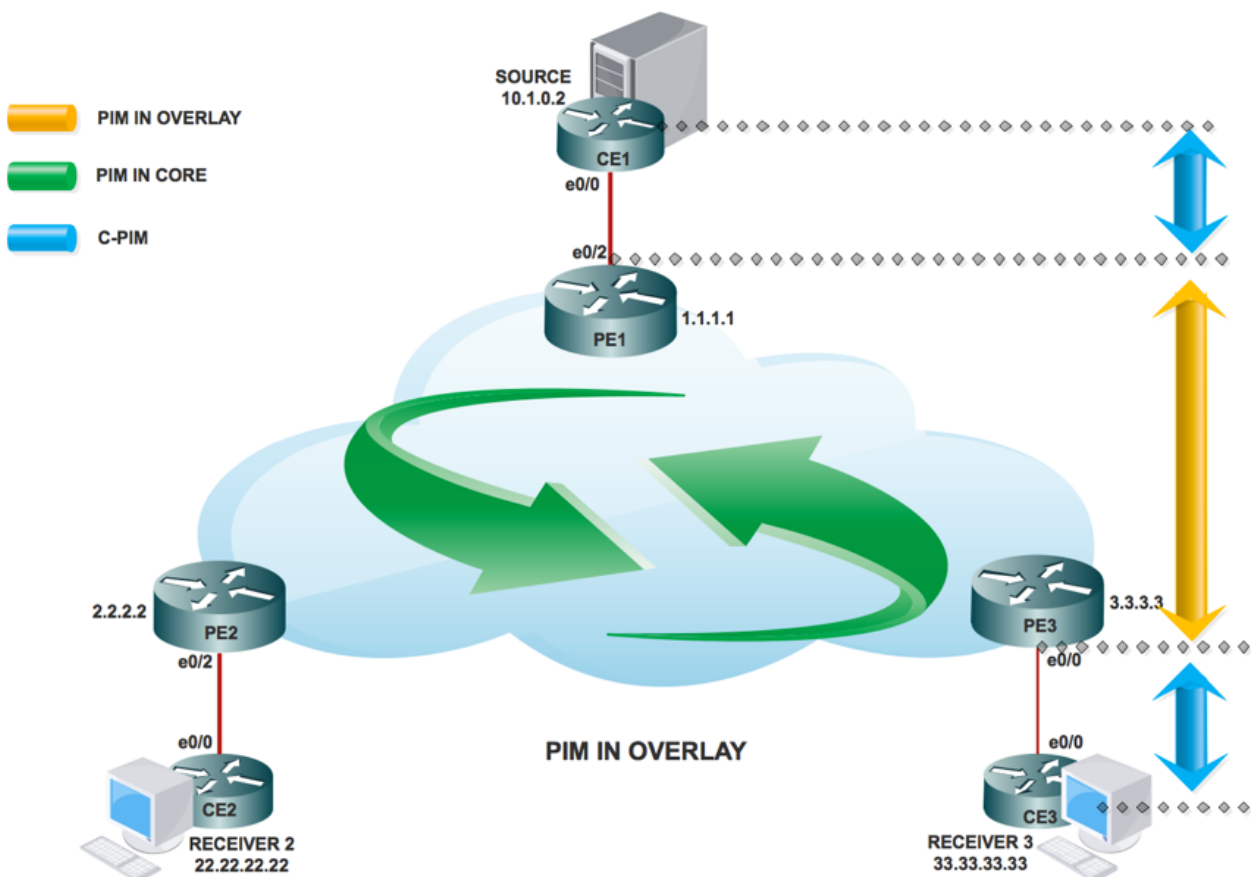
Adjacências de PIM - Os PEs se tornarão vizinhos de PIM entre si. Se for uma VPN grande e muito PEs, muitas adjacências de PIM serão criadas. Isso gera muitos sinais de hello e outros sinais que aumentam a carga do roteador.

Unicast vs multicast - O encaminhamento unicast usa MPLS, o multicast usa GRE. Isso aumenta a complexidade e significa que o unicast usa um mecanismo de encaminhamento diferente do multicast, que não é a solução ideal.

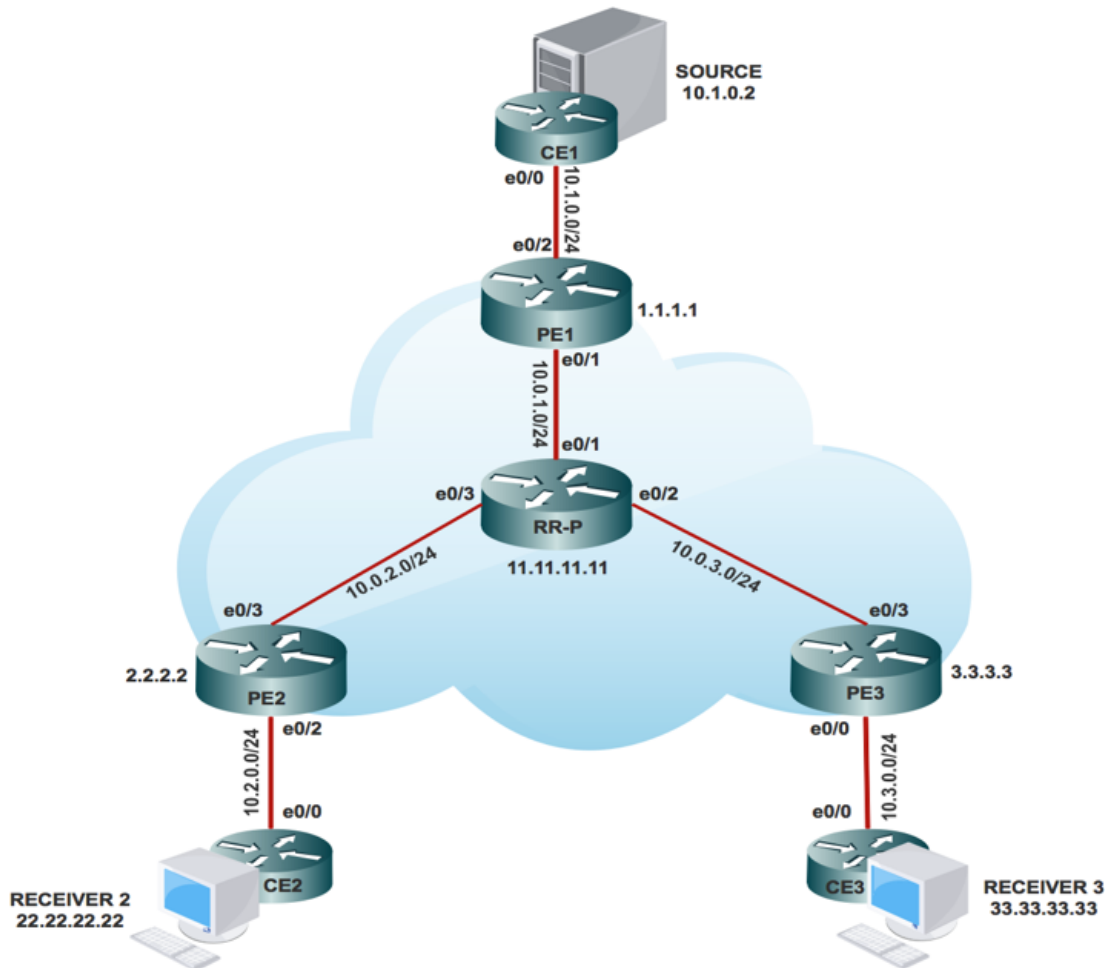
Ineficiência - O MDT padrão envia tráfego para todos os PEs na VPN, independentemente de o PE ter um receptor no (*,G) ou (S,G) para o grupo em uso.

- O MDT padrão será usado para conectar multicast a todo o PE em um VRF.
- Padrão significa que ele conecta todos os roteadores PE.
- Por padrão, ele transporta todo o tráfego.
- Todo o tráfego de controle do PIM e o tráfego do plano de dados. Por exemplo, tráfego (*,G) e tráfego (S,G).
- Isso representa multiponto para multiponto.
- Qualquer um pode enviar e todos podem receber da árvore.

PIM como sinalização de sobreposição



Topologia



Tarefas de configuração

1. Ative o roteamento multicast em todos os nós.
2. Ative o PIM Sparse Mode em toda a interface.
3. Com o VRF existente, configure o MDT padrão.
4. Configure o VRF na interface Ethernet0/x.
5. Ative o roteamento multicast em VRF.
6. Configure PIM SSM Default em todos os nós dentro do núcleo.
7. Configure o RP BSR no nó CE.
8. Pré-configurado:
 - VRF m-GRE
 - MBGP: Endereço da família VPNv4
 - Protocolo de roteamento VRF

Configurar

1. Ative o roteamento multicast em todos os nós.

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

2. Ative o PIM Sparse Mode em toda a interface.

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

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

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

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

3. Com o VRF que já existe, configure o MDT padrão.

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

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

4. Configure o VRF na interface Ethernet0/x.

Em 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. Ative o roteamento multicast em VRF.

Em PE1, PE2 e PE3.

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

6. Configure o RP para o núcleo do provedor de serviços.

Em nó PE1, PE2, PE3 e RR-P.

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

7. Configure o RP BSR no nó CE (receptor).

No Receptor2.

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

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

Verificar

Use esta seção para confirmar se a sua configuração funciona corretamente.

Tarefa 1: Verificar a conectividade física

- Verifique se toda a interface conectada está "UP"

Tarefa 2: Verificar unicast da família de endereços VPNv4

- Verifique se o BGP está ativado em todos os roteadores para unicast AF VPNv4 e os vizinhos BGP estão "UP"
- Verifique se a tabela unicast do BGP VPNv4 tem todos os prefixos do cliente.

Tarefa 3: Verifique o tráfego multicast de ponta a ponta.

- Verifique a vizinhança do PIM.
- Verifique se o estado multicast é criado de ponta a ponta.
- Verifique a entrada mRIB em PE1, PE2 e PE3
- Verifique se a entrada (S,G) mFIB, pacote sendo incrementado no encaminhamento de software.
- Verifique se os pacotes ICMP estão chegando de 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
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
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

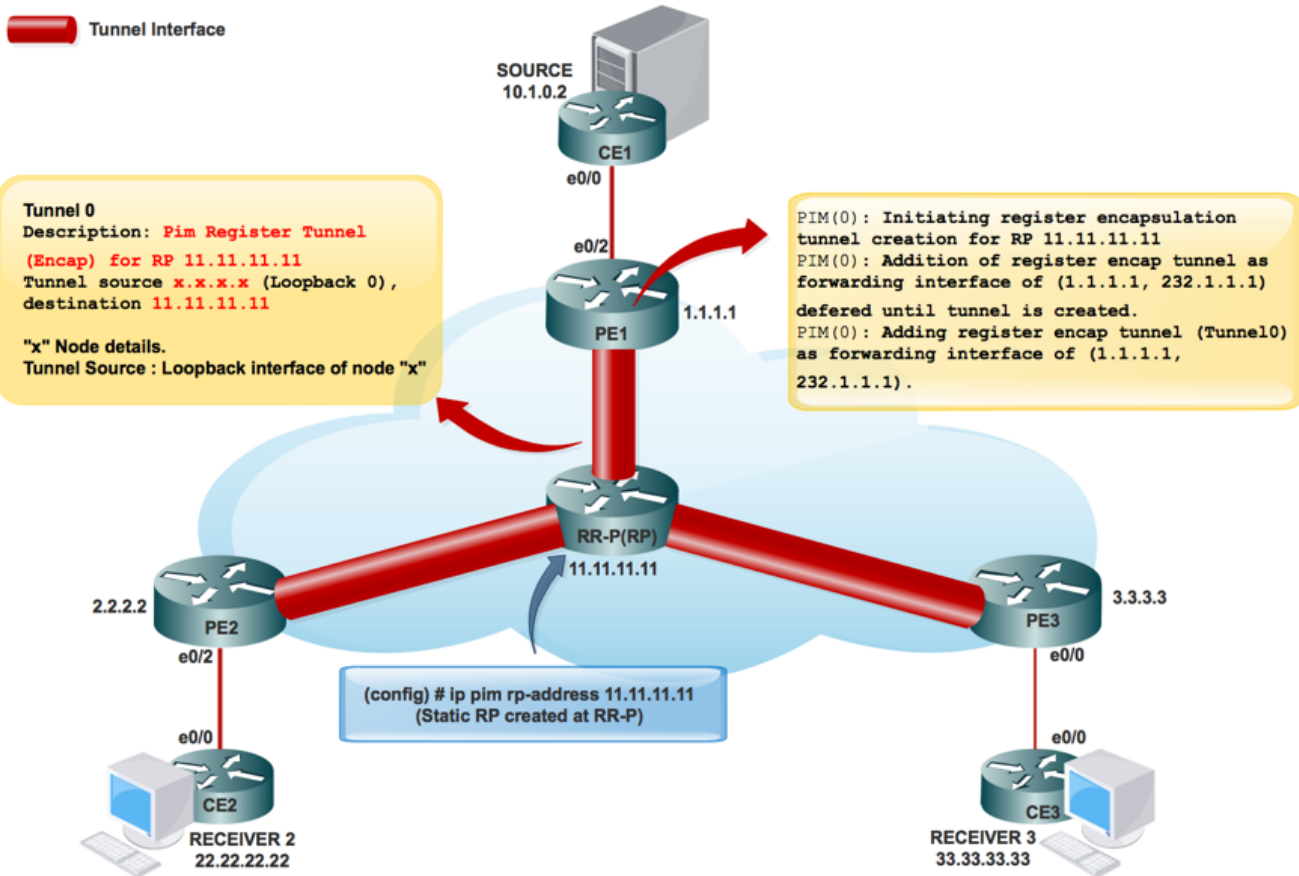
Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 230.1.1.1, timeout is 0
seconds:

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 as interfaces de túnel são criadas:

Criação do RP do provedor de serviços:

Depois que as informações de RP inundarem o núcleo. O túnel de interface 0 é criado.



PIM(0): Iniciando a criação do túnel de encapsulamento de registro para RP 11.11.11.11.

PIM(0): Criação inicial de túnel de registro bem-sucedida para RP 11.11.11.11.

PIM(0): Adição de túnel encap de registro como interface de encaminhamento de (1.1.1.1, 232.1.1.1) adiada até que o túnel seja criado.

9 de maio 17:34:56.155: PIM(0): Verifique o RP 11.11.11.11 no (, 232.1.1.1).

PIM(0): Adicionando túnel encap de registro (Tunnel0) como interface de encaminhamento 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

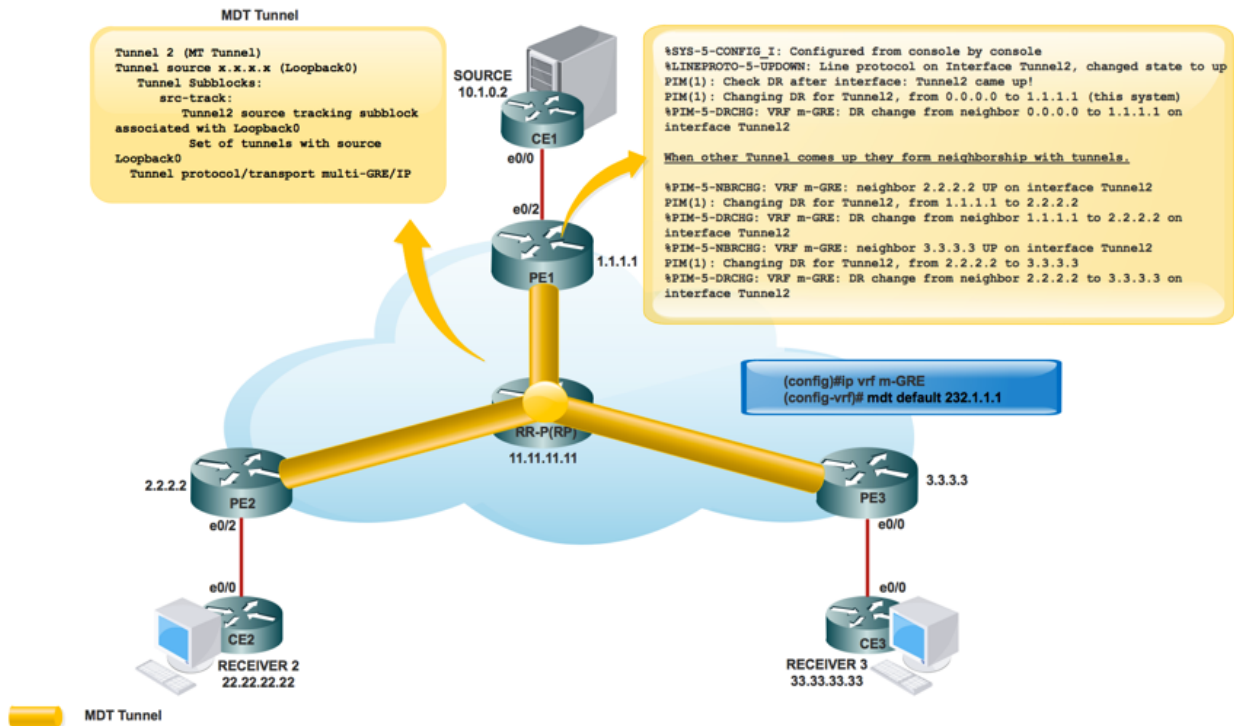
Set of tunnels with source Ethernet0/1, 1 member (includes iterators), on interface <OK>

Tunnel protocol/transport PIM/IPv4

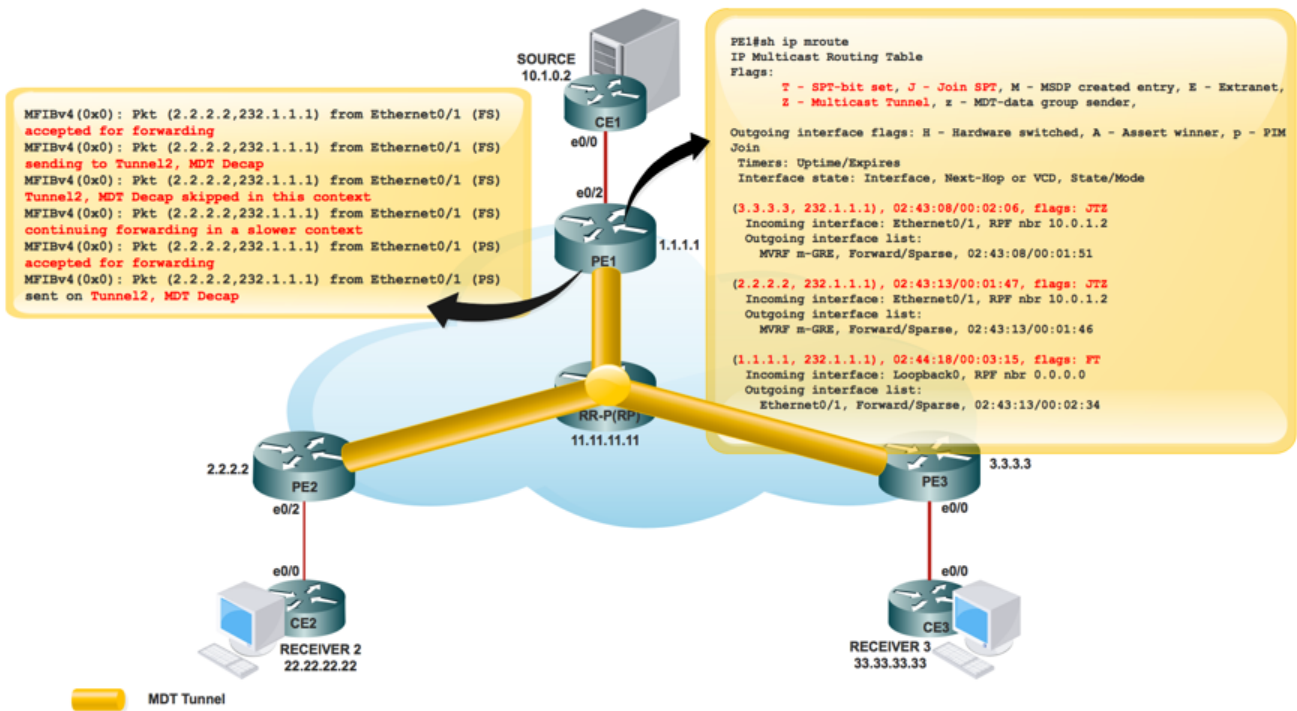
Tunnel TOS/Traffic Class 0xC0, Tunnel TTL 255

Tunnel transport MTU 1472 bytes

Criação de túnel MDT:



Criação de MRIB no núcleo:



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

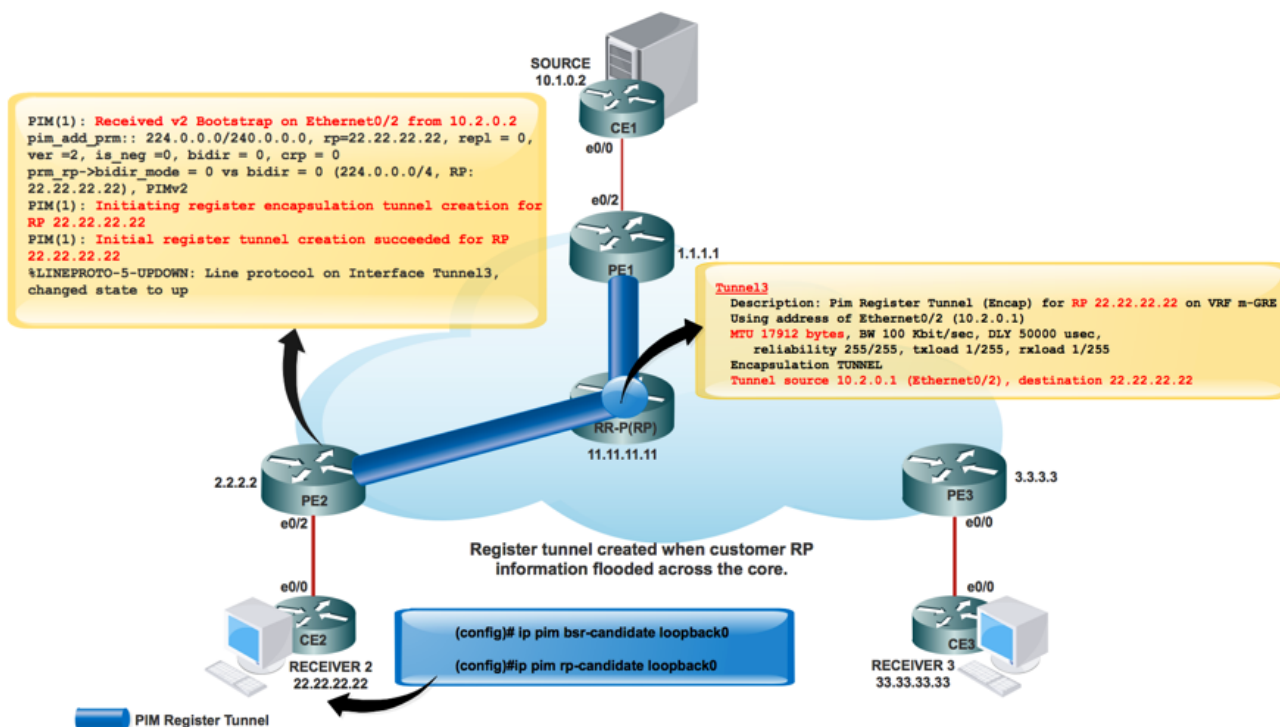
(1.1.1.1, 232.1.1.1), 00:10:15/00:03:20, flags: FT

Incoming interface: Loopback0, RPF nbr 0.0.0.0

Outgoing interface list:

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

Depois que o RP for criado para a rede do 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

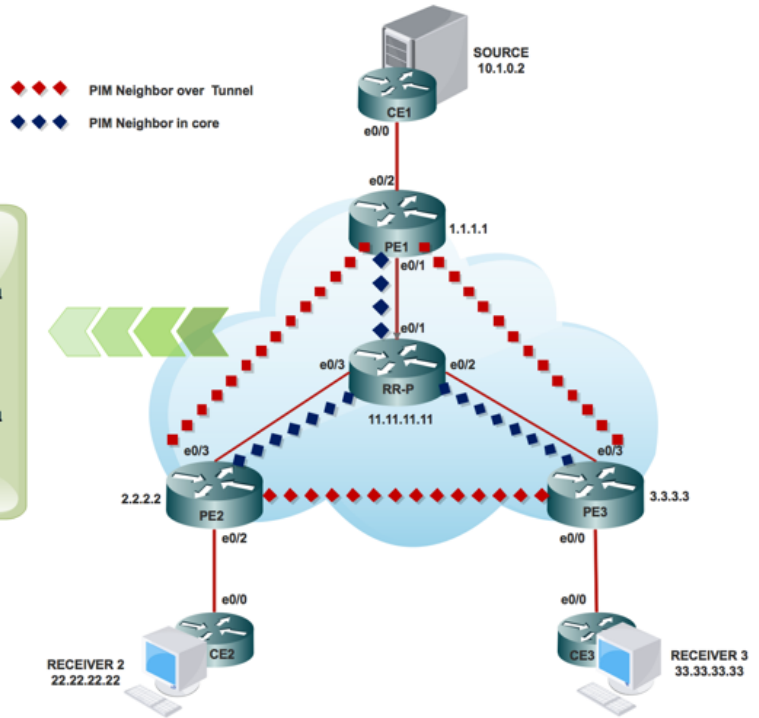
A interface de túnel é criada para transportar as informações de RP do cliente.

PIM(1): Iniciando a criação do túnel de encapsulamento de registro para RP 22.22.22.22.

Éo túnel criado para fazer o encapsulamento Register para RP.

Para cada RP de modo esparsa descoberto, um túnel de encapsulamento de registro é criado. No próprio RP de modo esparsa, há uma interface de túnel de desencapsulamento criada para receber pacotes de registro.

Vizinho 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

Fluxo de pacote:

O fluxo de pacote do plano de controle divide-se em duas partes.

1. O receptor está online.

2. A origem está ativa.

Quando o receptor está ativo:

```

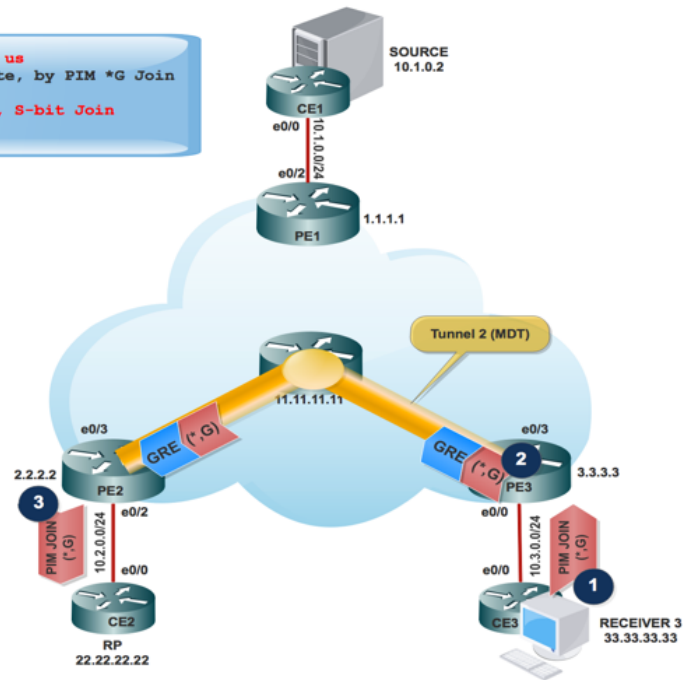
1
PIM(1): Received v2 Join/Prune on Ethernet0/0 from 10.3.0.2, to us
PIM(1): Add Ethernet0/0/10.3.0.2 to (*, 224.1.1.1), Forward state, by PIM *G Join
PIM(1): Upstream mode for (*, 224.1.1.1) changed from 0 to 1
PIM(1): Adding v2 (22.22.22.22/32, 224.1.1.1), WC-bit, RPT-bit, S-bit Join
PIM(1): Send v2 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. O receptor entra online, envia PIM JOIN (*,G) para PE3.

2. PE3 encapsula o PIM JOIN (*,G) no pacote GRE e envia pelo túnel 2 (túnel MDT), que é verificado na Interface de Entrada 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. O PE2 recebeu o pacote GRE com Source como 3.3.3.3 e Destination 232.1.1.1 e o encaminha para o MVRF m-GRE com base no 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
```

O pacote GRE é desencapsulado e o PIM JOIN envia para o RP.

Note: O Vizinho de RPF é 2.2.2.2 porque o PIM Join é destinado ao endereço RP para formar o RPT através do núcleo.

Note: Bit WC e Bit RPT:Disparado pelo estado (*,G), o DR cria uma mensagem Join/Prune com o endereço RP em sua lista de junções e o bit curinga (bit WC) e bit RP-tree (bit RPT) definidos como 1. O bit WC indica que qualquer fonte pode corresponder e ser encaminhada de acordo com essa entrada se não houver mais correspondência; o bit RPT indica que essa junção é enviada para a árvore RP compartilhada. A lista de ameixas está vazia. Quando o bit RPT é definido como 1, ele indica que a junção está associada à árvore RP compartilhada e, portanto, a mensagem Join/Prune é propagada pela árvore RP. Quando o bit WC é definido como 1, indica que o endereço é um RP e que os receptores downstream esperam receber pacotes de todas as origens por meio desse caminho (árvore compartilhada).

```
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. Alcance de pacote encapsulado GRE no PE1 de origem.

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) alcança a Fonte CE.

3. Agora, a origem recebeu as informações do receptor interessado e o tráfego começa a enviar para o PE1 de origem.

4. Na origem 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

Em 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
```

Captura PCAP de pacote multicast de PE1. Túnel no túnel padrão MDT. Encapsulado com GRE.

5. No Receptor PE3, o pacote é recebido.

```
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
```

Observação: o vizinho RPF é alterado assim que um Pacote Multicast do PE1 é recebido. Anteriormente, era PE2 como RP hospedado atrás dele. Depois de receber o primeiro Pacote Multicast, ele altera o RPF e define o bit SPT.

Fluxo de tráfego sobre túnel MDT padrão:

- O encaminhamento no MDT usa GRE, o pacote C torna-se um pacote P.
- Endereço S do pacote P = endereço de peering BGP do PE
G address = MDT-Group address (Padrão ou Dados)
- O TOS do IP do pacote C será copiado para o pacote P.
- Os rótulos MPLS NÃO são usados no núcleo, somente multicast nativo.

Fluxo de pacote:

1. Um pacote C chega em uma interface PE configurada para VRF, o mVRF é identificado

implicitamente. Verificação RPF normal na fonte C.

O pacote C replicou a interface de saída no OIL. Nesse ponto, essa seria uma interface PE no mesmo 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 o OIL contiver um MTI, o pacote C será encapsulado em um pacote P. Se o sinalizador "y" estiver definido no destino da entrada usado é grupo DATA-MDT, caso contrário, grupo padrão MDT. A origem é o endereço do peer do BGP do PE e o destino é o endereço do grupo MDT.

2. O pacote P é encaminhado através da rede P como por multicast normal.

O pacote chega à interface global. Entrada global (S,G) ou (*,G) para o grupo MDT referenciado. Verificação RPF normal em P-Source (PE Peer).

3. O pacote P é replicado na interface OIL. Neste ponto, este é P/PE na tabela de mroute global.

4. Se o flag "Z" definir o pacote, ele será desencapsulado para revelar o Pacote C. O mVRF de destino e a interface de entrada derivada do grupo MDT é o destino do cabeçalho encapsulado.

Verificação RPF do Pacote C em mVRF concluída, Pacote C replicado em OIL em 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,

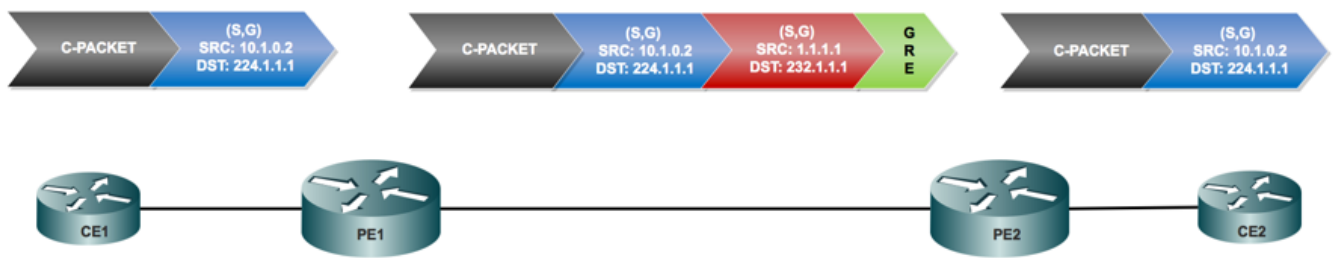
(1.1.1.1, 232.1.1.1), 1d01h/00:02:47, flags: **JTZ**

Incoming interface: Ethernet0/3, RPF nbr 10.0.3.2

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

5. Alcance do pacote C nativo no receptor 3.

Encapsulamento de pacotes:

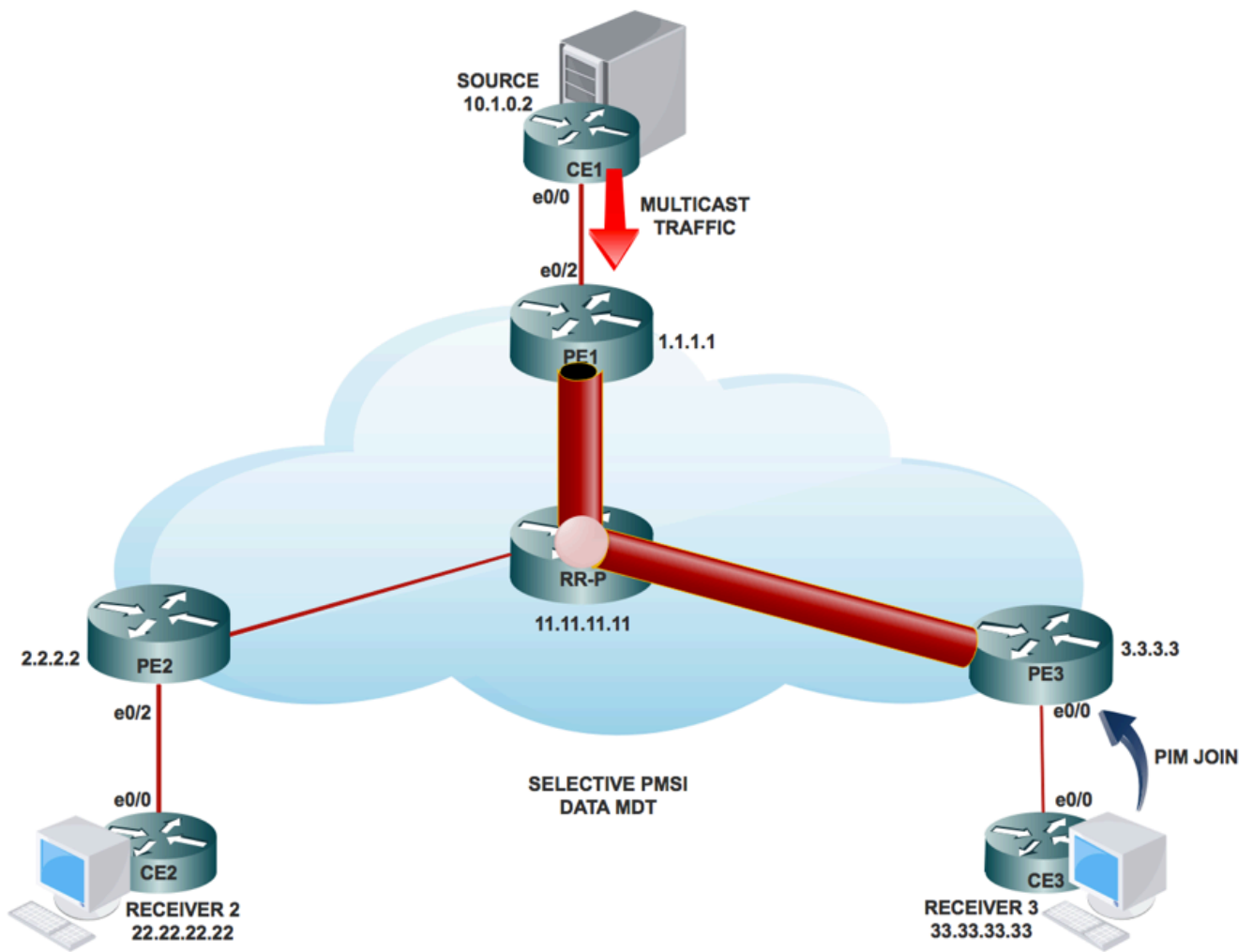


MDT de dados:

O que é MDT de dados?

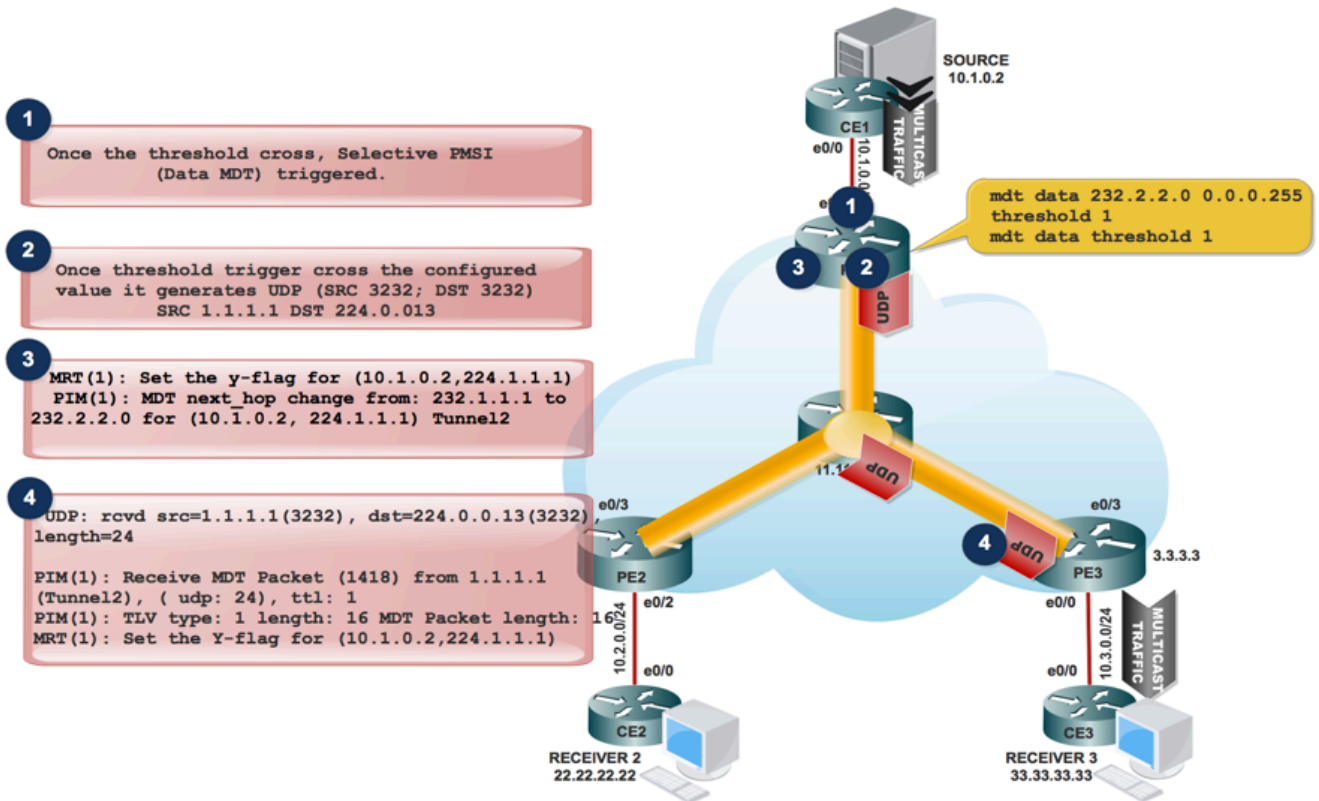
É opcional. Ele é criado sob demanda, transporta tráfego específico (S,G). Na última versão do IOS®, o limite configurado é "0" e "infinito". Sempre que um primeiro pacote atinge o VRF, o MDT de dados é inicializado e, se infinito, o MDT de dados nunca será criado, e o tráfego avança no MDT padrão. O MDT de dados é sempre a árvore de recebimento, eles nunca enviam nenhum tráfego. O MDT de dados é apenas para o tráfego (S,G).

PMSI seletiva:



- É opcional. Ele é criado sob demanda, transporta tráfego específico (S, G).
- Sempre que um primeiro pacote atinge o VRF, o MDT de dados é inicializado e, se infinito, o MDT de dados nunca é criado, e o tráfego avança no MDT padrão.
- O MDT de dados é sempre a árvore de recebimento, eles nunca enviam nenhum tráfego. O MDT de dados é apenas para o tráfego (S, G).
- A mensagem PIM transporta C- (S, G) e P-Group.

Como o MDT de dados é criado:



1. Quando o tráfego multicast entra no VRF e quando a taxa de tráfego atinge o limite. Gera um pacote MDT.
2. O pacote MDT é encapsulado em UDP com Source e Destination 3232. E enviá-lo para o receptor interessado.

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. Depois de enviar o pacote UDP ao receptor interessado, ele define "y" flag e altera o próximo salto MDT para o novo endereço do grupo MDT.

Na origem 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: O salto seguinte do OIL muda para 232.2.2.0.

2. AT PE3, quando recebe o pacote MDT encapsulado na porta UDP SRC 3232 e na 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: TYp

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

Outgoing interface list:

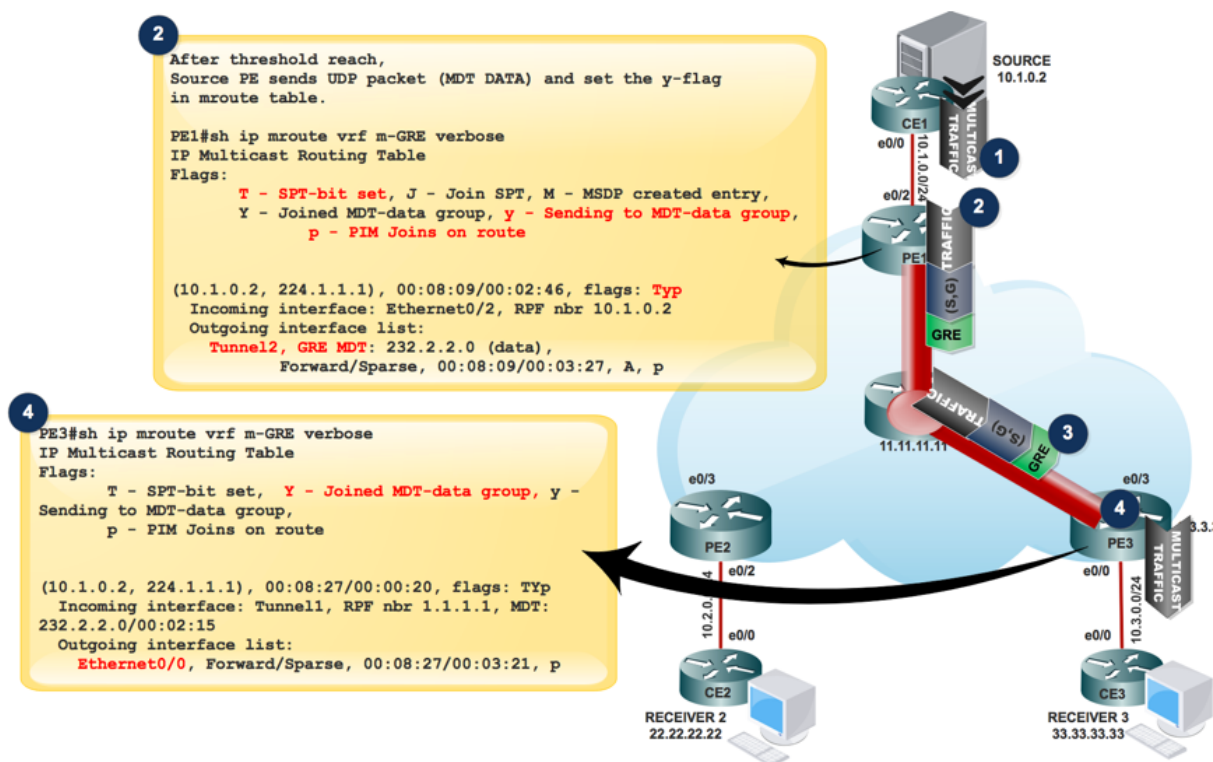
Ethernet0/0, Forward/Sparse, 00:08:27/00:03:21, p

A mensagem S-PMSI Join é uma mensagem encapsulada em UDP cujo endereço de destino é ALL-PIM-ROUTERS (224.0.0.13) e cuja porta de destino é 3232.

A mensagem S-PMSI Join contém estas informações: Um identificador para o fluxo multicast específico que deve ser vinculado ao túnel P. Isso pode ser representado como um par (S,G). Um identificador para o túnel P específico ao qual o fluxo deve ser associado. Este identificador é um

campo estruturado que inclui estas informações:

Fluxo de tráfego multicast no túnel de dados 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 o OIL contiver uma interface de túnel, o pacote será encapsulado com o uso do GRE, sendo a origem o endereço de peering BGP do roteador PE local e o destino o endereço do grupo MDT.
- A decisão que o grupo Data-MDT é selecionado depende se o sinalizador y é definido na entrada (S, G) no mVRF.
- Se a entrada (S, G) ou (*, G) tiver o sinalizador Z definido, então este é um padrão ou um Data-MDT com um mVRF associado.

- O pacote P deve ser desencapsulado para revelar o pacote 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

- Como apenas um MTI único existe no mVRF por domínio multicast, tanto o Data-MDT quanto o Default-MDT usam a mesma interface de túnel para o tráfego do cliente.
- Os flags Y/y são necessários para distinguir o tráfego Default-MDT do tráfego Data-MDT e garantir que as entradas de roteamento multicast do cliente usem o grupo MDT-Data correto e se referem a uma tabela interna que contém os mapeamentos (S, G, Data-MDT).

Troubleshoot

Atualmente, não existem informações disponíveis específicas sobre Troubleshooting para esta configuração.