

次世代マルチキャストデフォルトMDT:プロファイル0

内容

[概要](#)

[背景説明](#)

[オーバーレイシグナリングとしてのPIM](#)

[設定作業](#)

[設定](#)

[確認](#)

[トラブルシューティング](#)

概要

このドキュメントでは、次世代マルチキャストのマルチプロトコルラベルスイッチ(MPLS)コアを使用してマルチキャストパケットが通過する方法について説明します。

背景説明

デフォルトMDT - PIM C - mcastシグナリング

Draft RosenはGeneric Routing Encapsulation(GRE)をオーバーレイプロトコルとして使用します。つまり、すべてのマルチキャストパケットがGRE内でカプセル化されます。仮想LANは、VPN内のすべてのプロバイダーエッジ(PE)ルーターでエミュレートされ、マルチキャストグループに参加します。これは、デフォルトのマルチキャスト配信ツリー(MDT)と呼ばれます。デフォルトのMDTは、Protocol Independent Multicast(PIM)helloおよびその他のPIMシグナリングに使用されますが、データトラフィックにも使用されます。送信元が大量のトラフィックを送信する場合、デフォルトのMDTを使用することは非効率的であり、データMDTを作成できます。データMDTには、使用中のグループのレシーバを持つPEだけが含まれます。

Draft Rosenは導入が非常に簡単で、うまく動作しますが、いくつかの欠点があります。次の点を見てみましょう。

オーバーヘッド:GREはパケットに24バイトのオーバーヘッドを追加します。通常8または12バイトを追加するMPLSと比較すると、各パケットに100%以上のオーバーヘッドが追加されます。

コア内のPIM:Draft Rosenでは、PEがデフォルトに参加する必要があるため、PIMがコア内で有効になっている必要があります。また、PIMシグナリングによって実行されるデータMDTも必要です。コアでPIM ASMが使用されている場合は、RPも必要です。PIM SSMがコアで実行されている場合、RPは必要ありません。

Core state:PEからのPIMシグナリングにより、コアに不要な状態が作成されます。コアの状態は可能な限り小さくする必要があります。

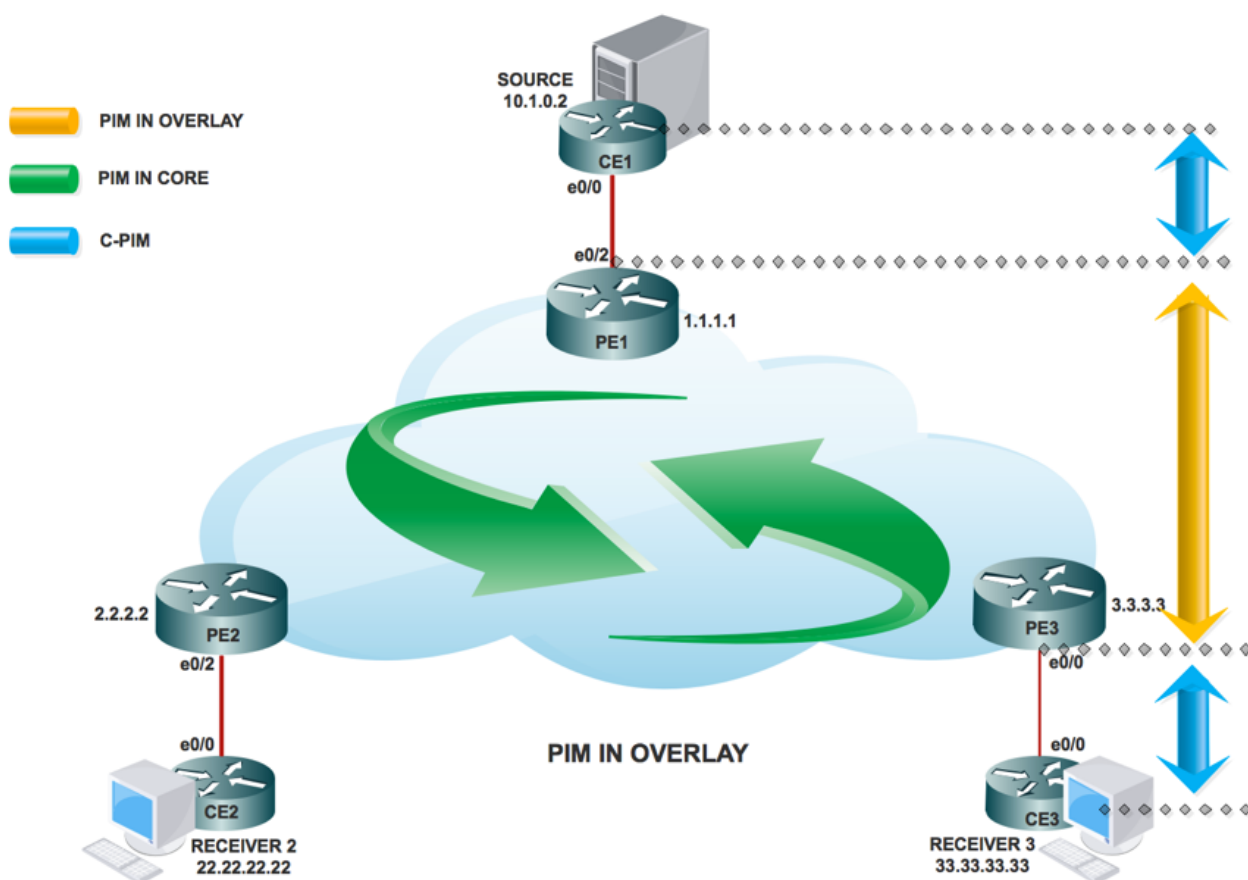
PIM隣接関係: PEは相互にPIMネイバーになります。大規模なVPNで多数のPEが存在する場合は、多数のPIM隣接関係が作成されます。これにより、helloおよびその他のシグナリングが大量に生成され、ルータの負荷が増大します。

ユニキャストとマルチキャスト: ユニキャスト転送はMPLSを使用し、マルチキャストはGREを使用します。これは複雑さを増し、ユニキャストがマルチキャストとは異なる転送メカニズムを使用することを意味します。これは最適なソリューションではありません。

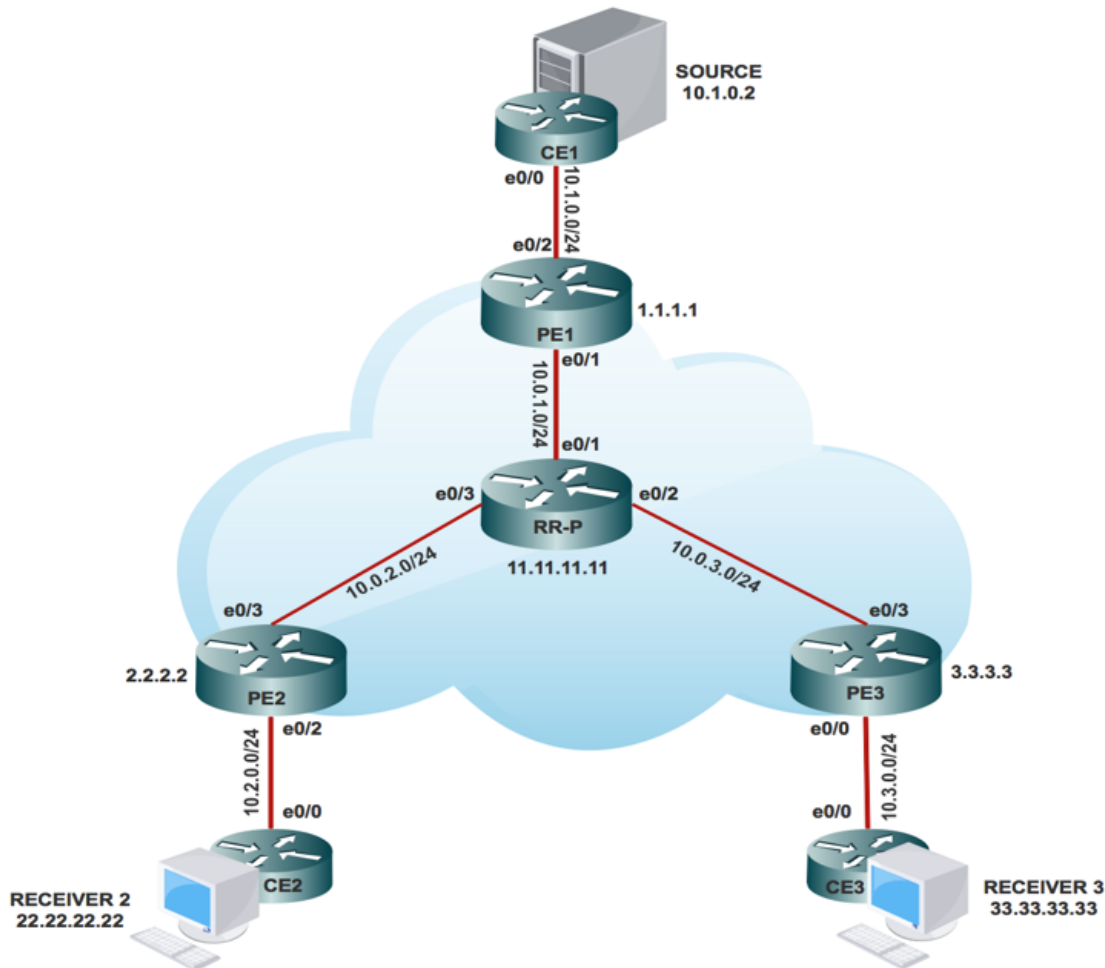
非効率性: デフォルトのMDTは、PEが(*,G)または(S,G)のグループで使用されているレシーバを持っているかどうかに関係なく、VPN内のすべてのPEにトラフィックを送信します。

- デフォルトのMDTは、1つのVRF内のすべてのPEにマルチキャストを接続するために使用されます。
- デフォルトは、すべてのPEルータを接続することを意味します。
- デフォルトでは、すべてのトラフィックが伝送されます。
- すべてのPIM制御トラフィックとデータプレーントラフィック。例：(*,G)トラフィックと(S,G)トラフィック
- これはマルチポイント間を表します。
- 誰でも送信でき、誰でもツリーから受信できます。

オーバーレイシグナリングとしてのPIM



トポロジ



設定作業

1. すべてのノードでマルチキャストルーティングを有効にします。
2. すべてのインターフェイスでPIMスパスモードを有効にします。
3. 既存のVRFでデフォルトMDTを設定します。
4. インターフェイスEthernet0/xにVRFを設定します。
5. VRFでマルチキャストルーティングを有効にします。
6. コア内のすべてのノードでPIM SSM Defaultを設定します。
7. CEノードでBSR RPを設定します。
8. 事前設定：

- VRF m-GRE
- MBGP:アドレスファミリVPNv4
- VRFルーティングプロトコル

設定

1. すべてのノードでマルチキャストルーティングを有効にします。

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

2. すべてのインターフェイスでPIMスパスモードを有効にします。

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

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

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

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

3.すでに存在するVRFで、デフォルトMDTを設定します。

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

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

4. インターフェイスEthernet0/xにVRFを設定します。

PE1、PE2、およびPE3上。

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

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

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

5. VRFでマルチキャストルーティングを有効にします。

PE1、PE2、およびPE3上。

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

6. サービスプロバイダーコアのRPを設定します。

PE1、PE2、PE3、およびRR-Pノード。

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

7. CEノード (レシーバ) でBSR RPを設定します。

Receiver2上。

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

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

確認

ここでは、設定が正常に機能しているかどうかを確認します。

タスク 1：物理的な接続を確認する

- 接続されているすべてのインターフェイスが「UP」であることを確認します。

タスク 2 : アドレスファミリVPNv4ユニキャストの確認

- AF VPNv4ユニキャストのすべてのルータでBGPが有効になっており、BGPネイバーが「UP」であることを確認します
- BGP VPNv4ユニキャストテーブルにすべてのCustomerプレフィックスがあることを確認します。

タスク 3 : マルチキャストトラフィックのエンドツーエンドを確認します。

- PIMネイバーシップをチェックします。
- マルチキャスト状態がエンドツーエンドで作成されることを確認します。
- PE1、PE2、およびPE3のmRIBエントリの確認
- (S,G)mFIBエントリがソフトウェアフォワーディングで増加することを確認します。
- ICMPパケットがCEからCEに到達することを確認します。

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

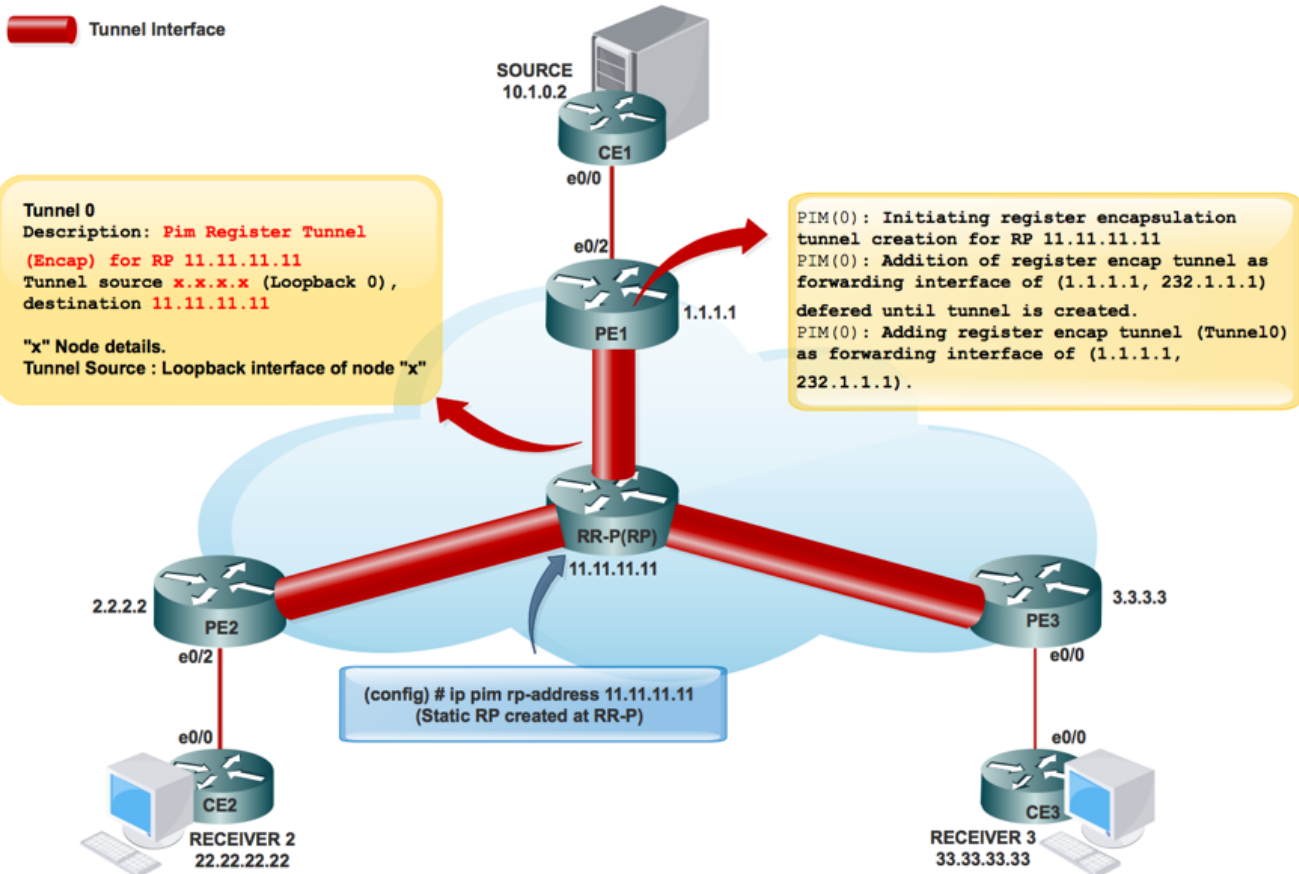
Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 230.1.1.1, timeout is 0
seconds:

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
```

トンネルインターフェイスが作成されたとき :

サービスプロバイダーRPの作成 :

RP情報がコアにフラッディングされると、Interface Tunnel 0が作成されます。



PIM(0):RP 11.11.11.11のレジスタカプセル化トンネルの作成を開始しています。

PIM(0):RP 11.11.11.11の初期レジスタトンネルの作成に成功しました。

PIM(0):(1.1.1.1, 232.1.1.1)の転送インターフェイスとしてのregister encapトンネルの追加は、トンネルが作成されるまで延期されます。

May 9 17:34:56.155:PIM(0):RP 11.11.11.11(, 232.1.1.1)をチェックします。

PIM(0):(1.1.1.1, 232.1.1.1)の転送インターフェイスとしてregister encapトンネル(Tunnel0)を追加します。

```
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

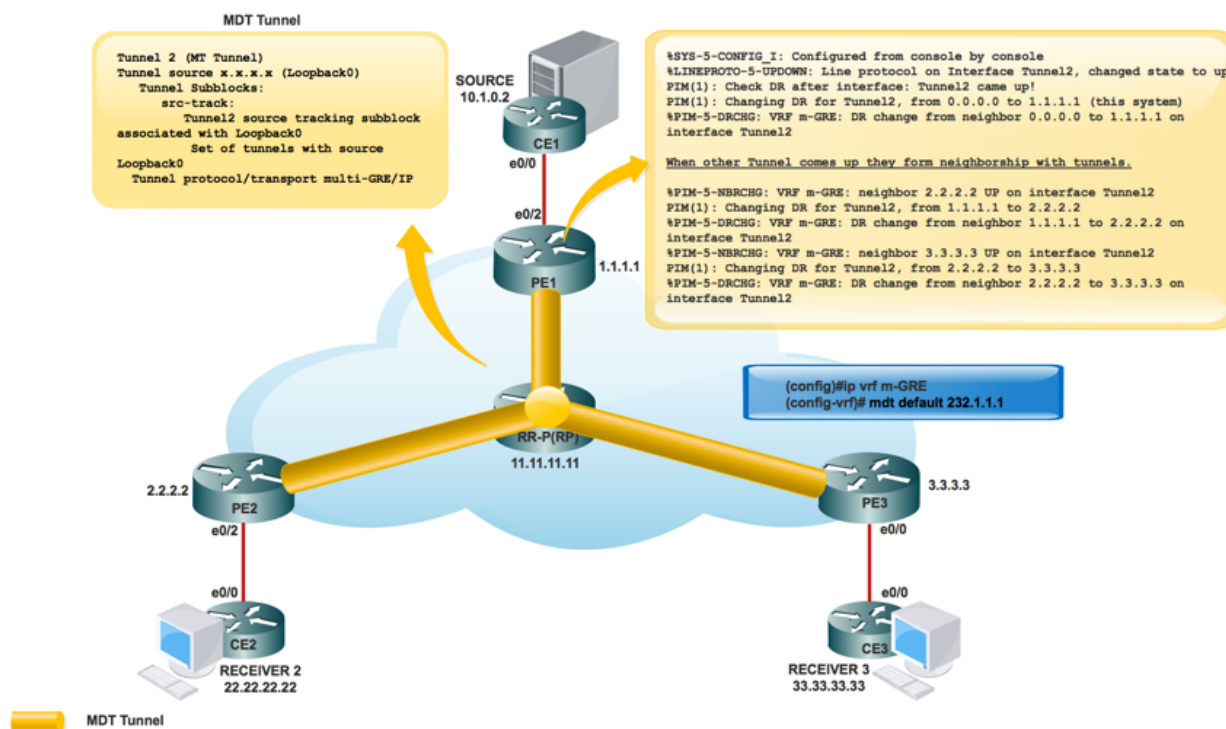
Set of tunnels with source Ethernet0/1, 1 member (includes iterators), on interface <OK>

Tunnel protocol/transport PIM/IPv4

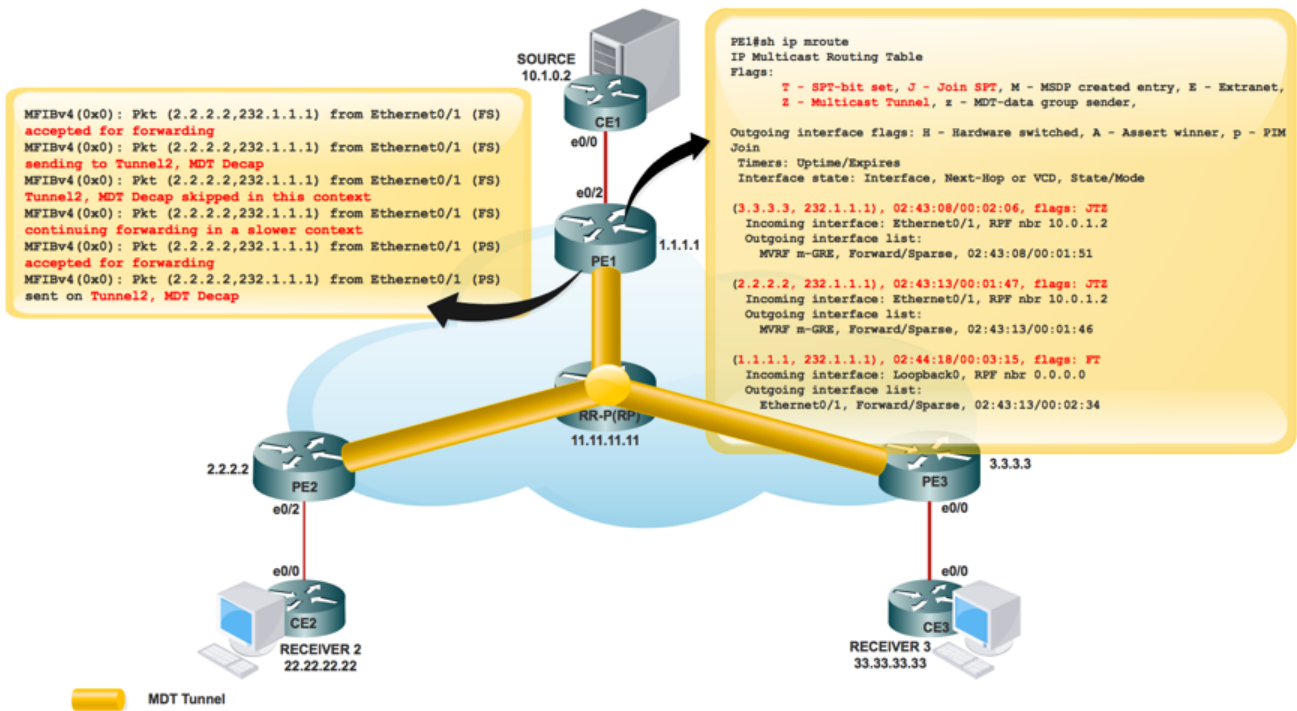
Tunnel TOS/Traffic Class 0xC0, Tunnel TTL 255

Tunnel transport MTU 1472 bytes

MDTトンネルの作成 :



コアでのMRIBの作成 :



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:

MVRF 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:

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

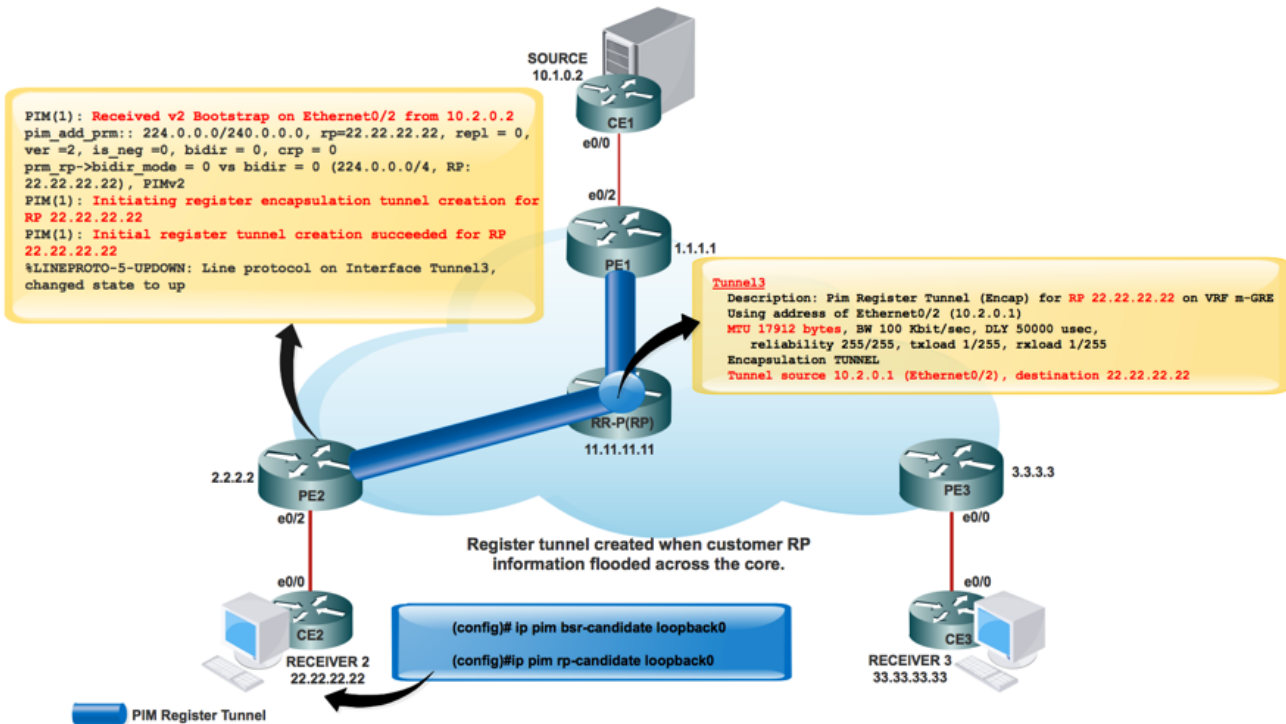
(1.1.1.1, 232.1.1.1), 00:10:15/00:03:20, flags: FT

Incoming interface: Loopback0, RPF nbr 0.0.0.0

Outgoing interface list:

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

顧客のネットワーク用にRPを作成したら、次の手順を実行します。



*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

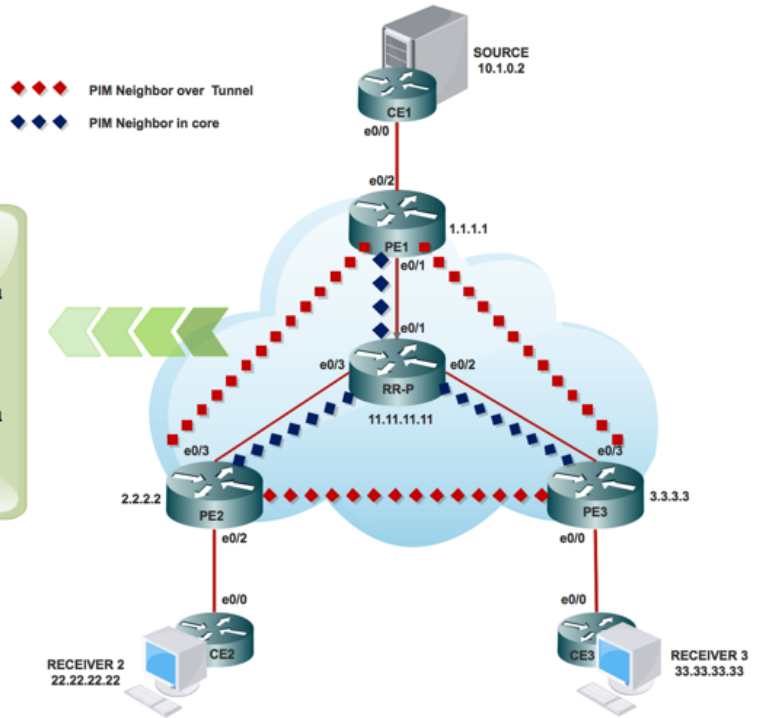
顧客のRP情報を伝送するために、トンネルインターフェイスが作成されます。

PIM(1):RP 22.22.22.22のレジスタカプセル化トンネルの作成を開始しています。

これは、RPにカプセル化を登録するために作成されたトンネルです。

検出されたスパスモードRPごとに、1つのRegisterカプセル化トンネルが作成されます。スパスモードRP自体には、レジスタパケットを受信するために作成された1つのカプセル化解除トンネルインターフェイスがあります。

PIMネイバーシップ:



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

パケットフローを次に示します。

コントロールプレーンパケットフローは2つの部分に分割されます。

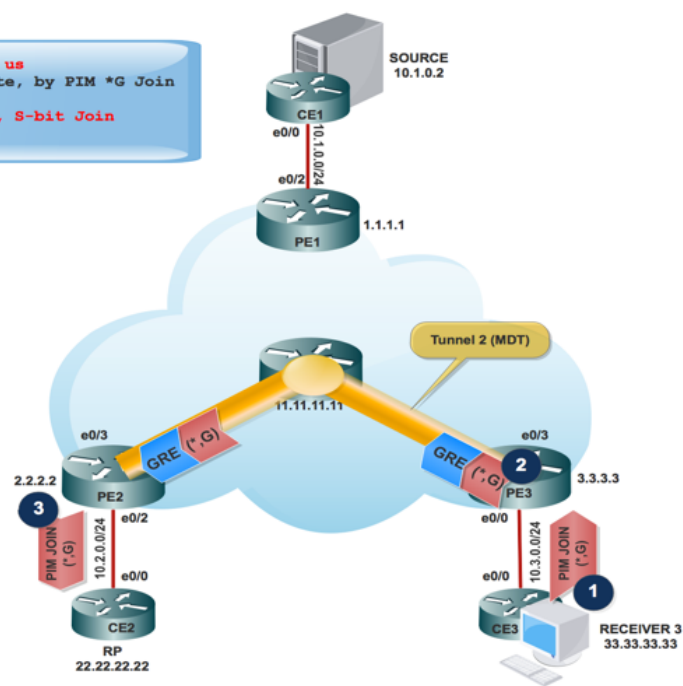
1. レシーバがオンラインになります。

2. ソースがアクティブです。
レシーバがアクティブな場合：

```
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. レシーバがオンラインになり、PIM JOIN (*,G)をPE3に向けて送信します。
2. PE3はPIM JOIN(*,G)をGREパケットにカプセル化し、トンネル2 (MDTトンネル) を介して送信します。これはshow ip mroute vrf m-GREの着信インターフェイスから確認されます。

```
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は、送信元が3.3.3.3、宛先が232.1.1.1のGREパケットを受信し、OILに基づいてMVRF m-GREに転送します。

```
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
```

GREパケットがカプセル化解除され、PIM JOINがRPに送信されます。

注：RPFネイバーは2.2.2.2です。これは、PIM Joinがコアを介してRPTを形成するRPアドレス宛てであるためです。

注：WCビットとRPTビット：(*,G)状態によってトリガーされ、DRは参加リストにRPアドレスを含むJoin/Pruneメッセージを作成し、ワイルドカードビット(WCビット)とRPツリービット(RPTビット)を1に設定します。一致するものがない場合、RPTビットは、この加入が共有RPツリーに送信されることを示します。プルーンリストは空のままです。RPTビットが1に設定されている場合は、参加が共有RPツリーに関連付けられていることを示します。したがって、Join/PruneメッセージがRPツリーに沿って伝搬されます。WCビットが1に設定されている場合は、アドレスがRPであり、ダウンストリームレシーバはこの(共有ツリー)パスを介してすべてのソースからパケットを受信することを想定しています。

```
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. 送信元PE PE1でのGREカプセル化パケット到達。

```
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)がソースCEで到達します。

3. これで送信元はInentiated Receiverの情報を取得し、トラフィックは送信元PE PE1への送信を開始します。

4. 送信元PE 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

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
```

PE1からのマルチキャストパケットのPCAPキャプチャ。MDTデフォルトトンネルでトンネリングされます。GREでカプセル化。

5. レシーバPE PE3では、パケットが受信されます。

```
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
```

注:PE1からのマルチキャストパケットが受信されると、RPFネイバーが変更されます。以前は、背後でホストされるRPとしてPE2でした。最初のマルチキャストパケットを受信すると、RPFが変更され、SPTビットが設定されます。

デフォルトMDTトンネル上のトラフィックフロー：

- MDTでの転送はGREを使用し、CパケットはPパケットになります。
- P-Packet S address = PEのBGPピアリングアドレス
Gアドレス= MDTグループアドレス (デフォルトまたはデータ)
- C-Packet IP TOSはP-Packetにコピーされます。
- MPLSラベルはコアでは使用されず、ネイティブマルチキャストでのみ使用されます。

パケット フローを次に示します。

1. CパケットがVRF設定のPEインターフェイスに到着すると、mVRFは暗黙的に識別されます。Cソースの通常のRPFチェック。

C-PacketはOILのインターフェイスから複製されます。この時点で、これは同じVRF内のPEインターフェイスになります。

```
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)
```

OILにMTIが含まれている場合、CパケットはPパケットにカプセル化されます。使用するエントリの宛先に"y"フラグが設定されている場合、DATA-MDTグループが設定されていない場合、デフォルトMDTグループが設定されます。送信元はPE BGPピアアドレスで、宛先はMDTグループアドレスです。

2. Pパケットは、通常のマルチキャストに従ってPネットワーク経由で転送されます。パケットがグローバルインターフェイスに到着します。参照されているMDTグループのグローバル(S、G)または(*、G)エントリ。Pソース (PEピア) の通常のRPFチェック。

3. PパケットはOILのインターフェイスから複製されます。この時点で、これはグローバルmrouteテーブルのP/PEです。

4. 「Z」フラグが設定されている場合、パケットはカプセル化解除されてCパケットが表示されます。MDTグループから派生したターゲットmVRFおよび着信インターフェイスは、カプセル化されたヘッダーの宛先です。

mVRFでのCパケットのRPFチェックが完了し、mVRFでCパケットがOILに複製されます。

```
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,

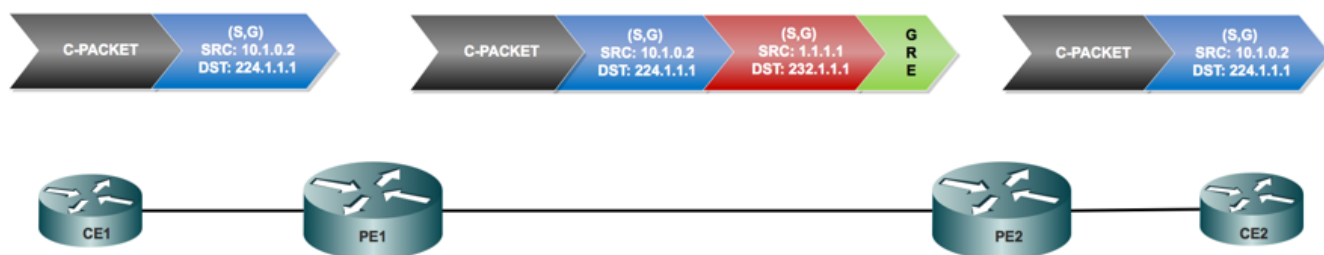
Z - Multicast Tunnel, z - MDT-data group sender,

(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. レシーバ3でのネイティブCパケットの到達。
パケットカプセル化:

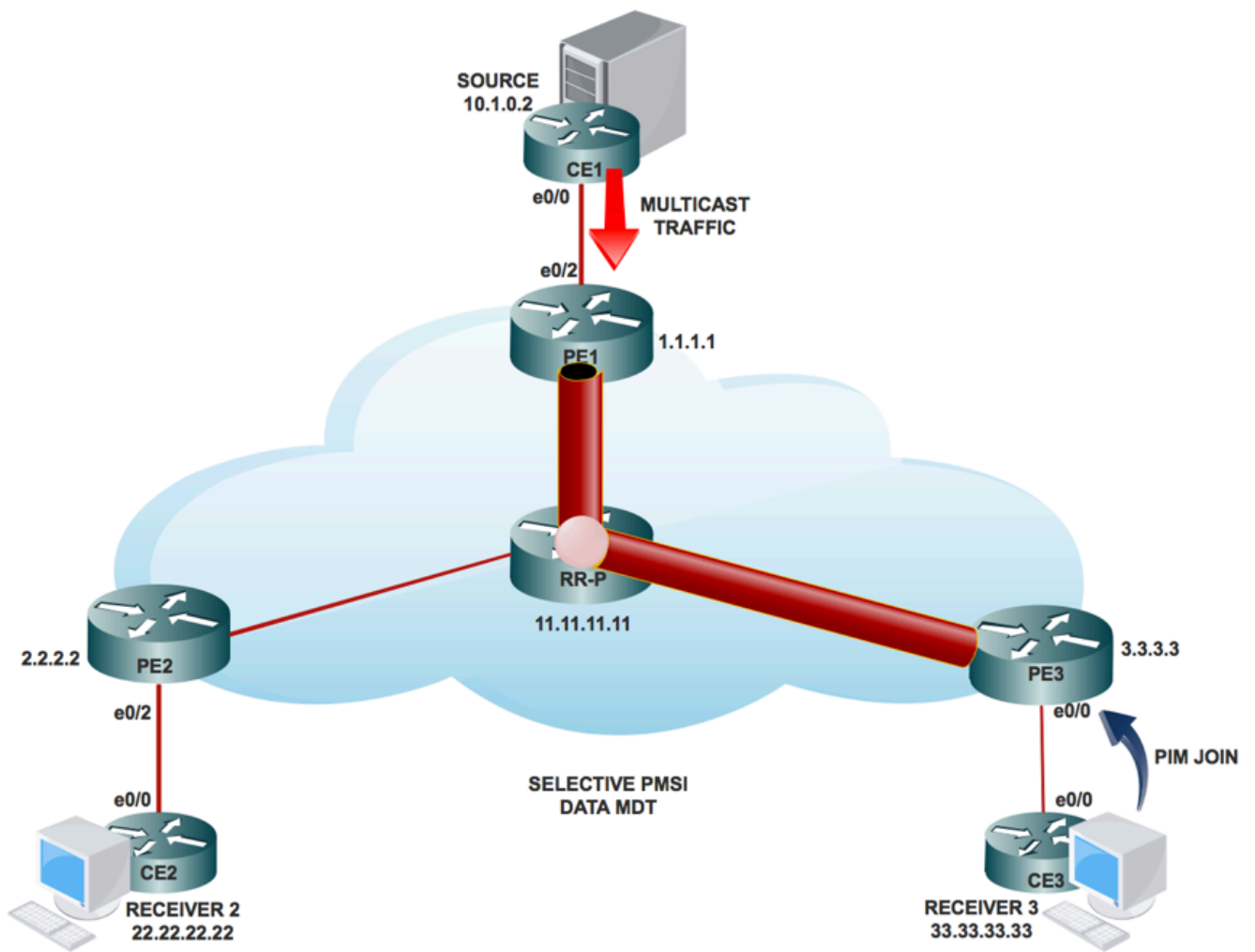


データMDT:

データMDTとは

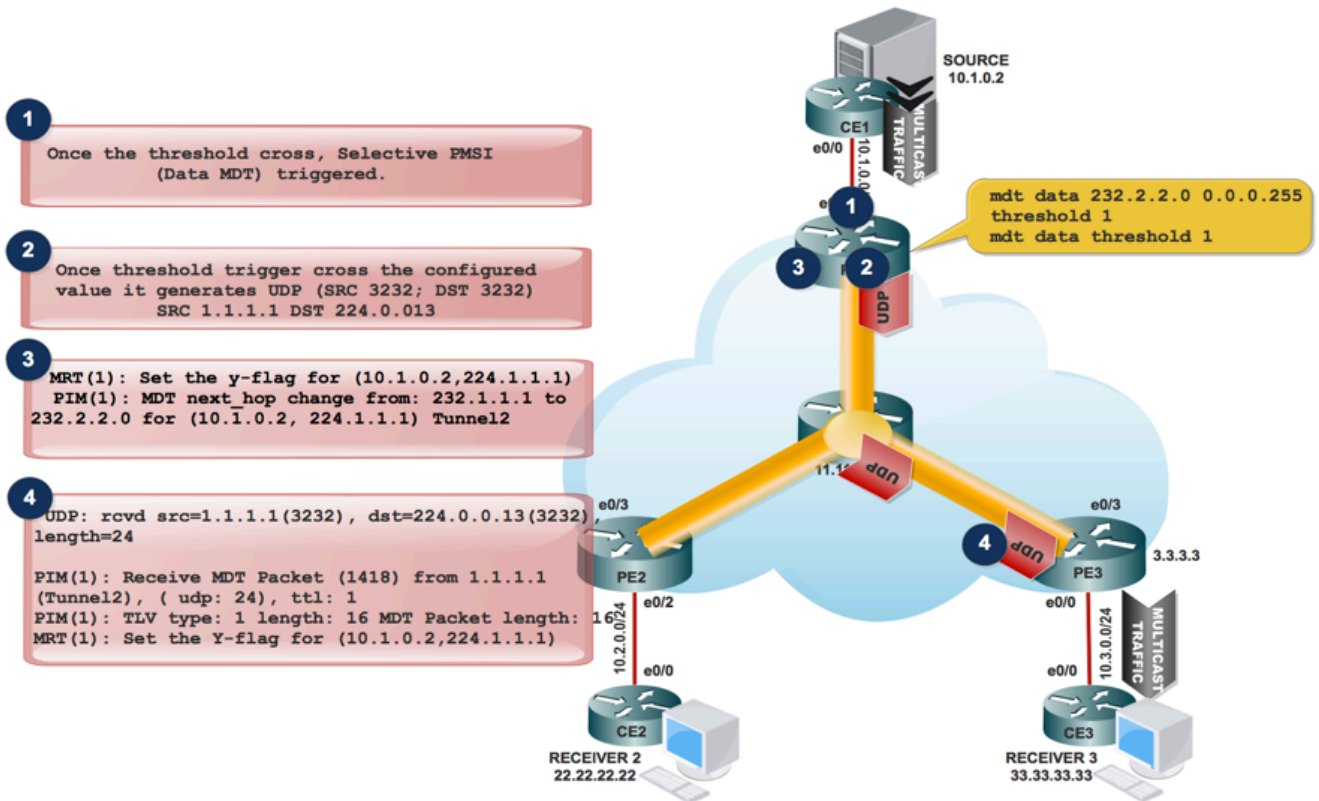
オプションです。需要に応じて作成され、特定の(S,G)トラフィックを伝送します。最新のIOS®リリースでは、設定されているしきい値は「0」および「無限」です。最初のパケットがVRFに到達すると、データMDTが初期化され、無限の場合はデータMDTは作成されず、トラフィックはデフォルトのMDTに進みます。データMDTは常に受信側ツリーであり、トラフィックを送信することはありません。データMDTは(S,G)トラフィック専用です。

選択的PMSI:



- オプションです。需要に応じて作成され、特定の(S、G)トラフィックを伝送します。
- 最初のパケットがVRFに到達すると、データMDTが初期化され、無限の場合はデータMDTは作成されず、トラフィックはデフォルトのMDTに進みます。
- データMDTは常に受信側ツリーであり、トラフィックを送信することはありません。データMDTは(S、G)トラフィック専用です。
- PIMメッセージはC-(S,G)およびP-Groupを伝送します。

データMDTの作成方法：



1. マルチキャストトラフィックがVRFに入り、トラフィックレートがしきい値に達したとき。MDTパケットを生成します。
2. MDTパケットは、送信元と宛先が3232のUDPにカプセル化されます。興味のある受信者に送信します。

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. UDPパケットを対象のレシーバに送信した後、「y」フラグを設定し、MDT next_hopを新しいMDTグループアドレスに変更します。

送信元PE 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: Typ

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)

注：OILのネクストホップが232.2.2.0に変更されます。

2. AT PE3では、UDP SRCポート3232およびDSTポート3232でカプセル化されたMDTパケットを受信します。

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

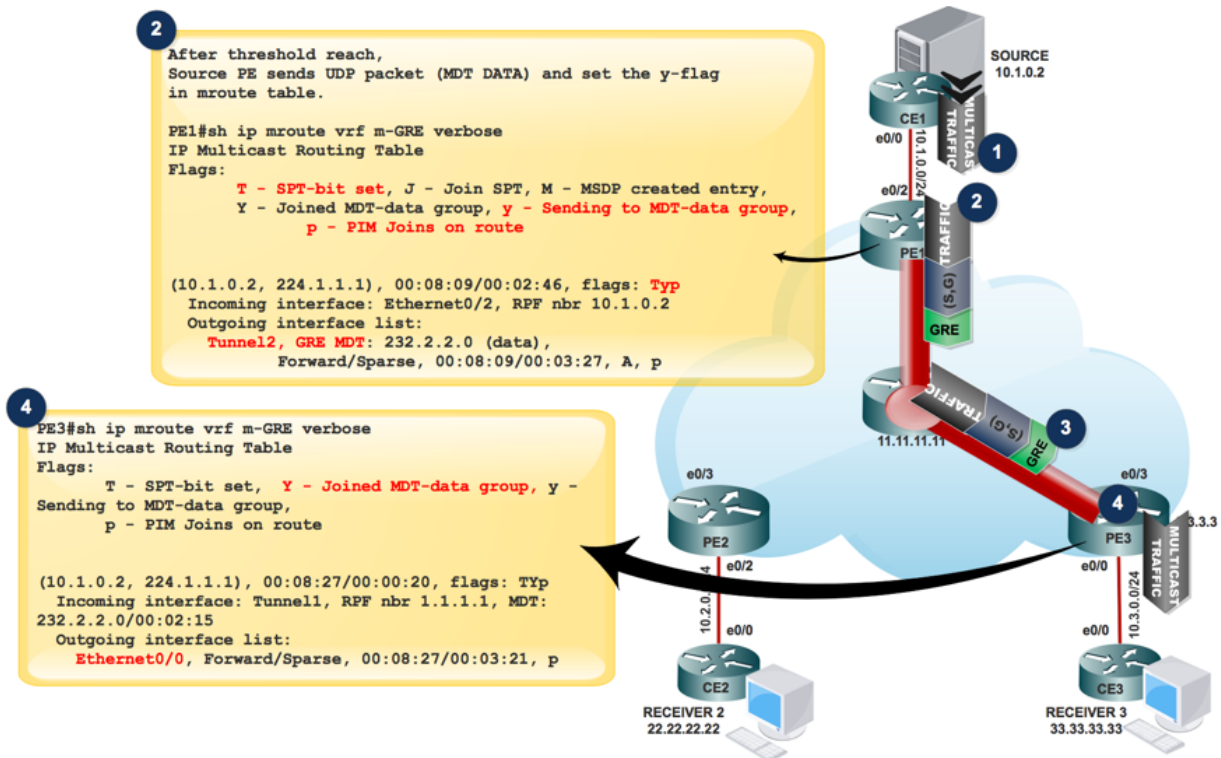
Outgoing interface list:

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

S-PMSI Joinメッセージは、宛先アドレスがALL-PIM-ROUTERS(224.0.0.13)で、宛先ポートが3232のUDPカプセル化メッセージです。

S-PMSI Joinメッセージには、次の情報が含まれます。Pトンネルにバインドされる特定のマルチキャストストリームの識別子。これは(S,G)ペアとして表すことができます。ストリームがバインドされる特定のPトンネルの識別子。この識別子は、次の情報を含む構造化フィールドです。

MDT DATAトンネルのマルチキャストトラフィックフロー :



```
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
```

- OILにトンネルインターフェイスが含まれている場合、パケットはGREを使用してカプセル化されます。送信元はローカルPEルータのBGPピアリングアドレス、宛先はMDTグループアドレスです。
- Data-MDTグループが選択される決定は、mVRFの(S, G)エントリにyフラグが設定されているかどうかによって異なります。
- (S, G)または(*, G)エントリにZフラグが設定されている場合、これは関連付けられたmVRFを持つデフォルトまたはデータMDTです。
- Cパケットを明らかにするには、Pパケットをカプセル解除する必要があります。

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

- マルチキャストドメインごとのmVRFには1つのMTIしか存在しないため、Data-MDTとDefault-MDTの両方が顧客のトラフィックに同じトンネルインターフェイスを使用します。
- Y/yフラグは、Default-MDTトラフィックとData-MDTトラフィックを区別し、カスタマーマルチキャストルーティングエントリが正しいMDT-Dataグループを使用し、(S、G、Data-MDT)マッピングを保持する内部テーブルを参照するために必要です。

トラブルシューティング

現在、この設定に関する特定のトラブルシューティング情報はありません。