

Multicast-Standard-MDT der nächsten Generation: Profil 0

Inhalt

[Einführung](#)

[Hintergrundinformationen](#)

[PIM als Overlay Signaling](#)

[Konfigurationsaufgaben](#)

[Konfigurieren](#)

[Überprüfen](#)

[Fehlerbehebung](#)

Einführung

In diesem Dokument wird beschrieben, wie Multicast-Pakete mit dem Multiprotocol Label Switching (MPLS)-Core im Multicast der nächsten Generation übertragen werden.

Hintergrundinformationen

Standard-MDT - PIM C - Multicast-Signalisierung

Draft Rosen verwendet Generic Routing Encapsulation (GRE) als Overlay-Protokoll. Das bedeutet, dass alle Multicast-Pakete innerhalb der GRE gekapselt sind. Ein virtuelles LAN wird mit allen Provider Edge (PE)-Routern in einer Multicast-Gruppe emuliert. Dies wird als Standard-Multicast Distribution Tree (MDT) bezeichnet. Der Standard-MDT wird für Protocol Independent Multicast (PIM)-Hellos und andere PIM-Signalisierungen, aber auch für den Datenverkehr verwendet. Wenn die Quelle viel Datenverkehr sendet, ist es ineffizient, den Standard-MDT zu verwenden, und es kann ein Daten-MDT erstellt werden. Der Daten-MDT umfasst nur PEs, die über Empfänger für die verwendete Gruppe verfügen.

Rosen-Entwurf ist relativ einfach bereitzustellen und funktioniert gut, aber es hat einige Nachteile. Sehen wir uns die folgenden Punkte an:

Overhead - GRE fügt dem Paket 24 Byte Overhead hinzu. Im Vergleich zu MPLS, das normalerweise 8 oder 12 Byte hinzufügt, werden jedem Paket mindestens 100 % des Overhead hinzugefügt.

PIM im Core - Draft Rosen erfordert, dass PIM im Core aktiviert ist, da die PEs dem Standard- und/oder Daten-MDT beitreten müssen, der durch PIM-Signalisierung erfolgt. Wenn im Core PIM ASM verwendet wird, ist auch ein RP erforderlich. Wenn PIM SSM im Core ausgeführt wird, ist kein RP erforderlich.

Core-Status: Im Core wird ein unnötiger Zustand aufgrund der PIM-Signalisierung von den PEs erstellt. Der Core sollte so wenig Zustand wie möglich aufweisen.

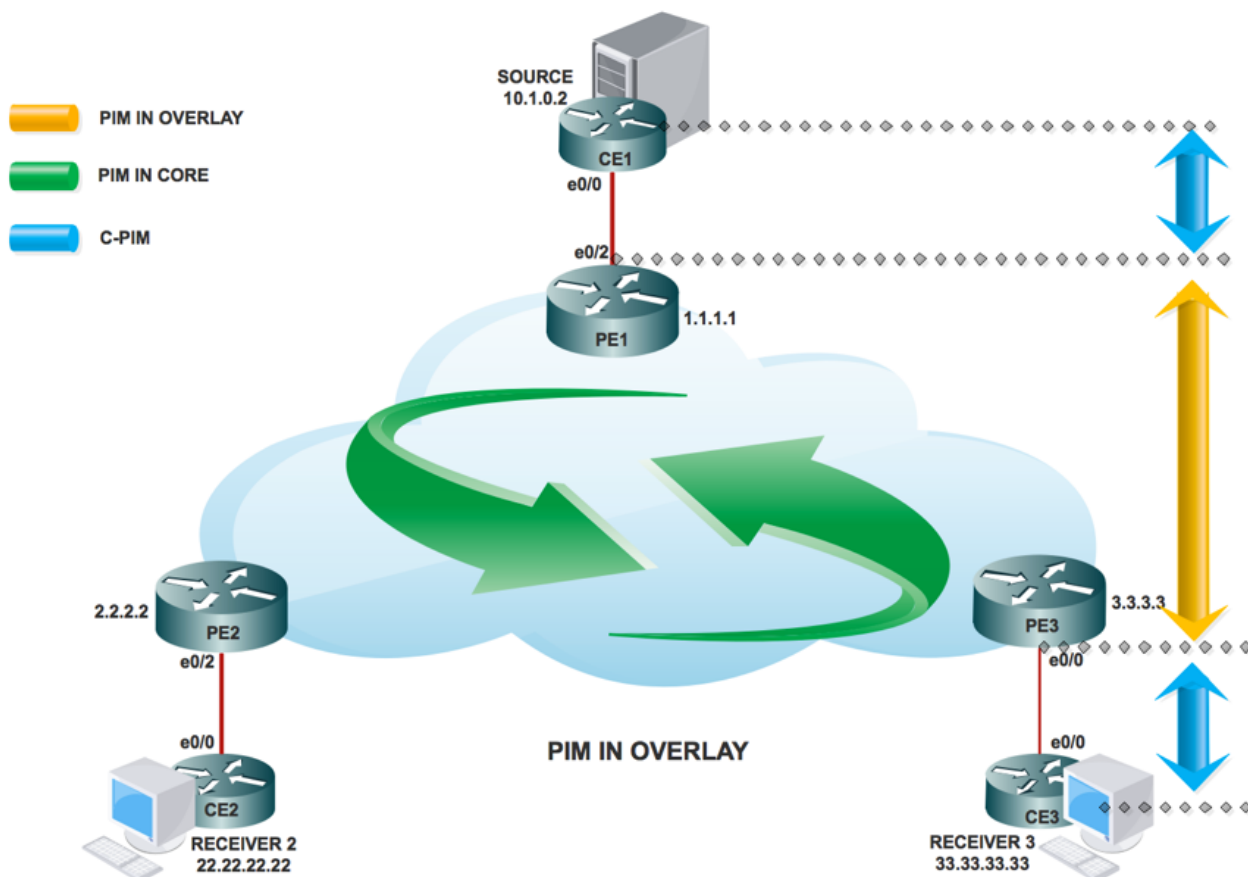
PIM-Adjacencies: Die PEs werden zu PIM-Nachbarn. Wenn es sich um ein großes VPN und viele PEs handelt, werden viele PIM-Adjacencies erstellt. Dies führt zu einer Vielzahl von Hello- und anderen Signalisierungen, die den Router zusätzlich belasten.

Unicast vs. Multicast - Die Unicast-Weiterleitung verwendet MPLS, Multicast verwendet GRE. Dies erhöht die Komplexität und bedeutet, dass Unicast einen anderen Weiterleitungsmechanismus verwendet als Multicast, was nicht die optimale Lösung darstellt.

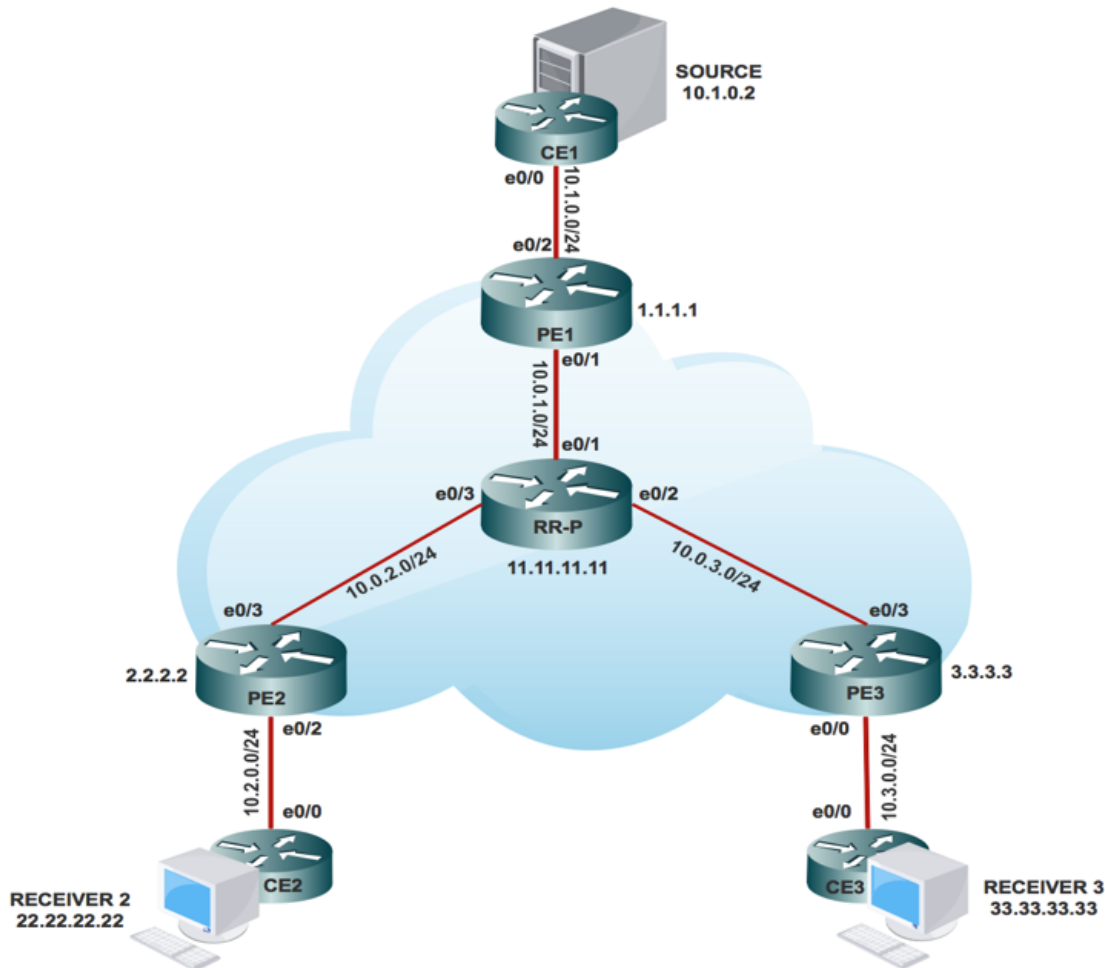
Ineffizienz - Der Standard-MDT sendet Datenverkehr an alle PEs im VPN, unabhängig davon, ob der PE über einen Empfänger in der (*,G) oder (S,G) für die verwendete Gruppe verfügt.

- Der Standard-MDT wird verwendet, um Multicast mit allen PEs in einem VRF zu verbinden.
- Standard bedeutet, dass alle PE-Router miteinander verbunden werden.
- Standardmäßig wird der gesamte Datenverkehr übertragen.
- Der gesamte PIM-Kontrollverkehr und der Datenverkehr auf Datenebene. z. B. (*,G)-Datenverkehr und (S,G)-Datenverkehr.
- Dies stellt Multipoint-zu-Multipoint dar.
- Jeder kann senden und jeder kann von dem Baum empfangen.

PIM als Overlay Signaling



Topologie



Konfigurationsaufgaben

1. Aktivieren Sie Multicast-Routing auf allen Knoten.
2. Aktivieren Sie PIM Sparse Mode auf der gesamten Schnittstelle.
3. Mit vorhandenem VRF konfigurieren Sie einen Standard-MDT.
4. Konfigurieren Sie die VRF-Instanz an der Schnittstelle Ethernet0/x.
5. Aktivieren Sie Multicast-Routing auf VRF.
6. Konfigurieren Sie den PIM SSM-Standard in allen Knoten im Core.
7. Konfigurieren Sie BSR RP im CE-Knoten.
8. Vorkonfiguriert:

- VRF m-GRE
- mBGP: Adressfamilie VPNv4
- VRF-Routing-Protokoll

Konfigurieren

1. Aktivieren Sie Multicast-Routing auf allen Knoten.

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

2. Aktivieren Sie PIM Sparse Mode auf der gesamten Schnittstelle.

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

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

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

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

3. Konfigurieren Sie bei bereits vorhandenem VRF den Standard-MDT.

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

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

4. Konfigurieren Sie die VRF-Instanz an der Schnittstelle Ethernet0/x.

Auf PE1, PE2 und 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. Aktivieren Sie Multicast-Routing auf VRF.

Auf PE1, PE2 und PE3.

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

6. RP für den Core des Service Providers konfigurieren.

Auf PE1-, PE2-, PE3- und RR-P-Knoten.

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

7. Konfigurieren Sie BSR RP im CE-Knoten (Receiver).

Auf Receiver2.

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

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

Überprüfen

In diesem Abschnitt überprüfen Sie, ob Ihre Konfiguration ordnungsgemäß funktioniert.

Aufgabe 1: Überprüfen der physischen Verbindung

- Stellen Sie sicher, dass alle angeschlossenen Schnittstellen "UP" sind.

Aufgabe 2: VPNv4-Unicast der Adressfamilie verifizieren

- Stellen Sie sicher, dass BGP auf allen Routern für AF-VPNv4-Unicast aktiviert ist, und dass die BGP-Nachbarn "UP" sind.
- Überprüfen Sie, ob die BGP VPNv4-Unicast-Tabelle über alle Kundenpräfixe verfügt.

Aufgabe 3: Überprüfung des gesamten Multicast-Datenverkehrs.

- Überprüfen Sie die PIM-Nachbarschaft.
- Stellen Sie sicher, dass der Multicast-Status durchgängig erstellt wird.
- Überprüfen Sie den mRIB-Eintrag auf PE1, PE2 und PE3.
- Überprüfen Sie, ob der (S,G) mFIB-Eintrag, der Paketabruf, in der Software-Weiterleitung erhöht wird.
- Überprüfen der Reichweite von ICMP-Paketen zwischen CE und 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

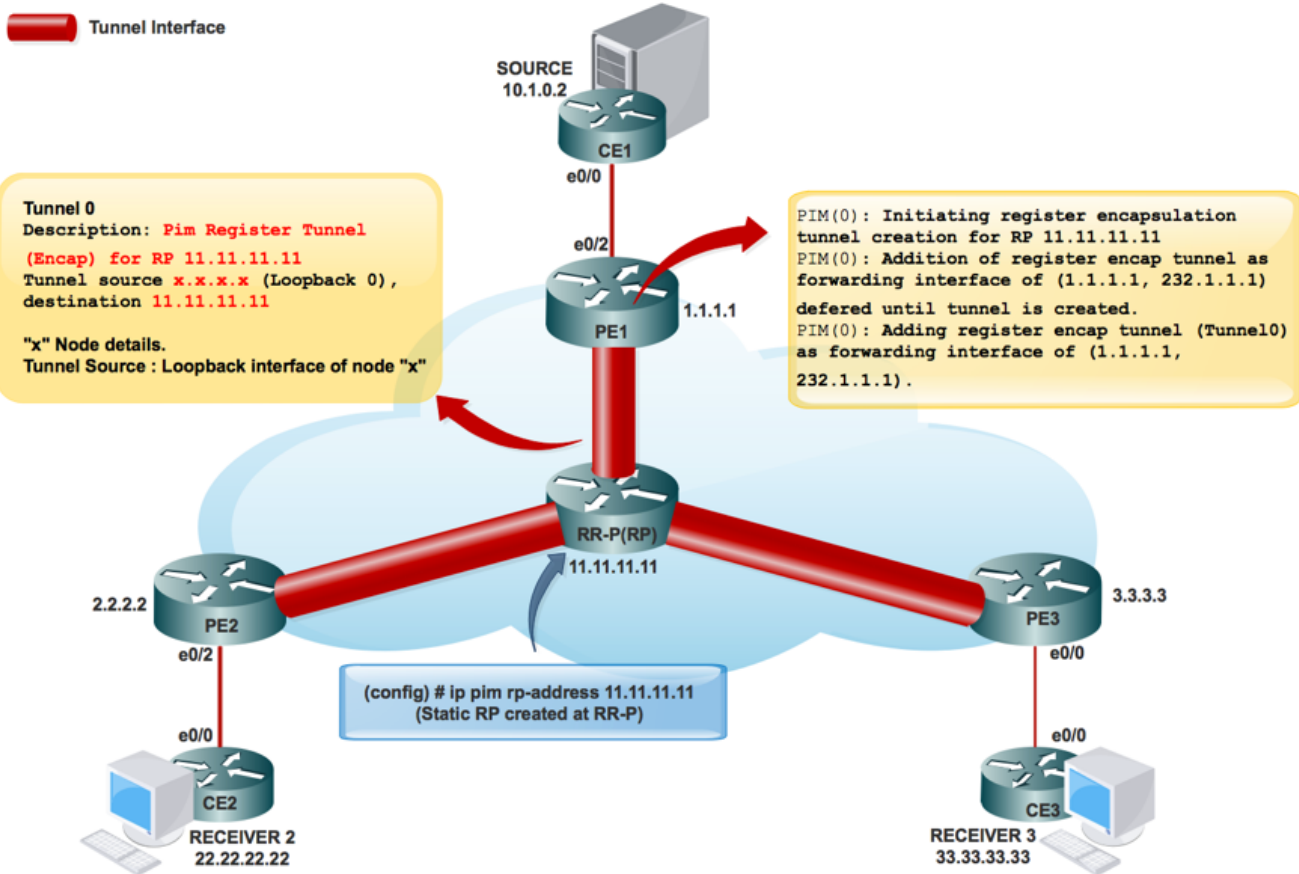
Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 230.1.1.1, timeout is 0
seconds:

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
```

Wenn Tunnelschnittstellen erstellt werden, gilt Folgendes:

Erstellung von Service Provider RPs:

Sobald die RP-Informationen im Core überflutet waren. Interface Tunnel 0 wird erstellt.



PIM(0): Initiieren der Erstellung von Registerkapselungstunneln für RP 11.11.11.11.

PIM(0): Die Erstellung des ersten Registertunnel wurde für RP 11.11.11.11 erfolgreich durchgeführt.

PIM(0): Die Aufnahme des Registerumschließungstunnels als Weiterleitungsschnittstelle von (1.1.1.1, 232.1.1.1) wird bis zur Erstellung des Tunnels zurückgestellt.

* 9. Mai 17:34:56.155: PIM(0): Überprüfen Sie RP 11.11.11.11 in das Fenster (*, 232.1.1.1).

PIM(0): Registerencap-Tunnel (Tunnel0) als Weiterleitungsschnittstelle von (1.1.1.1, 232.1.1.1) hinzugefügt.

```
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

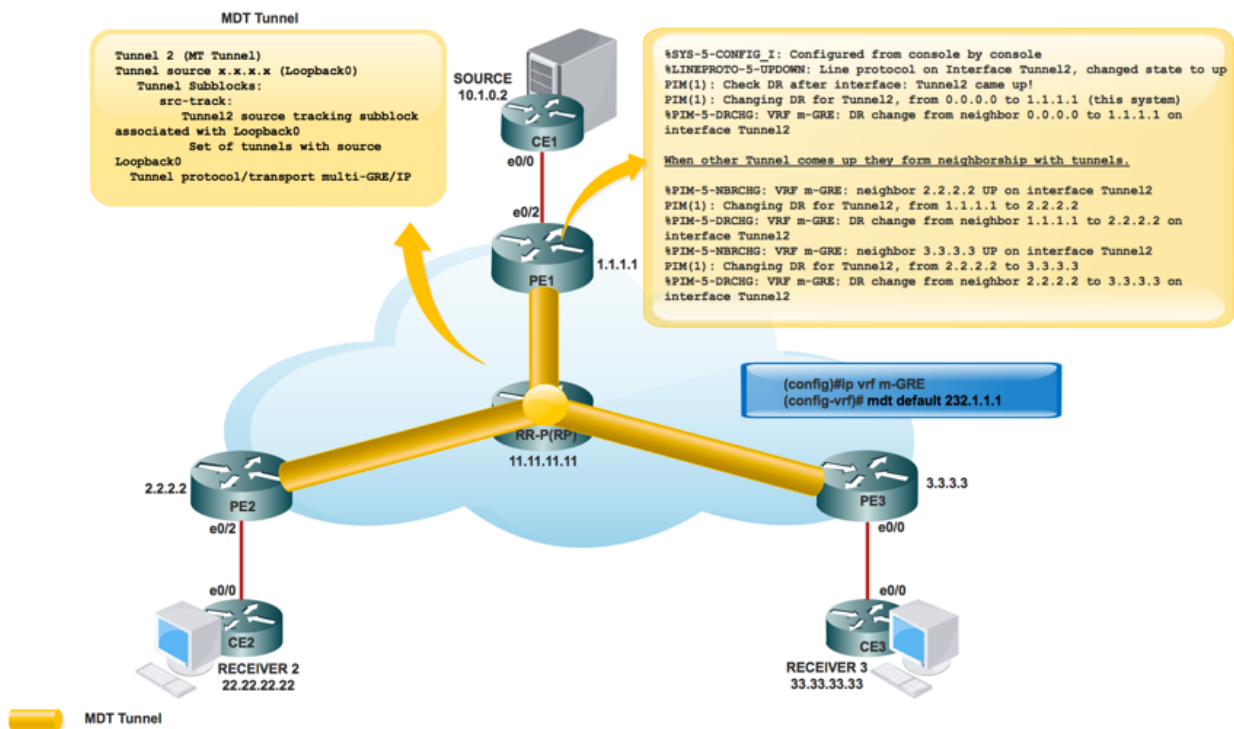
Set of tunnels with source Ethernet0/1, 1 member (includes iterators), on interface <OK>

Tunnel protocol/transport PIM/IPv4

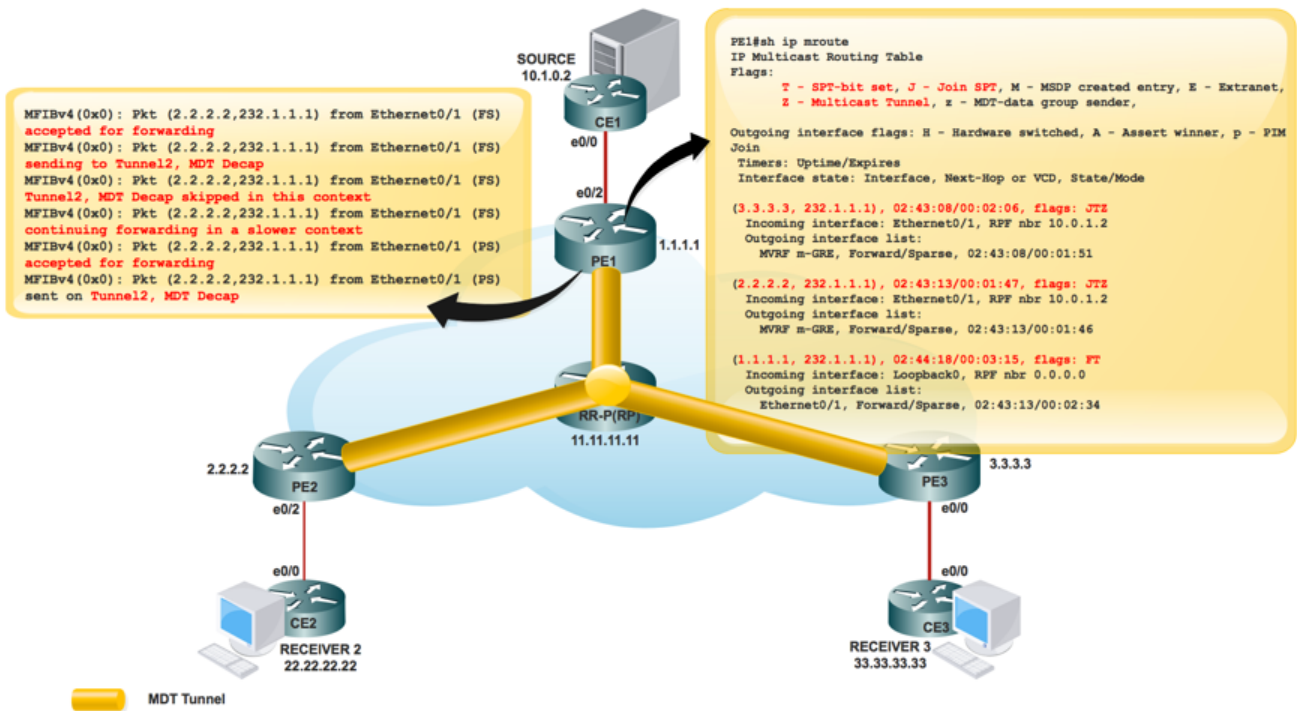
Tunnel TOS/Traffic Class 0xC0, Tunnel TTL 255

Tunnel transport MTU 1472 bytes

MDT-Tunnelerstellung:



MRIB-Erstellung im 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:

MVRP 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:

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

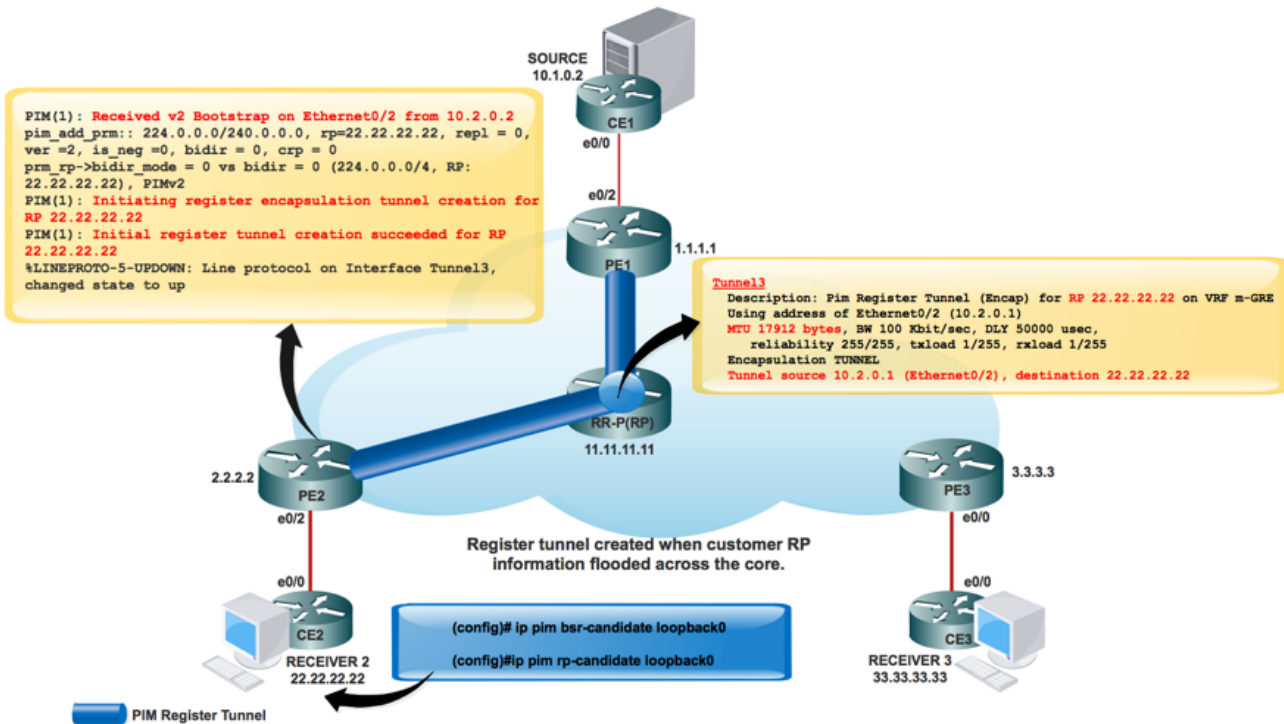
(1.1.1.1, 232.1.1.1), 00:10:15/00:03:20, flags: FT

Incoming interface: Loopback0, RPF nbr 0.0.0.0

Outgoing interface list:

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

Sobald der RP für das Kundennetzwerk erstellt wurde:



*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

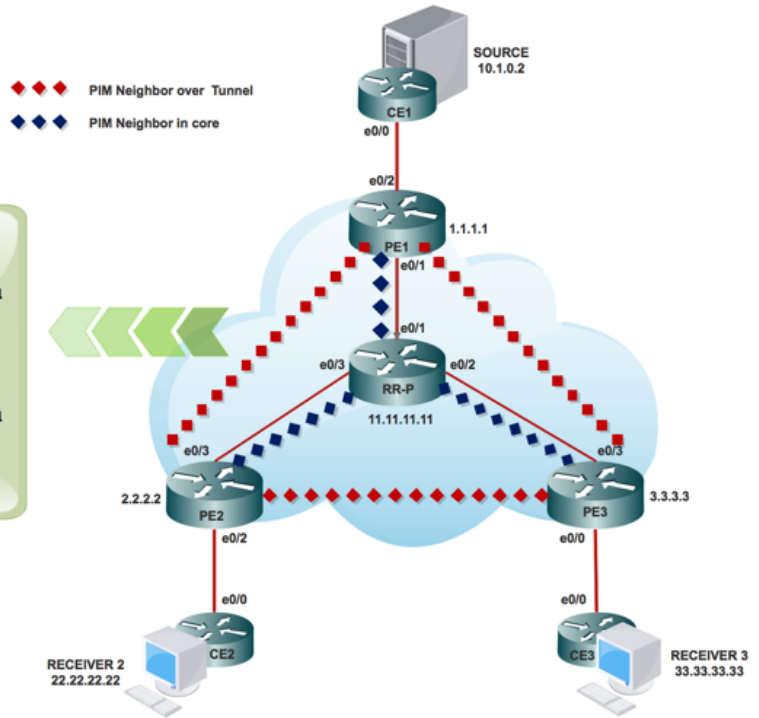
Die Tunnelschnittstelle wird erstellt, um die RP-Informationen des Kunden zu übertragen.

PIM(1): Initiieren der Erstellung von Registerkapselungstunneln für RP 22.22.22.22.

Es ist der Tunnel, der erstellt wurde, um die Kapselung an RP registrieren zu können.

Für jeden entdeckten Sparse-Mode-RP wird ein Register-Kapselungs-Tunnel erstellt. Auf dem Sparse-Mode-RP selbst gibt es eine Entkapselungstunnel-Schnittstelle, die zum Empfangen von Registerpaketen erstellt wurde.

PIM-Nachbarschaft:



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

Paketfluss:

Der Paketfluss der Kontrollebene ist in zwei Teile unterteilt.

1. Empfänger ist online.

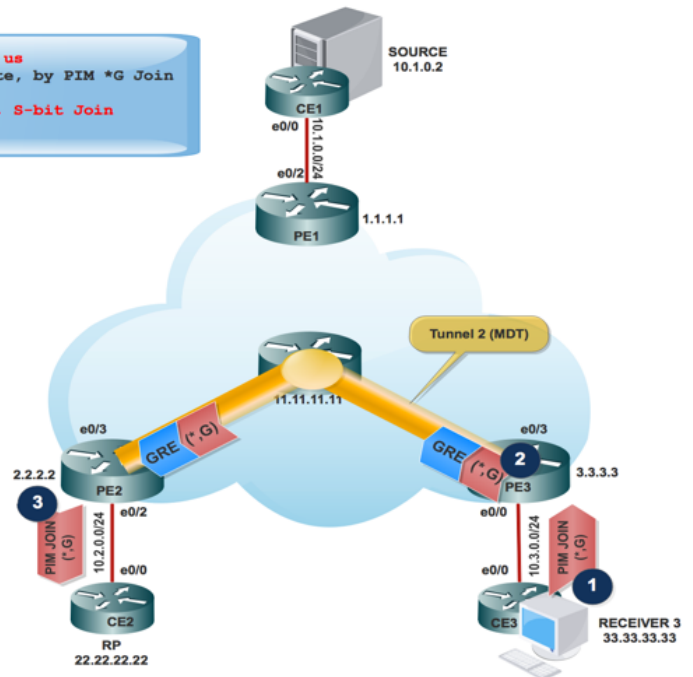
2. Quelle ist aktiv.

Wenn Receiver aktiv ist:

```
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. Empfänger kommt online und sendet PIM JOIN (*,G) an PE3.

2. PE3 kapselt den PIM JOIN (*,G) in GRE-Paket und sendet über den Tunnel 2 (MDT-Tunnel), der von der eingehenden Schnittstelle von `show ip mroute vrf m-GRE` verifiziert wird.

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 empfängt das GRE-Paket mit Source als 3.3.3.3 und Destination 232.1.1.1 und leitet es basierend auf dem OIL an MVRF m-GRE weiter.

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

Das GRE-Paket wird entkapselt, und PIM JOIN sendet an den RP.

Hinweis: RPF Neighbor ist 2.2.2.2, da die PIM-Join-Verbindung zur RP-Adresse bestimmt ist, um den RPT durch den Core zu bilden.

Hinweis: WC Bit und RPT Bit: Wird durch den Zustand (*,G) ausgelöst, erstellt der DR eine Join/Prune-Nachricht, wobei die RP-Adresse in der Join-Liste und das Platzhalter-Bit (WC-Bit) und das RP-Tree-Bit (RPT-Bit) auf 1 festgelegt sind. Das WC-Bit gibt an, dass eine Quelle übereinstimmen und gemäß diesem Eintrag weitergeleitet werden kann, wenn keine Übereinstimmung mehr besteht. Das RPT-Bit gibt an, dass diese Verbindung über den gemeinsam genutzten RP-Tree gesendet wird. Die Beschneidungsliste ist leer. Wenn das RPT-Bit auf 1 festgelegt ist, bedeutet dies, dass die Join-Nachricht mit dem freigegebenen RP-Tree verknüpft ist und daher die Join/Prune-Nachricht über den RP-Tree weitergeleitet wird. Wenn das WC-Bit auf 1 gesetzt ist, weist dies darauf hin, dass es sich bei der Adresse um einen RP handelt und die Downstream-Empfänger erwarten, über diesen (Shared Tree)-Pfad Pakete von allen Quellen zu empfangen.

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. Gekapselte GRE-Paketreichweite am Quell-PE 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) erreicht den Source CE.

3. Nun erhielt die Quelle die Informationen des Interested Receivers und der Datenverkehr beginnt an den Quell-PE1 zu senden.

4. Beim Quell-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

Bei 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

PCAP-Erfassung von Multicast-Paketen aus PE1. Getunnelt im MDT-Standard-Tunnel. Kapselt mit GRE.

5. Beim Empfänger-PE-3 wird das Paket empfangen.

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
```

Hinweis: Der RPF Neighbor wird geändert, sobald ein Multicast-Paket von PE1 empfangen wird. Zuvor war es PE2 als RP gehostet dahinter. Nachdem das erste Multicast-Paket empfangen wurde, wird das RPF geändert und das SPT-Bit festgelegt.

Datenverkehrsfluss über Standard-MDT-Tunnel:

- Bei der Weiterleitung auf dem MDT wird GRE verwendet. Das C-Paket wird zu einem P-Paket.
- P-Paket S-Adresse = PEs BGP-Peering-Adresse
G-Adresse = MDT-Gruppenadresse (Standard oder Daten)
- Die IP-Nutzungsbedingungen für das C-Paket werden in das P-Paket kopiert.
- MPLS-Labels werden NICHT im Core, sondern nur im nativen Multicast verwendet.

Paketfluss:

1. Ein C-Paket erreicht eine VRF-konfigurierte PE-Schnittstelle, mVRF ist implizit identifiziert. Normale RPF-Prüfung auf C Source.

Das C-Paket replizierte die Schnittstelle in OIL. An diesem Punkt handelt es sich um eine PE-Schnittstelle in derselben VRF-Instanz.

```
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)
```

Wenn die OIL einen MTI enthält, kapselt das C-Paket in ein P-Paket. Wenn die Markierung "y" für das verwendete Einstiegsziel festgelegt ist, ist die DATA-MDT-Gruppe, ansonsten die Standard-MDT-Gruppe. Die Quelle ist die PE-BGP-Peer-Adresse, und das Ziel ist die Adresse der MDT-Gruppe.

2. Das P-Paket wird als normales Multicast über das P-Netzwerk weitergeleitet.

Das Paket kommt an der globalen Schnittstelle an. Globaler (S,G) oder (*,G) Eintrag für die referenzierte MDT-Gruppe. Normale RPF-Prüfung auf P-Source (PE-Peer).

3. Das P-Paket wird außerhalb der Schnittstelle in OIL repliziert. An diesem Punkt befindet sich P/PE in der globalen Routing-Tabelle.

4. Wenn das "Z"-Flag festlegt, wird das Paket entkapselt, um das C-Paket preiszugeben. Die von der MDT-Gruppe abgeleitete Ziel-mVRF-Instanz und die eingehende Schnittstelle sind das Ziel des gekapselten Headers.

RPF-Prüfung des C-Packets in mVRF erfolgt, replizierte das C-Paket OIL 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,

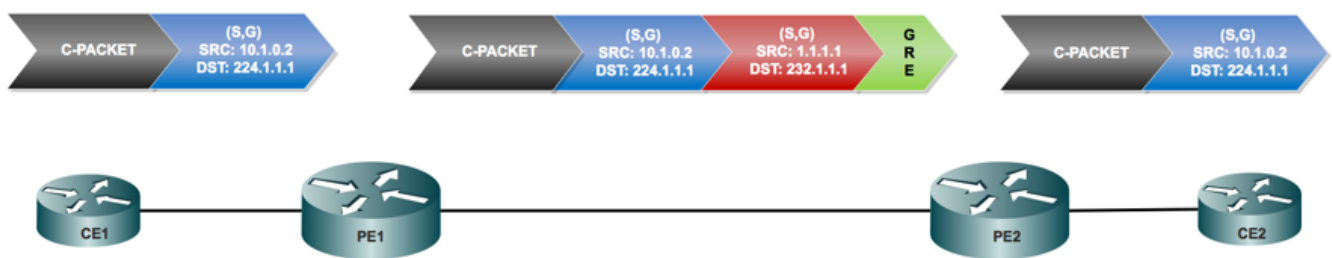
(1.1.1.1, 232.1.1.1), 1d01h/00:02:47, flags: **JTZ**

Incoming interface: Ethernet0/3, RPF nbr 10.0.3.2

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

5. Native C-Paketreichweite am Receiver 3.

Packet Encapsulation:

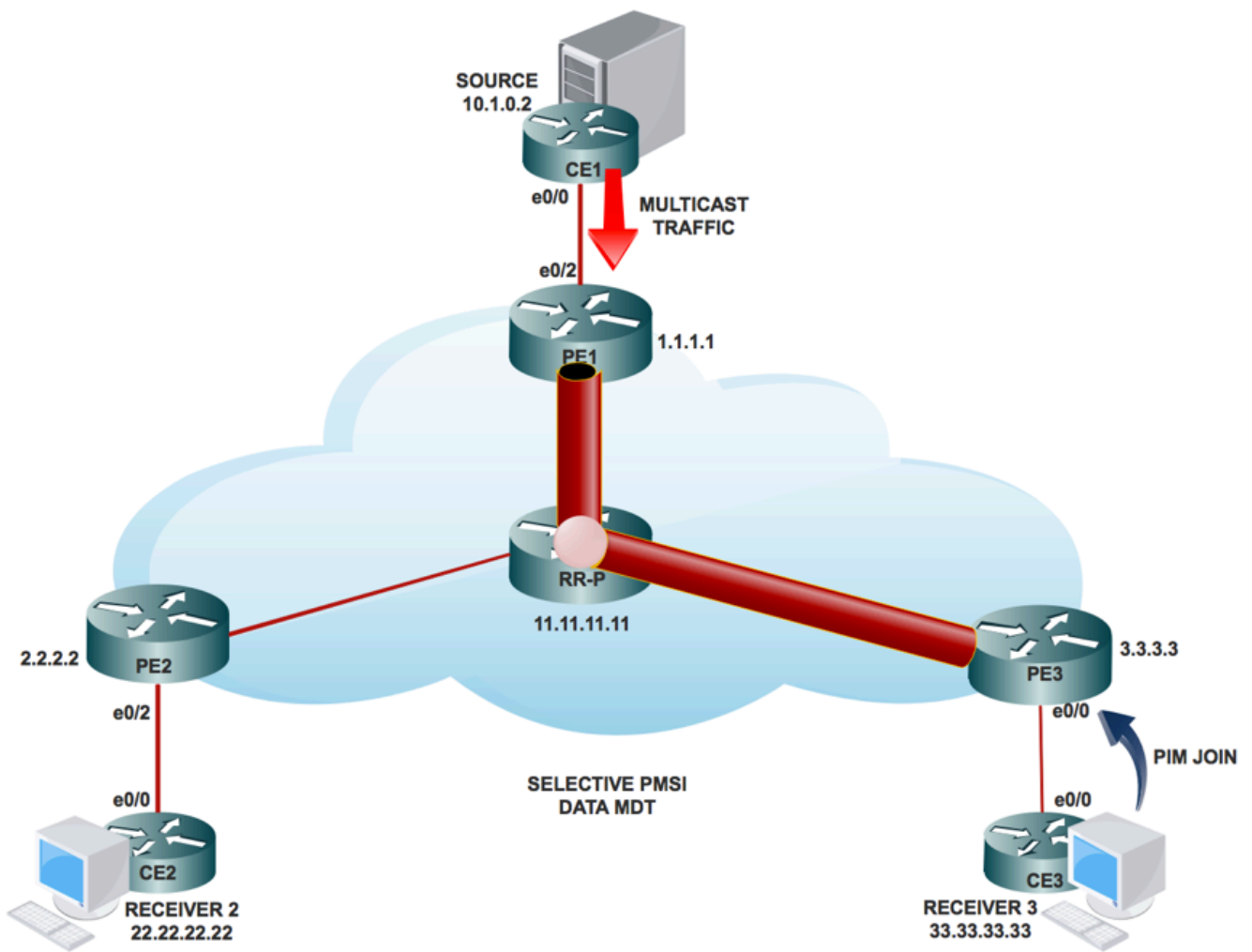


Daten-MDT:

Was ist Daten-MDT?

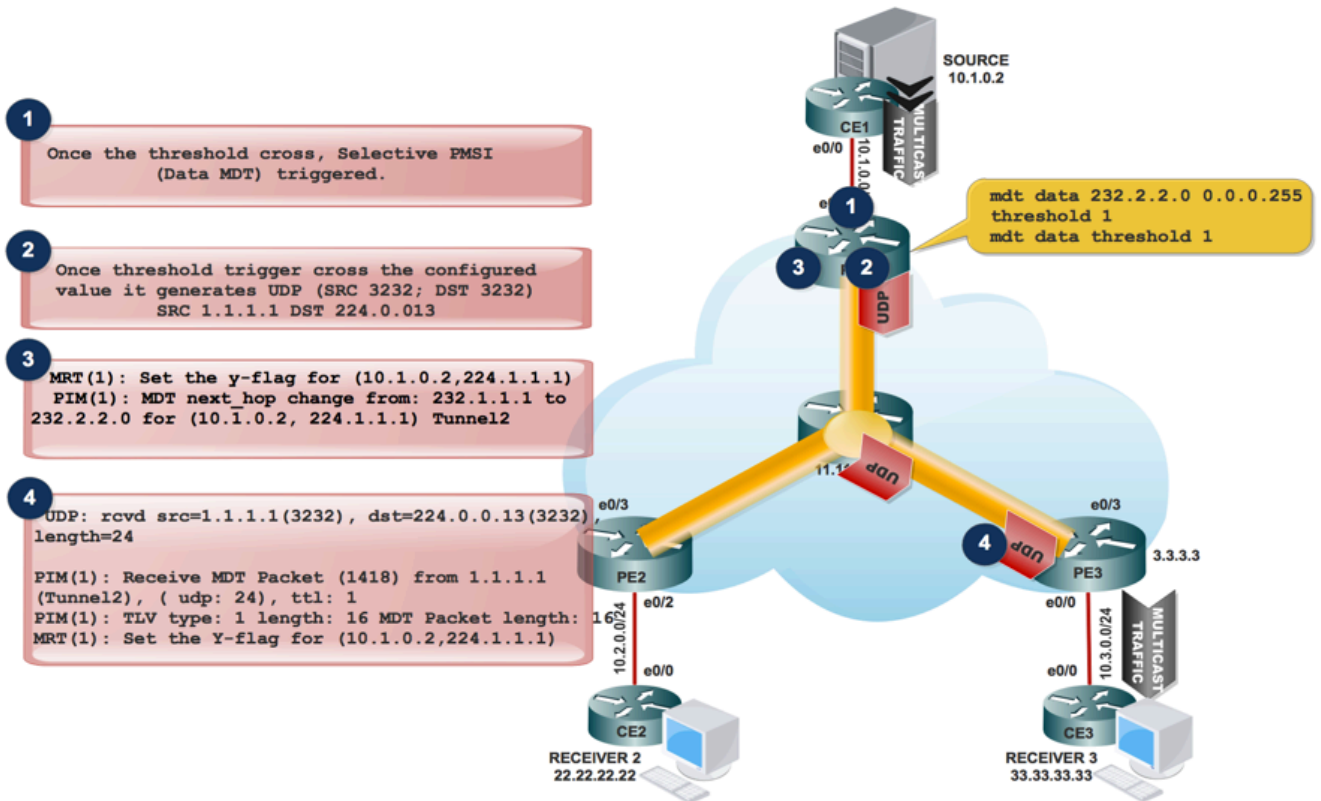
Sie ist optional. Er wird nach Bedarf erstellt und überträgt spezifischen (S,G)-Datenverkehr. In der neuesten IOS®-Version ist der konfigurierte Grenzwert "0" und "unendlich". Wenn ein erstes Paket auf die VRF-Instanz trifft, initialisiert der Daten-MDT die Daten-MDT. Bei Unendlichkeit wird der Daten-MDT niemals erstellt, und der Datenverkehr wird im Standard-MDT weitergeleitet. Der Daten-MDT ist immer der empfangende Tree, der niemals Datenverkehr sendet. Daten-MDT ist nur für den (S,G)-Datenverkehr bestimmt.

Selektive PMSI:



- Sie ist optional. Er wird nach Bedarf erstellt und überträgt spezifischen (S, G) Datenverkehr.
- Wenn ein erstes Paket auf die VRF-Instanz trifft, initialisiert der Daten-MDT den Daten-MDT. Bei Unendlichkeit wird der Daten-MDT niemals erstellt, und der Datenverkehr wird im Standard-MDT weitergeleitet.
- Der Daten-MDT ist immer der empfangende Tree, der niemals Datenverkehr sendet. Daten-MDT ist nur für den (S-, G-) Datenverkehr bestimmt.
- PIM-Nachrichten enthalten C- (S, G) und P-Group.

Erstellung von DATA MDT:



1. Wenn Multicast-Datenverkehr in die VRF-Instanz gelangt und die Datenverkehrsrate den Grenzwert erreicht. Er generiert ein MDT-Paket.

2. Das MDT-Paket wird in UDP mit Source und Destination 3232 gekapselt. und an den interessierten Empfänger senden.

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. Nachdem das UDP-Paket an den interessierten Empfänger gesendet wurde, wird das "y"-Flag festgelegt und der MDT next_hop in die neue MDT-Gruppenadresse geändert.

Beim Quell-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: T**yp**

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)

Hinweis: Der nächste Hop von OIL wechselt zu 232.2.2.0.

2. AT PE3, wenn es das MDT-Paket empfängt, das in den UDP SRC-Port 3232 und den DST-Port 3232 eingekapselt ist.

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**Yp**

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

Outgoing interface list:

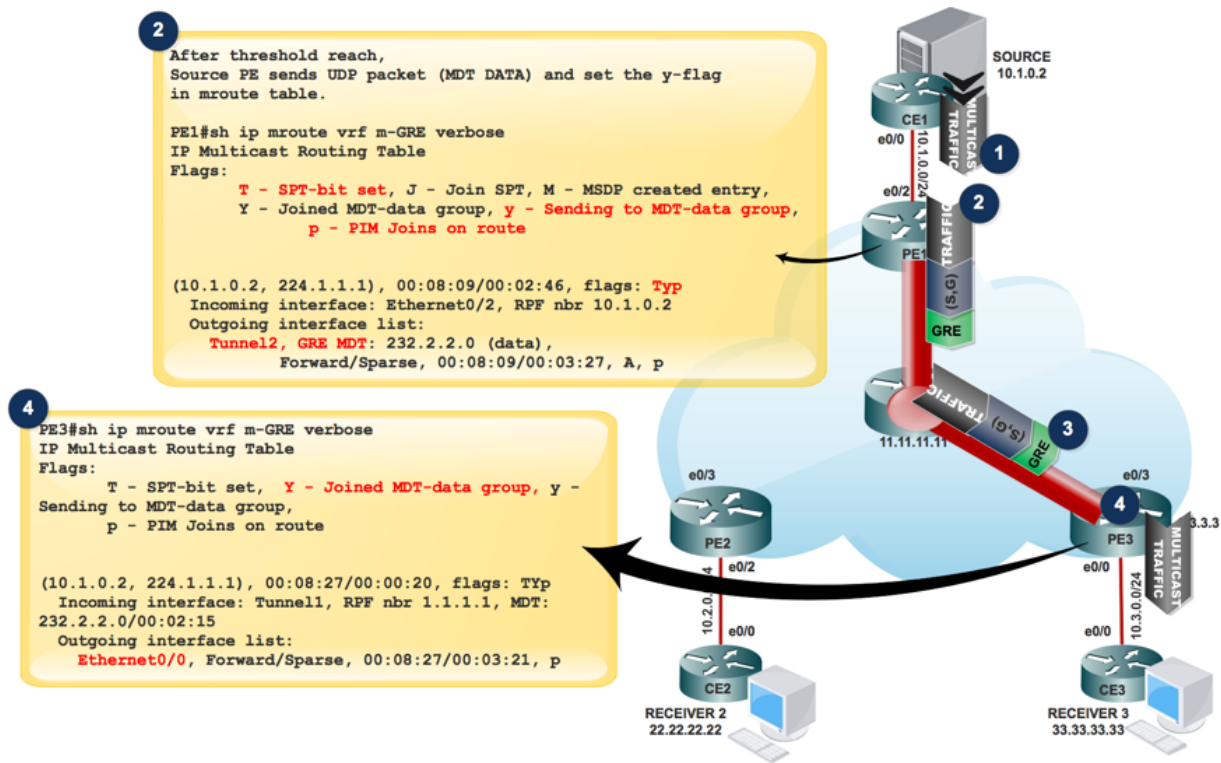
Ethernet0/0, Forward/Sparse, 00:08:27/00:03:21, p

Die S-PMSI-Join-Nachricht ist eine durch UDP gekapselte Nachricht, deren Zieladresse ALL-PIM-ROUTERS (224.0.0.13) ist und deren Zielport 3232 lautet.

Die S-PMSI-Join-Nachricht enthält folgende Informationen: Ein Bezeichner für den bestimmten Multicast-Stream, der an den P-Tunnel gebunden werden soll. Diese kann als (S,G) Paar dargestellt werden. Eine Kennung für den bestimmten P-Tunnel, an den der Stream gebunden

werden soll. Dieser Bezeichner ist ein strukturiertes Feld, das folgende Informationen enthält:

Multicast-Datenverkehrsfluss im MDT-DATA-Tunnel:



```
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
```

- Wenn die OIL eine Tunnelschnittstelle enthält, wird das Paket mit der Verwendung von GRE gekapselt, wobei die Quelle die BGP-Peering-Adresse des lokalen PE-Routers und das Ziel die MDT-Gruppenadresse ist.
- Die Entscheidung für die Daten-MDT-Gruppe hängt davon ab, ob das y-Flag auf dem (S,G)-Eintrag im mVRF festgelegt ist.
- Wenn für den Eintrag (S, G) oder (*, G) das Z-Flag festgelegt ist, handelt es sich um einen Standard- oder Daten-MDT mit einem zugeordneten mVRF.

- Das P-Paket muss entkapselt werden, um das C-Paket preiszugeben.

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

- Da in der mVRF-Instanz pro Multicast-Domäne nur eine einzige MTI vorhanden ist, verwenden sowohl der Daten-MDT als auch der Standard-MDT dieselbe Tunnelschnittstelle für den Kundendatenverkehr.
- Die Y/y-Markierungen sind erforderlich, um den Standard-MDT-Datenverkehr vom Daten-MDT-Datenverkehr zu unterscheiden und sicherzustellen, dass die Multicast-Routing-Einträge des Kunden die richtige MDT-Datengruppe verwenden und auf eine interne Tabelle verweisen, in der die (S, G, Data-MDT)-Zuordnungen gespeichert sind.

Fehlerbehebung

Für diese Konfiguration sind derzeit keine spezifischen Informationen zur Fehlerbehebung verfügbar.