

Implementieren eines MPLS-VPN über TE-Tunnel

Inhalt

[Einführung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Konventionen](#)

[Hintergrundtheorie](#)

[Anfängliche VPN-Einrichtung zwischen CE1 und CE2 ohne TE-Tunnel](#)

[Topologie](#)

[Konfiguration](#)

[Überprüfung](#)

[Fall 1: VPN über einen TE-Tunnel, wenn der TE-Tunnel vom PE1 zum PE2 verläuft](#)

[Topologie](#)

[Konfiguration](#)

[Überprüfung](#)

[Fall 2: VPN über einen TE-Tunnel, wenn der TE-Tunnel vom PE1 zum P2 verläuft](#)

[Topologie](#)

[Konfiguration](#)

[Überprüfung](#)

[Erläuterung](#)

[Lösung](#)

[Fall 3: VPN zwischen CE1 und CE2 über einen TE-Tunnel von P1 bis P2, wenn TDP/LDP nicht aktiviert ist](#)

[Topologie](#)

[Konfiguration](#)

[Überprüfung](#)

[Lösung](#)

[Fall 4: VPN über einen TE-Tunnel zwischen P1 und P2 mit aktiviertem LDP](#)

[Topologie](#)

[Konfiguration](#)

[Überprüfung](#)

[Fall 5: MPLS-VPN über einen Tunnel zwischen P1 und PE2](#)

[Topologie](#)

[Konfiguration](#)

[Überprüfung](#)

[Bekannte Probleme](#)

[Schlussfolgerung](#)

[Einführung](#)

Dieses Dokument enthält Beispielkonfigurationen für die Implementierung eines MPLS-VPNs (Multiprotocol Label Switching) über Traffic Engineering-Tunnel (TE) in einem MPLS-Netzwerk. Um von den Vorteilen eines MPLS-VPN über TE-Tunnel zu profitieren, sollten beide im Netzwerk vorhanden sein. Dieses Dokument zeigt verschiedene Szenarien, in denen erläutert wird, warum die Paketweiterleitung innerhalb eines MPLS-VPN über TE-Tunnel fehlschlägt. Es bietet auch eine mögliche Lösung.

[Voraussetzungen](#)

[Anforderungen](#)

Die Leser dieses Dokuments sollten folgende Themen kennen:

- [MPLS Traffic Engineering und Erweiterungen](#)
- [Konfigurieren eines einfachen MPLS-VPN](#)

[Verwendete Komponenten](#)

Dieses Dokument ist nicht auf bestimmte Software- und Hardwareversionen beschränkt.

[Konventionen](#)

Weitere Informationen zu Dokumentkonventionen finden Sie unter [Cisco Technical Tips Conventions](#) (Technische Tipps zu Konventionen von Cisco).

[Hintergrundtheorie](#)



Wie in dieser Topologie gezeigt, erfährt der Provider Edge 1 (PE1) in einer einfachen MPLS-VPN-Konfiguration das VPN-Label (Label 1 [L1]) für das VPN-Präfix 172.16.13.0/24 direkt über das Multiprotocol Border Gateway Protocol (MPBGP) von PE2, wobei der nächste Hop als PE2-Loopback-Adresse gilt. PE1 erfährt auch das Label (L2) für die PE2-Loopback-Adresse über das Label Distribution Protocol (LDP) vom nächsten Hop P1.

Beim Weiterleiten von Daten an das VPN-Präfix 172.16.13.13 verwendet PE1 einen Label-Stack {L2 L1} mit L2 als äußere Bezeichnung. L2 wird durch den Transit-Label-Switch-Router (LSR), P1 ersetzt. P2 öffnet das äußere L2 und leitet das Paket mit nur einem L1 an PE2 weiter. Um besser zu verstehen, warum P2 L2 aufruft, lesen Sie Abschnitt 3.16 über vorläufiges Hop Popping (PHP) in [RFC 3031](#). Daher werden Pakete mit dem VPN IP Version 4 (IPv4)-Präfix 172.16.13.0/24 über ein MPLS-Netzwerk weitergeleitet.

Der MPLS-VPN-Weiterleitungsvorgang schlägt fehl, wenn ein P-Router das Paket mit dem L1-Label (VPN-Label) als einziges Außenlabel anstelle des {L2 L1}-Label-Stacks empfängt. Dies liegt daran, dass keiner der P-Router L1 in seiner Label Forwarding Information Base (LFIB) zum Umschalten des Pakets hat.

Ein MPLS-TE verwendet das Resource Reservation Protocol (RSVP), um Labels auszutauschen. Wenn ein Router sowohl für das TE- als auch für das Tag Distribution Protocol (TDP)/LDP konfiguriert ist, erhält der Router für ein bestimmtes Präfix verschiedene Labels vom LDP und vom RSVP. Die Labels von LDP und RSVP müssen nicht in allen Situationen identisch sein. Der Router installiert ein LDP-Label in der Weiterleitungstabelle, wenn das Präfix über eine LDP-Schnittstelle erfasst wird, und er installiert das RSVP-Label in der Weiterleitungstabelle, wenn das Präfix über eine TE-Tunnelschnittstelle erfasst wird.

Im Fall eines reinen TE-Tunnels (ohne dass LDP/TDP im Tunnel aktiviert ist) verwendet der Eingangs-LSR (der LSR am Headend des TE-Tunnels) für alle Routen, die durch einen TE-Tunnel erfasst werden, das gleiche Label wie für das Erreichen des Zielendes des TE-Tunnels.

Zum Beispiel gibt es einen TE-Tunnel von PE1 zu P2, der das Präfix 10.11.11.11/32 über den Tunnel erfährt. Der Tunnelausschnitt auf P2 beträgt 10.5.5.5, und das Label, das 10.5.5.5 in PE1 erreicht, ist L3. Anschließend erreicht PE1 mithilfe von L3 das Ziel 10.11.11.11/32, das über den TE-Tunnel gelernt wurde.

Wenn im Szenario [oben](#) ein TE-Tunnel zwischen PE1 und P2 vorhanden ist, berücksichtigen Sie, dass PE1 Daten an Customer Edge 2 (CE2) weiterleitet. Wenn L4 das VPN-Label ist, leitet PE1 die Daten mit dem Label-Stack {L3 L4} weiter. P1 öffnet L3, und P2 empfängt das Paket mit L4. PE2 ist der einzige LSR, der das Paket korrekt mit dem äußeren Label L4 weiterleiten kann. P2 verfügt über keine MPBGP-Sitzung mit PE2, daher wird das L4 von PE2 nicht empfangen. Aus diesem Grund hat P2 keine Kenntnis von L2 und verwirft das Paket.

Die Konfigurationen und **zeigenden** Ergebnisse zeigen dies und illustrieren eine mögliche Lösung für dieses Problem.

Anfängliche VPN-Einrichtung zwischen CE1 und CE2 ohne TE-Tunnel

Topologie



Konfiguration

Hier sind nur die relevanten Teile der Konfigurationsdateien enthalten:

PE1

```
hostname PE1
ip cef
```

```

!
ip vrf aqua
  rd 100:1
  route-target export 1:1
  route-target import 1:1
!
mpls traffic-eng tunnels
!
interface Loopback0
  ip address 10.2.2.2 255.255.255.255
  no ip directed-broadcast
!
interface Ethernet2/0/1
  ip vrf forwarding aqua
  ip address 172.16.1.2 255.255.255.0
!
interface Ethernet2/0/2
  ip address 10.7.7.2 255.255.255.0
  ip router isis
  mpls traffic-eng tunnels
  tag-switching ip
!
router isis
  passive-interface Loopback0
  net 47.1234.2222.2222.00
  is-type level-1
  metric-style wide
  mpls traffic-eng router-id Loopback0
  mpls traffic-eng level-1
!
router bgp 1
  bgp log-neighbor-changes
  neighbor 10.11.11.11 remote-as 1
  neighbor 10.11.11.11 update-source Loopback0
  !
  address-family vpnv4
  neighbor 10.11.11.11 activate
  neighbor 10.11.11.11 send-community extended
  exit-address-family
  !
  address-family ipv4
  neighbor 10.11.11.11 activate
  no auto-summary
  no synchronization
  exit-address-family
  !
  address-family ipv4 vrf aqua
  redistribute connected
  no auto-summary
  no synchronization
  exit-address-family

```

PE2

```

hostname PE2
!
ip vrf aqua
  rd 100:1
  route-target export 1:1
  route-target import 1:1
!
mpls traffic-eng tunnels
!

```

```

interface Loopback0
 ip address 10.11.11.11 255.255.255.255
!
interface POS0/1
 ip address 10.12.12.10 255.255.255.0
 ip router isis
 mpls traffic-eng tunnels
 tag-switching ip
 crc 16
 clock source internal
!
interface POS5/1
 ip vrf forwarding aqua
 ip address 172.16.13.11 255.255.255.0
 crc 32
 clock source internal
!
router isis
 passive-interface Loopback0
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng level-1
 net 47.1234.1010.1010.1010.00
 is-type level-1
 metric-style wide
!
router bgp 1
 bgp log-neighbor-changes
 neighbor 10.2.2.2 remote-as 1
 neighbor 10.2.2.2 update-source Loopback0
 no auto-summary
!
 address-family vpnv4
 neighbor 10.2.2.2 activate
 neighbor 10.2.2.2 send-community extended
 exit-address-family
!
 address-family ipv4 vrf aqua
 redistribute connected
 no auto-summary
 no synchronization
 exit-address-family
!

```

Überprüfung

PE2 erhält das PE1 VPN IPv4-Präfix 172.16.1.0/24 über MPBGP-Peering zwischen PE1 und PE2.
Hier sehen Sie:

```
PE2# show ip route vrf aqua
```

```

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR

```

```
Gateway of last resort is not set
```

```
10.0.0.0/24 is subnetted, 2 subnets
```

```

B    172.16.1.0 [200/0] via 10.2.2.2, 16:09:10
C    172.16.13.0 is directly connected, POS5/1

```

Ebenso empfängt PE1 das PE2 VPN IPv4-Präfix 172.16.13.0/24 über MPBGP-Peering zwischen PE1 und PE2. Hier sehen Sie:

```
PE1# show ip route vrf aqua
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
```

Gateway of last resort is not set

```
10.0.0.0/24 is subnetted, 2 subnets
B      172.16.13.0 [200/0] via 10.11.11.11, 16:09:49
C      172.16.1.0 is directly connected, Ethernet2/0/1
```

```
PE1# show ip route vrf aqua 172.16.13.13
Routing entry for 172.16.13.0/24
  Known via "bgp 1", distance 200, metric 0, type internal
  Last update from 10.11.11.11 16:13:19 ago
  Routing Descriptor Blocks:
  * 10.11.11.11 (Default-IP-Routing-Table), from 10.11.11.11, 16:13:19 ago
    Route metric is 0, traffic share count is 1
    AS Hops 0, BGP network version 0
```

```
PE1# show ip cef vrf aqua 172.16.13.13
172.16.13.0/24, version 11, cached adjacency 10.7.7.7
0 packets, 0 bytes
  tag information set
    local tag: VPN route head
    fast tag rewrite with Et2/0/2, 10.7.7.7, tags imposed {17 12308}
  via 10.11.11.11, 0 dependencies, recursive
    next hop 10.7.7.7, Ethernet2/0/2 via 10.11.11.11/32
    valid cached adjacency
    tag rewrite with Et2/0/2, 10.7.7.7, tags imposed {17 12308}
!--- The label stack used to reach 172.16.13.13 is !--- {17 12308}, where 17 is the outer label
to reach next hop 10.11.11.11 !--- and 12308 is the VPN IPv4 label for 172.16.13.0/24. PE1# show
ip cef 10.11.11.11
10.11.11.11/32, version 31, cached adjacency 10.7.7.7
0 packets, 0 bytes
  tag information set
    local tag: 21
    fast tag rewrite with Et2/0/2, 10.7.7.7, tags imposed {17}
  via 10.7.7.7, Ethernet2/0/2, 1 dependency
    next hop 10.7.7.7, Ethernet2/0/2
    valid cached adjacency
    tag rewrite with Et2/0/2, 10.7.7.7, tags imposed {17}
!--- Outer label 17 is used to reach next hop 10.11.11.11.
```

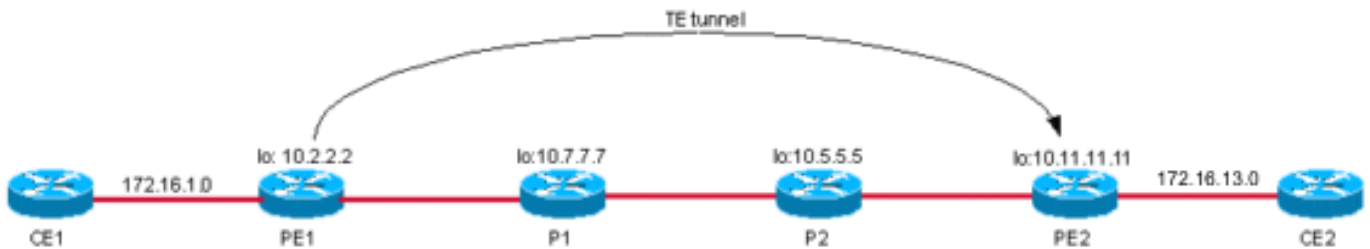
So kann CE1 über die VPN Routing and Forwarding (VRF)-Instanz "aqua", die auf PE1 mit dem Label-Stack {17 12308} konfiguriert ist, 172.16.13.13 im CE2-Netzwerk erreichen.

Diese Ping-Ausgabe bestätigt die Konnektivität:

```
CE1# ping 172.16.13.13
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.13, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
```

Fall 1: VPN über einen TE-Tunnel, wenn der TE-Tunnel vom PE1 zum PE2 verläuft

Topologie



Wenn der TE-Tunnel zwischen den PE-Routern mit der verwendeten automatischen Ankündigung erstellt wird, ist der nächste BGP-Egress-PE-Router über die TE-Tunnelschnittstelle erreichbar. Daher verwendet PE1 das TE-Label, um PE2 zu erreichen.

Hinweis: MPLS TE ist LDP-unabhängig, d. h. wenn Sie über ein vollständiges Tunnelnetz vom PE zum PE verfügen, können Sie LDP in den Routern effektiv deaktivieren und LDP nicht auf den TE-Tunnelschnittstellen ausführen. Sie müssen jedoch alle Tunnel zum nächsten BGP-Hop der VPN-Routen der Version 4 (VPNv4) erstellen. Im Beispiel in dieser [Konfiguration](#) sehen Sie, dass dieser BGP Next Hop der Loopback0 auf PE2, 10.11.11.11 ist. Dieselbe Loopback ist auch das Tunnelziel für den Tunnel von PE1 zu PE2. Dies erklärt, warum Sie in diesem Beispiel LDP im Core deaktivieren können, wenn es auch einen Tunnel von PE2 zu PE1 für den Rückverkehr gibt. Die Weiterleitung vom CE zum CE funktioniert dann mit dem gesamten VPNv4-Datenverkehr, der über die TE-Tunnel übertragen wird. Wenn der nächste BGP-Hop nicht mit dem Ziel des TE-Tunnels identisch ist, muss LDP im Core und im TE-Tunnel ausgeführt werden.

Konfiguration

Die zusätzliche Konfiguration für PE1 zum Einrichten eines PE-Tunnels wird hier gezeigt:

```
PE1
PE1# show run interface tunnel 0
!
interface Tunnel0
 ip unnumbered Loopback0
 no ip directed-broadcast
 no ip route-cache distributed
 tunnel destination 10.11.11.11
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 10 dynamic
end
```

Überprüfung

```
PE1# show ip cef vrf aqua 172.16.13.13
```

```

172.16.13.0/24, version 11
0 packets, 0 bytes
tag information set
  local tag: VPN route head
  fast tag rewrite with Tu0, point2point, tags imposed {19 12308}
via 10.11.11.11, 0 dependencies, recursive
  next hop 10.11.11.11, Tunnel0 via 10.11.11.11/32
  valid adjacency
  tag rewrite with Tu0, point2point, tags imposed {19 12308}
!--- The label stack to reach 172.16.13.13 is {19 12308}. !--- BGP next hop for the VPNv4 prefix
is 10.11.11.11, which is !--- the same as the TE tunnel destination. PE1# show ip route
10.11.11.11
Routing entry for 10.11.11.11/32
  Known via "isis", distance 115, metric 40, type level-1
  Redistributing via isis
  Last update from 10.11.11.11 on Tunnel0, 00:02:09 ago
  Routing Descriptor Blocks:
  * 10.11.11.11, from 10.11.11.11, via Tunnel0
!--- The route is via Tunnel0. Route metric is 40, traffic share count is 1

```

Jetzt bestätigen Sie das äußere Label, das verwendet wird, um den nächsten Hop 10.11.11.11 über Tunnel0 zu erreichen.

```
PE1# show mpls traffic-eng tunnels tunnel 0
```

```

Name: PE1_t0 (Tunnel0) Destination: 10.11.11.11
Status:
  Admin: up      Oper: up      Path: valid      Signalling: connected

  path option 10, type dynamic (Basis for Setup, path weight 30)

Config Parameters:
  Bandwidth: 0 kbps (Global) Priority: 7 7 Affinity: 0x0/0xFFFF
  Metric Type: TE (default)
  AutoRoute: enabled LockDown: disabled Loadshare: 0 bw-based
  auto-bw: disabled

InLabel : -
OutLabel : Ethernet2/0/2, 19
!--- Label 19 from RSVP is used to reach destination 10.11.11.11/32. RSVP Signalling Info: Src
10.2.2.2, Dst 10.11.11.11, Tun_Id 0, Tun_Instance 31 RSVP Path Info: My Address: 10.7.7.2
Explicit Route: 10.7.7.7 10.8.8.7 10.8.8.5 10.12.12.10 10.11.11.11 Record Route: NONE Tspec: ave
rate=0 kbits, burst=1000 bytes, peak rate=0 kbits RSVP Resv Info: Record Route: NONE Fspec: ave
rate=0 kbits, burst=1000 bytes, peak rate=Inf Shortest Unconstrained Path Info: Path Weight: 30
(TE) Explicit Route: 10.7.7.2 10.7.7.7 10.8.8.7 10.8.8.5 10.12.12.10 10.11.11.11 History:
Tunnel: Time since created: 17 hours, 17 minutes Time since path change: 32 minutes, 54 seconds
Current LSP: Uptime: 32 minutes, 54 seconds Prior LSP: ID: path option 10 [14] Removal Trigger:
tunnel shutdown

```

Eine andere Möglichkeit, diese Informationen schnell anzuzeigen, besteht darin, die Ausgabemodifizierer in den Befehlen **show** zu verwenden, wie hier gezeigt:

```

PE1# show mpls traffic-eng tunnels tunnel 0 | include Label
InLabel : -
OutLabel : Ethernet2/0/2, 19
!--- This is the label to reach 10.11.11.11.

```

Sehen Sie sich den Tag-Stack an. Es ist 19, das die TE-Bezeichnung ist, die zum Weiterleiten von Paketen an den nächsten Hop 10.11.11.0 über Tunnel0 verwendet wird.

```
PE1# show tag forwarding-table 10.11.11.11 detail
```



```

Local  Outgoing  Prefix          Bytes tag  Outgoing  Next Hop
tag    tag or VC   or Tunnel Id    switched   interface
21     Pop tag     10.11.11.11/32  0          Tu0        point2point
      MAC/Encaps=14/18, MTU=1500, Tag Stack{19}, via Et2/0/2
      00603E2B02410060835887428847 00013000
      No output feature configured
      Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
PE1#

```

Daher sendet PE1 ein Paket, das für 172.16.13.13 bestimmt ist, mit dem Label-Stack {19 12308}. P1 tauscht das Etikett 19 aus. Das Paket erreicht P2, wodurch das äußere Label geöffnet wird. Anschließend wird das Paket mit dem Label 12308 an PE2 weitergeleitet.

Auf PE2 wird das Paket mit dem Label 12308 empfangen und gemäß den Informationen in der Weiterleitungstabelle umgeschaltet. Hier sehen Sie:

```

PE2# show tag for tags 12308 detail
Local  Outgoing  Prefix          Bytes tag  Outgoing  Next Hop
tag    tag or VC   or Tunnel Id    switched   interface
12308  Aggregate  172.16.13.0/24[V] 12256
      MAC/Encaps=0/0, MTU=0, Tag Stack{}
      VPN route: aqua
      No output feature configured
      Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
PE2#

```

Hinweis: Es wird keine Ausgangsschnittstelle angezeigt, da das ausgehende Tag `Aggregate` ist. Dies liegt daran, dass das Präfix, das der Bezeichnung zugeordnet ist, die direkt verbundene Route ist.

Pings von CE1 zu einem Host auf CE2 bestätigen die VPN-Verbindung über den TE-Tunnel:

```

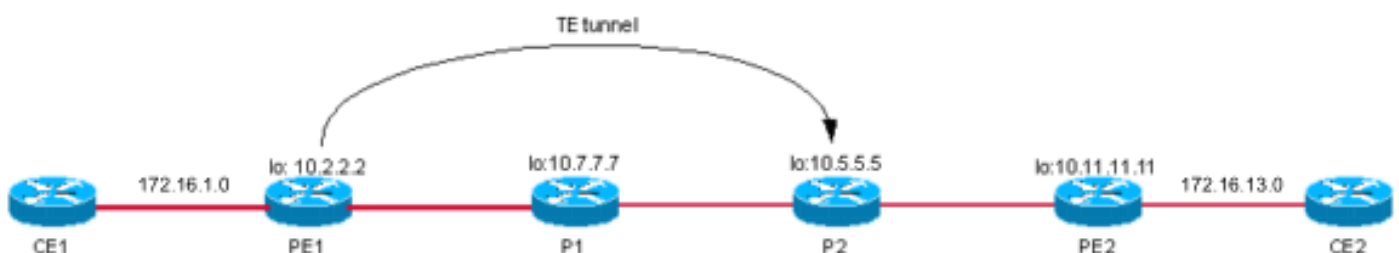
CE1# ping 172.16.13.13

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.13, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/13/36 ms
CE1#

```

Fall 2: VPN über einen TE-Tunnel, wenn der TE-Tunnel vom PE1 zum P2 verläuft

Topologie



Konfiguration

Die zusätzliche TE-Konfiguration über die Basiskonfiguration auf PE1 wird hier angezeigt:

```
PE1
PE1# show run interface tunnel 0
!
interface Tunnel0
 ip unnumbered Loopback0
 no ip directed-broadcast
 no ip route-cache distributed
 tunnel destination 10.5.5.5
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 10 dynamic
end
!
```

Überprüfung

Überprüfen Sie die Route zum Präfix 172.16.13.13 für PE1 VRF-Aqua. Er zeigt auf den nächsten Hop 10.11.11.11/32 (über Tunnel0), wobei der Label-Stack {19 12308} verwendet wird.

```
PE1# show ip cef vrf aqua 172.16.13.13
172.16.13.0/24, version 11
0 packets, 0 bytes
 tag information set
   local tag: VPN route head
   fast tag rewrite with Tu0, point2point, tags imposed {19 12308}
 via 10.11.11.11, 0 dependencies, recursive
   next hop 10.5.5.5, Tunnel0 via 10.11.11.11/32
   valid adjacency
   tag rewrite with Tu0, point2point, tags imposed {19 12308}
PE1#
```

Label 19, das äußere Label, wird verwendet, um den nächsten Hop 10.11.11.11/32 zu erreichen, wie hier gezeigt:

```
PE1# show ip cef 10.11.11.11
10.11.11.11/32, version 37
0 packets, 0 bytes
 tag information set
   local tag: 21
   fast tag rewrite with Tu0, point2point, tags imposed {19}
 via 10.5.5.5, Tunnel0, 1 dependency
   next hop 10.5.5.5, Tunnel0
   valid adjacency
   tag rewrite with Tu0, point2point, tags imposed {19}
```

```
PE1# show mpls traffic-eng tunnels tunnel 0
```

```
Name: PE1_t0 (Tunnel0) Destination: 10.5.5.5
Status:
Admin: up Oper: up Path: valid Signalling: connected

path option 10, type dynamic (Basis for Setup, path weight 20)
```

```
Config Parameters:
  Bandwidth: 0          kbps (Global)  Priority: 7 7  Affinity: 0x0/0xFFFF
  Metric Type: TE (default)
  AutoRoute: enabled   LockDown: disabled Loadshare: 0          bw-based
  auto-bw: disabled
```

```
InLabel : -
```

```
OutLabel : Ethernet2/0/2, 19
```

```
RSVP Signalling Info:
```

```
  Src 10.2.2.2, Dst 10.5.5.5, Tun_Id 0, Tun_Instance 33
```

```
RSVP Path Info:
```

```
  My Address: 10.7.7.2
```

```
  Explicit Route: 10.7.7.7 10.8.8.7 10.8.8.5 10.5.5.5
```

```
  Record Route: NONE
```

```
  Tspec: ave rate=0 kbits, burst=1000 bytes, peak rate=0 kbits
```

```
RSVP Resv Info:
```

```
  Record Route: NONE
```

```
  Fspec: ave rate=0 kbits, burst=1000 bytes, peak rate=Inf
```

```
Shortest Unconstrained Path Info:
```

```
  Path Weight: 20 (TE)
```

```
  Explicit Route: 10.7.7.2 10.7.7.7 10.8.8.7 10.8.8.5
                  10.5.5.5
```

```
History:
```

```
Tunnel:
```

```
  Time since created: 17 hours, 31 minutes
```

```
  Time since path change: 8 minutes, 49 seconds
```

```
Current LSP:
```

```
  Uptime: 8 minutes, 49 seconds
```

```
  Selection: reoptimization
```

```
Prior LSP:
```

```
  ID: path option 10 [31]
```

```
  Removal Trigger: path verification failed
```

```
PE1#
```

```
PE1# show mpls traffic-eng tunnels tunnel 0 | i Label
```

```
  InLabel : -
```

```
  OutLabel : Ethernet2/0/2, 19
```

```
PE1#
```

Das Paket von PE1 wird mit dem Label-Stack {19 12308} über den TE-Tunnel gesendet. Sobald P1 das Paket empfängt, öffnet es (PHP) das Tag 19 und sendet das Paket mit Label Stack {12308}. Der Befehl **show** bestätigt Folgendes:

```
P1> show tag for tag 19
```

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
19	Pop tag	10.2.2.2 0 [33]	2130	Et2/0	10.8.8.5

```
P1>
```

```
P1> show tag for tag 19 detail
```

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
19	Pop tag	10.2.2.2 0 [33]	2257	Et2/0	10.8.8.5

```
  MAC/Encaps=14/14, MTU=1504, Tag Stack{
  006009E08B0300603E2B02408847
  No output feature configured
```

```
P1>
```

Wenn P2 das Paket mit dem Label-Stack {12308} empfängt, überprüft es seinen LFIB und verwirft das Paket, da keine Übereinstimmung vorhanden ist. Dies ist die **Ausgabe des Befehls show** auf P2:

```

P2# show tag forwarding-table tags 12308 detail
Local  Outgoing  Prefix  Bytes tag  Outgoing  Next Hop
tag    tag or VC  or Tunnel Id  switched  interface
P2#

P2#
7w4d: TAG: Et0/3: recvd: CoS=0, TTL=253, Tag(s)=12308
7w4d: TAG: Et0/3: recvd: CoS=0, TTL=253, Tag(s)=12308
7w4d: TAG: Et0/3: recvd: CoS=0, TTL=253, Tag(s)=12308
7w4d: TAG: Et0/3: recvd: CoS=0, TTL=253, Tag(s)=12308
P2#
P2#

```

Erläuterung

Die Lösung für dieses Problem besteht darin, TDP/LDP auf dem TE-Tunnel zu aktivieren und zu einer Tag-Switched-Schnittstelle zu machen. In dem in der [Lösung](#) gezeigten Beispiel ist TDP auf Tunnel0 von PE1 aktiviert. P2 ist so konfiguriert, dass er gezielte Hellos akzeptiert und gezielte TDP-Nachbarn bildet. PE1 empfängt das Label für 10.11.11.11 von P2 über LDP. Nachdem Tunnel0 zu einer Tag-gesteuerten Schnittstelle gemacht wurde und TDP für den Datenverkehr auf 10.11.11.11 aktiviert wurde, verwendet PE1 beide Labels. Es wird das RSVP-Label verwendet, um den TE-Zielpunkt zu erreichen, und das TDP-Label erreicht 10.11.11.11.

In diesem Szenario verwendet PE1 den Label-Stack {L2 L3 L1}, um Daten an CE2 weiterzuleiten, wenn diese Elemente zutreffend sind:

- L1 ist das VPN-Label.
- L2 ist das RSVP-Label, das den TE-Zielanschluss erreicht.
- L3 ist das TDP-Label, das 10.11.11.11 erreicht (von P2 empfangen).

Lösung

Die Lösung besteht darin, TDP über den TE-Tunnel zu aktivieren.

Konfiguration

Hier sehen Sie die TE-Tunnelkonfiguration auf PE1 mit aktiviertem TDP. Die Zusätze sind fett formatiert.

```

PE1

PE1# show run interface tunnel 0
!
interface Tunnel0
 ip unnumbered Loopback0
 no ip directed-broadcast
 no ip route-cache distributed
 tag-switching ip
 !--- This enables TDP. tunnel destination 10.5.5.5
 tunnel mode mpls traffic-eng tunnel mpls traffic-eng
 autoroute announce tunnel mpls traffic-eng path-option
 10 dynamic end !

```

Dies ist die zusätzliche Konfiguration auf dem Heckende des TE-Tunnels, um gezielte TDP-Hellos

zu akzeptieren:

```
P2# show run | i directed-hello
tag-switching tdp discovery directed-hello accept
!--- This configures P2 to accept directed TDP hellos. P2#
```

Überprüfung

```
PE1# show tag tdp neighbor | i Peer
Peer TDP Ident: 10.7.7.7:0; Local TDP Ident 10.2.2.2:0
Peer TDP Ident: 10.5.5.5:0; Local TDP Ident 10.2.2.2:0

PE1#
PE1# show ip cef vrf aqua 172.16.13.13
172.16.13.0/24, version 11
0 packets, 0 bytes
tag information set
local tag: VPN route head
fast tag rewrite with Tu0, point2point, tags imposed {19 18 12308}
via 10.11.11.11, 0 dependencies, recursive
next hop 10.5.5.5, Tunnel0 via 10.11.11.11/32
valid adjacency
tag rewrite with Tu0, point2point, tags imposed {19 18 12308}
PE1#
```

```
PE1# show mpls traffic-eng tunnels tunnel 0 | i Label
InLabel : -
OutLabel : Ethernet2/0/2, 19
!--- This is the TE label learned via RSVP. PE1# PE1# show tag tdp bind 10.11.11.11 32
tib entry: 10.11.11.11/32, rev 20
local binding: tag: 21
remote binding: tsr: 10.7.7.7:0, tag: 17
remote binding: tsr: 10.5.5.5:0, tag: 18
!--- This is the TDP label from P2.
```

Wenn P1 das Paket mit dem Label-Stack {19 18 12308} empfängt, öffnet es das Tag 19 und sendet das Paket mit dem Label-Stack {18 12308} an P2. P2 prüft seinen LFIB auf Label 18, öffnet das Tag und sendet es über die ausgehende Schnittstelle PO2/0/0 an PE1. PE1 empfängt das Paket mit dem Label 12308 und schaltet es erfolgreich auf CE2 um.

```
P2# show tag for tag 18
Local Outgoing Prefix Bytes tag Outgoing Next Hop
tag tag or VC or Tunnel Id switched interface
18 Pop tag 10.11.11.11/32 117496 POS2/0/0 point2point
```

```
P2# show tag tdp discovery
Local TDP Identifier:
10.5.5.5:0
Discovery Sources:
Interfaces:
Ethernet0/3 (tdp): xmit/rcv
TDP Id: 10.7.7.7:0
POS2/0/0 (tdp): xmit/rcv
TDP Id: 10.11.11.11:0
Directed Hellos:
10.5.5.5 -> 10.2.2.2 (tdp): passive, xmit/rcv
TDP Id: 10.2.2.2:0
```

```
P2# show tag tdp neighbor 10.2.2.2
```

```

Peer TDP Ident: 10.2.2.2:0; Local TDP Ident 10.5.5.5:0
TCP connection: 10.2.2.2.711 - 10.5.5.5.11690
State: Oper; PIEs sent/rcvd: 469/465; Downstream
Up time: 01:41:08
TDP discovery sources:
  Directed Hello 10.5.5.5 -> 10.2.2.2, passive
Addresses bound to peer TDP Ident:
  10.7.7.2      172.16.47.166  10.2.2.2

```

```
PE1# show tag tdp neighbor 10.5.5.5
```

```

Peer TDP Ident: 10.5.5.5:0; Local TDP Ident 10.2.2.2:0
TCP connection: 10.5.5.5.11690 - 10.2.2.2.711
State: Oper; PIEs sent/rcvd: 438/441; Downstream
Up time: 01:35:08
TDP discovery sources:
  Directed Hello 10.2.2.2 -> 10.5.5.5, active

```

!--- This indicates the directed neighbor. Addresses bound to peer TDP Ident: 10.5.5.5

```
10.12.12.5 10.8.8.5 PE1# show ip route 10.11.11.11
```

```
Routing entry for 10.11.11.11/32
```

```
Known via "isis", distance 115, metric 40, type level-1
Redistributing via isis
```

```
B Last update from 10.5.5.5 on Tunnel0, 01:52:21 ago
```

```
Routing Descriptor Blocks:
```

```
* 10.5.5.5, from 10.11.11.11, via Tunnel0
```

```
Route metric is 40, traffic share count is 1
```

Ein Ping-Befehl von CE1 zu einem Host auf CE2 bestätigt die Lösung.

```
CE1# ping 172.16.13.13
```

Type escape sequence to abort.

```
Sending 5, 100-byte ICMP Echos to 172.16.13.13, timeout is 2 seconds:
```

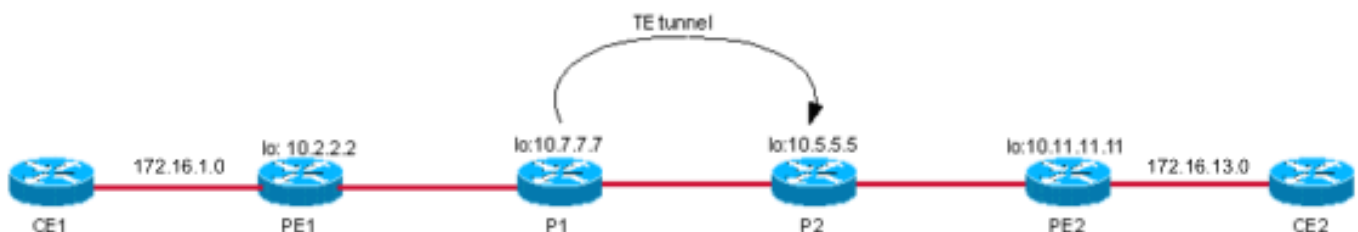
```
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms
```

```
CE1#
```

Fall 3: VPN zwischen CE1 und CE2 über einen TE-Tunnel von P1 bis P2, wenn TDP/LDP nicht aktiviert ist

Topologie



Konfiguration

Die Tunnelkonfiguration auf PE1 wird hier angezeigt:

```
PE1
```

```
P1# show run interface tunnel 0
```

```

Building configuration...

Current configuration : 255 bytes
!
interface Tunnel0
 ip unnumbered Loopback0
 no ip directed-broadcast
 ip route-cache distributed
 tunnel destination 10.5.5.5
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 10 dynamic
end

```

Überprüfung

Überprüfen Sie hier, wie Pakete, die für CE2 172.16.13.13 bestimmt sind, geswitcht werden. Die Ausgabe des Befehls **show ip cef** zeigt, dass Pakete zum Ziel 172.16.13.13 mit dem Label-Stack {17 12308} geswitcht werden:

```

PE1# show ip cef vrf aqua 172.16.13.13
172.16.13.0/24, version 18, cached adjacency 10.7.7.7
0 packets, 0 bytes
 tag information set
   local tag: VPN route head
   fast tag rewrite with Et2/0/2, 10.7.7.7, tags imposed {17 12308}
 via 10.11.11.11, 0 dependencies, recursive
   next hop 10.7.7.7, Ethernet2/0/2 via 10.11.11.11/32
   valid cached adjacency
   tag rewrite with Et2/0/2, 10.7.7.7, tags imposed {17 12308}

```

Wenn P1 dieses Paket empfängt, entfernt es das äußere Label 17 und schaltet das Paket nach der Überprüfung in der IP-Routing-Tabelle auf Tunnel0 um. Beachten Sie das `implicit-Null-OutLabel` in dieser Ausgabe. Das bedeutet, dass die ausgehende Schnittstelle nicht Label Switched ist.

```

P1# show ip cef 10.11.11.11 detail
10.11.11.11/32, version 52
0 packets, 0 bytes
 tag information set
   local tag: 17
   fast tag rewrite with Tu0, point2point, tags imposed {}
 via 10.5.5.5, Tunnel0, 0 dependencies
   next hop 10.5.5.5, Tunnel0
   valid adjacency
   tag rewrite with Tu0, point2point, tags imposed {}

```

```

P1# show mpls traffic-eng tunnel tunnel 0 | i Label
InLabel : -
OutLabel : Ethernet2/0, implicit-null

```

```

P1# show tag for 10.11.11.11 detail
Local  Outgoing  Prefix          Bytes tag  Outgoing  Next Hop
tag    tag or VC  or Tunnel Id    switched  interface
17     Untagged  10.11.11.11/32  882       Tu0       point2point
      MAC/Encaps=14/14, MTU=1500, Tag Stack{}, via Et2/0
      006009E08B0300603E2B02408847
      No output feature configured
Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

```

```

P1# show ip route 10.11.11.11
Routing entry for 10.11.11.11/32
  Known via "isis", distance 115, metric 30, type level-1
  Redistributing via isis
  Last update from 10.5.5.5 on Tunnel0, 00:03:20 ago
  Routing Descriptor Blocks:
  * 10.5.5.5, from 10.11.11.11, via Tunnel0
    Route metric is 30, traffic share count is 1

```

Sobald P2 das Paket mit dem Label 12308 empfängt, wird die Weiterleitungstabelle geprüft. Da P2 den VPN-Tag 12308 von CE2 nicht erkennen kann, wird das Paket verworfen.

```

P2# show tag for tag 12308 detail
Local   Outgoing   Prefix           Bytes tag  Outgoing   Next Hop
tag     tag or VC  or Tunnel Id    switched  interface

```

Dadurch wird der Pfad der für CE2 bestimmten VPN-Pakete unterbrochen. Es wird durch den Ping an CE2 172.16.13.13/32 bestätigt.

```

PE1#
CE1# ping 172.16.13.13

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.13, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
CE1#

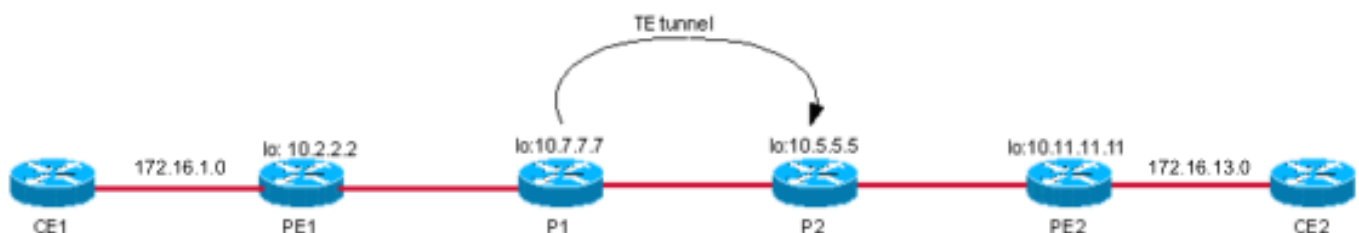
```

Lösung

Die Lösung besteht darin, LDP/TDP über den Tunnel zu aktivieren. Im nächsten Abschnitt wird diese Lösung erläutert.

Fall 4: VPN über einen TE-Tunnel zwischen P1 und P2 mit aktiviertem LDP

Topologie



Konfiguration

Wenn LDP auf dem Tunnel aktiviert ist, werden die Konfigurationen auf P1 wie hier gezeigt angezeigt. Additionen sind fett dargestellt.

PE1


```

P1# show run interface tunnel 0
Building configuration...

Current configuration : 273 bytes
!
interface Tunnel0
 ip unnumbered Loopback0
 no ip directed-broadcast
 ip route-cache distributed
 mpls label protocol ldp
 tunnel destination 10.5.5.5
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 10 dynamic
end
!
```

Überprüfung

PE1 sendet Pakete mit dem Label-Stack {17 12308} an das Präfix 172.16.13.13/32.

```

PE1#
PE1# show tag for 10.11.11.11 detail
Local   Outgoing   Prefix           Bytes tag  Outgoing   Next Hop
tag     tag or VC  or Tunnel Id     switched   interface
21    17      10.11.11.11/32   0          Et2/0/2    10.7.7.7
      MAC/Encaps=14/18, MTU=1500, Tag Stack{17}
      00603E2B02410060835887428847 00011000
      No output feature configured
      Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
```

```

PE1#
PE1# show ip cef 10.11.11.11 detail
10.11.11.11/32, version 60, cached adjacency 10.7.7.7
0 packets, 0 bytes
 tag information set
  local tag: 21
  fast tag rewrite with Et2/0/2, 10.7.7.7, tags imposed {17}
  via 10.7.7.7, Ethernet2/0/2, 1 dependency
  next hop 10.7.7.7, Ethernet2/0/2
  valid cached adjacency
  tag rewrite with Et2/0/2, 10.7.7.7, tags imposed {17}
```

```

PE1# show ip cef vrf aqua 172.16.13.13
172.16.13.0/24, version 18, cached adjacency 10.7.7.7
0 packets, 0 bytes
 tag information set
  local tag: VPN route head
  fast tag rewrite with Et2/0/2, 10.7.7.7, tags imposed {17 12308}
  via 10.11.11.11, 0 dependencies, recursive
  next hop 10.7.7.7, Ethernet2/0/2 via 10.11.11.11/32
  valid cached adjacency
  tag rewrite with Et2/0/2, 10.7.7.7, tags imposed {17 12308}
```

P1 empfängt das Paket mit dem Label-Stack {17 12308} und prüft seinen LFIB für Label 17.

```

P1# show tag for tag 17 detail
Local   Outgoing   Prefix           Bytes tag  Outgoing   Next Hop
tag     tag or VC  or Tunnel Id     switched   interface
```

```

17      18      10.11.11.11/32    1158      Tu0      point2point
      MAC/Encaps=14/18, MTU=1496, Tag Stack{18}, via Et2/0
      006009E08B0300603E2B02408847 00012000
      No output feature configured
      Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
P1#

```

```

P1# show ip cef 10.11.11.11 detail
10.11.11.11/32, version 52
0 packets, 0 bytes
tag information set
  local tag: 17
  fast tag rewrite with Tu0, point2point, tags imposed {18}
via 10.5.5.5, Tunnel0, 0 dependencies
  next hop 10.5.5.5, Tunnel0
  valid adjacency
  tag rewrite with Tu0, point2point, tags imposed {18}

```

Es zeigt, dass das Etikett 17 gegen das Etikett 18 ausgetauscht werden sollte. Daher wird dieses Paket über die Tunnelschnittstelle mit dem Label-Stack {18 12308} umgeschaltet.

P2 empfängt das Paket über die Tunnelschnittstelle mit dem Label-Stack {18 12308}. Es öffnet das Tag 18 (da es sich um den vorletzten Hop-Router handelt) und schaltet das Paket mit dem Label 12308 auf PE2 um.

```

P2# show tag for tag 18 detail
Local  Outgoing  Prefix          Bytes tag  Outgoing  Next Hop
tag    tag or VC  or Tunnel Id    switched   interface
18    Pop tag   10.11.11.11/32  127645     PO2/0/0   point2point
      MAC/Encaps=4/4, MTU=4474, Tag Stack{}
      0F008847
      No output feature configured
      Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
P2#

```

PE2 empfängt das Paket mit dem Label 12308, das das Paket erfolgreich auf CE2 umschaltet.

```

PE2# show tag forwarding tags 12308 detail
Local  Outgoing  Prefix          Bytes tag  Outgoing  Next Hop
tag    tag or VC  or Tunnel Id    switched   interface
12308 Aggregate 172.16.13.0/24[V] 12256
      MAC/Encaps=0/0, MTU=0, Tag Stack{}
      VPN route: aqua
      No output feature configured
      Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
PE2#

```

```

CE1# ping 172.16.13.13

```

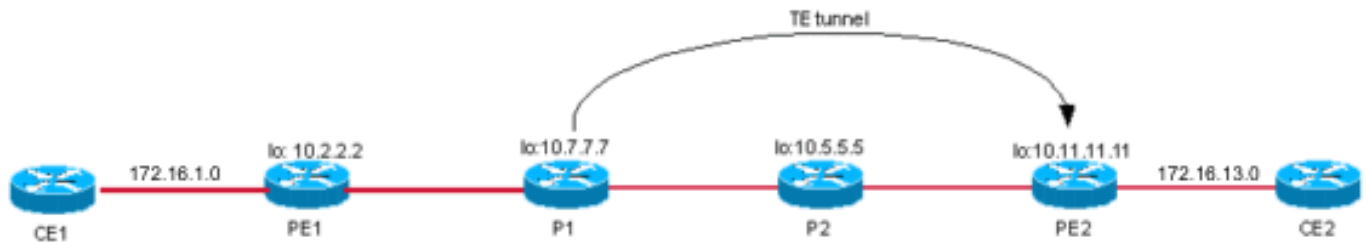
```

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.13, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
CE1#

```

Fall 5: MPLS-VPN über einen Tunnel zwischen P1 und PE2

Topologie



Konfiguration

PE1

```
P1# show run interface tunnel 0
Building configuration...

Current configuration : 258 bytes
!
interface Tunnel0
 ip unnumbered Loopback0
 no ip directed-broadcast
 ip route-cache distributed
 tunnel destination 10.11.11.11
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 10 dynamic
end
```

Überprüfung

PE1 sendet ein Paket, das für 172.16.13.13 bestimmt ist, an den nächsten Hop 10.11.11.11 mit dem Label-Stack {17 12308}.

```
PE1# show ip cef vrf aqua 172.16.13.13
172.16.13.0/24, version 18, cached adjacency 10.7.7.7
0 packets, 0 bytes
 tag information set
   local tag: VPN route head
   fast tag rewrite with Et2/0/2, 10.7.7.7, tags imposed {17 12308}
 via 10.11.11.11, 0 dependencies, recursive
   next hop 10.7.7.7, Ethernet2/0/2 via 10.11.11.11/32
   valid cached adjacency
   tag rewrite with Et2/0/2, 10.7.7.7, tags imposed {17 12308}
```

P1 empfängt das Paket mit dem Label-Stack {17 12308}. P1 überprüft seine LFIB-Tabelle und prüft den Tag-Stack {17} und schaltet das Paket mit dem Label {17} in Richtung P2.

```
P1# show tag for 10.11.11.11 detail
Local  Outgoing  Prefix          Bytes tag  Outgoing  Next Hop
tag    tag or VC  or Tunnel Id    switched  interface
17     Untagged  10.11.11.11/32  411       Tu0       point2point
      MAC/Encaps=14/18, MTU=1500, Tag Stack{17}, via Et2/0
      006009E08B0300603E2B02408847 00011000
      No output feature configured
```

Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

P1# **show tag for tag 17 detail**

```
Local  Outgoing  Prefix          Bytes tag  Outgoing  Next Hop
tag    tag or VC   or Tunnel Id    switched   interface
17    Untagged   10.11.11.11/32  685        Tu0        point2point
      MAC/Encaps=14/18, MTU=1500, Tag Stack{17}, via Et2/0
      006009E08B0300603E2B02408847 00011000
      No output feature configured
```

Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

P1#

P1# **show ip cef 10.11.11.11**

```
10.11.11.11/32, version 67
0 packets, 0 bytes
tag information set
local tag: 17
fast tag rewrite with Tu0, point2point, tags imposed {17}
via 10.11.11.11, Tunnel0, 0 dependencies
next hop 10.11.11.11, Tunnel0
valid adjacency
tag rewrite with Tu0, point2point, tags imposed {17}
```

P2 empfängt das Paket mit Label-Stack {17 12308}. P2 ist der vorletzte Hop-Router und Pops-Label 17.

P2# **show tag for tag 17 detail**

```
Local  Outgoing  Prefix          Bytes tag  Outgoing  Next Hop
tag    tag or VC   or Tunnel Id    switched   interface
17    Pop tag   10.7.7.7 0 [5]  535        P02/0/0    point2point
      MAC/Encaps=4/4, MTU=4474, Tag Stack{}
      0F008847
      No output feature configured
```

P2#

PE2 empfängt dann das Paket mit dem Label 12308. PE2 ist sich bewusst, dass das Ziel für das Label 12308 direkt verbunden ist. Aus diesem Grund ist der Ping von CE1 zu CE2 10.

PE2# **show tag for tag 12308 detail**

```
Local  Outgoing  Prefix          Bytes tag  Outgoing  Next Hop
tag    tag or VC   or Tunnel Id    switched   interface
12308 Aggregate 172.16.13.0/24[V] 12776
      MAC/Encaps=0/0, MTU=0, Tag Stack{}
      VPN route: aqua
      No output feature configured
```

Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

PE2#

Hinweis: Es wird keine Ausgangsschnittstelle angezeigt, da das ausgehende Tag Aggregate ist. Dies liegt daran, dass das Präfix, das der Bezeichnung zugeordnet ist, die direkt verbundene Route ist.

CE1# **ping 172.16.13.13**

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.16.13.13, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms

CE1#

Bekannte Probleme

Siehe [Problemhinweis: MPLS VPN mit TE und MPLS InterAS Beratung zur Cisco IOS® Software](#) für weitere Details.

Schlussfolgerung

Wenn der TE-Tunnel am Ausgangs-PE terminiert wird, arbeiten MPLS VPN und TE ohne zusätzliche Konfiguration zusammen. Wenn der TE-Tunnel auf einem P-Router terminiert wird (vor dem PE im Core), schlägt die Weiterleitung des MPLS-VPN-Datenverkehrs fehl, da Pakete mit VPN-Labels als äußere Beschriftungen eingehen, die sich nicht in den LFIBs dieser Geräte befinden. Daher können diese zwischengeschalteten Router Pakete nicht an das Endziel, das VPN-Kundennetzwerk, weiterleiten. In diesem Fall sollte LDP/TDP auf dem TE-Tunnel aktiviert werden, um das Problem zu beheben.

Zugehörige Informationen

- [Häufig gestellte Fragen zu MPLS für Anfänger](#)
- [Fehlerbehebung für MPLS-VPN](#)
- [MPLS Basic Traffic Engineering mit OSPF-Konfigurationsbeispiel](#)
- [Konfigurieren eines einfachen MPLS-VPN](#)
- [Fehlerbehebung bei LSP-Ausfällen im MPLS-VPN](#)
- [MPLS-Support-Seite](#)
- [Technischer Support - Cisco Systems](#)