

# Implementar un MPLS VPN sobre los túneles TE

## Contenido

[Introducción](#)

[prerrequisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Convenciones](#)

[Teoría Precedente](#)

[Configuración de VPN inicial entre CE1 y CE2 sin un túnel TE](#)

[Topología](#)

[Configuración](#)

[Verificación](#)

[Caso 1: VPN sobre túnel TE cuando túnel TE es de PE1 al PE2](#)

[Topología](#)

[Configuración](#)

[Verificación](#)

[Caso 2: VPN sobre túnel TE cuando túnel TE es de PE1 al P2](#)

[Topología](#)

[Configuración](#)

[Verificación](#)

[Explicación](#)

[Solución](#)

[Caso 3: VPN entre el CE1 y el CE2 sobre a túnel TE del P1 al P2 cuando el TDP/LDP no se habilita](#)

[Topología](#)

[Configuración](#)

[Verificación](#)

[Solución](#)

[Caso 4: VPN sobre a túnel TE entre el P1 y el P2 con el LDP habilitado](#)

[Topología](#)

[Configuración](#)

[Verificación](#)

[Caso 5: MPLS VPN sobre un túnel entre el P1 y el PE2](#)

[Topología](#)

[Configuración](#)

[Verificación](#)

[Problemas conocidos](#)

[Conclusión](#)

[Información Relacionada](#)

## Introducción

Este documento proporciona configuraciones de ejemplo para implementar una VPN MPLS (Multiprotocol Label Switching) sobre los túneles de ingeniería de tráfico (TE) en una red MPLS. Para ganar las ventajas de un MPLS VPN sobre los túneles TE, ambos deben coexistir en la red. Este documento ilustra los diversos escenarios que explican por qué el reenvío de paquete dentro de un MPLS VPN sobre los túneles TE pudo fallar. También proporciona una solución posible.

## prerrequisitos

### Requisitos

Quienes lean este documento deben tener conocimiento de los siguientes temas:

- [Ingeniería de tráfico MPLS y mejoras](#)
- [Configuración de una VPN MPLS básica](#)

### Componentes Utilizados

Este documento no tiene restricciones específicas en cuanto a versiones de software y de hardware.

### Convenciones

Consulte [Convenciones de Consejos TécnicosCisco](#) para obtener más información sobre las convenciones del documento.

### Teoría Precedente



Tal y como se muestra en de esta topología, en una configuración simple del MPLS VPN, el borde 1 (PE1) del proveedor aprende la escritura de la etiqueta VPN (escritura de la etiqueta 1 [L1]) para el prefijo 172.16.13.0/24 VPN vía el protocolo Protocolo de la puerta de enlace marginal (BGP) multiprotocol (MPBGP) del PE2 directamente, con el salto siguiente como el Loopback Address PE2. El PE1 también aprende la escritura de la etiqueta (L2) para el Loopback Address PE2 vía el Protocolo de distribución de etiquetas (LDP) de su salto siguiente P1.

Cuando los reenvíos de datos al VPN prefijan 172.16.13.13, el PE1 utiliza una pila de etiquetas {L2 L1} con el L2 como la escritura de la etiqueta externa. L2 es intercambiada por el router de switch de etiqueta (LSR) de tránsito, P1. El P2 hace estallar el L2 externo y adelante el paquete al PE2 con solamente un L1. Para entender mejor por qué el P2 hace estallar el L2, refiera a la sección 3.16 sobre el Penultimate Hop Popping (PHP) en el [RFC 3031](#) . [Así, los paquetes al prefijo 4 \(IPv4\) VPN versión IP 172.16.13.0/24 son escritura de la etiqueta conmutada sobre una red MPLS.](#)

El funcionamiento de reenvío del MPLS VPN falla si cualquier router P recibe el paquete con L1 (escritura de la etiqueta VPN) como la única escritura de la etiqueta externa en vez de la pila de etiquetas {L2 L1}. Esto ocurre porque ninguno de los Routers P tiene L1 en su Base de información de reenvío de etiquetas (LFIB) para conmutar el paquete.

Un MPLS TE utiliza el Resource Reservation Protocol (RSVP) para intercambiar las escrituras de la etiqueta. El router recibe diferentes nombres de LDP y RSVP para un prefijo dado cuando se lo configura tanto para TE como para el Protocolo de distribución de etiqueta (TDP)/LDP. Las escrituras de la etiqueta del LDP y de RSVP no necesitan ser lo mismo en todas las situaciones. El router instala una escritura de la etiqueta LDP en la tabla de reenvío si el prefijo es docto a través de una interfaz LDP, y instala la escritura de la etiqueta de RSVP en la tabla de reenvío si el prefijo es docto sobre túnel TE una interfaz.

En el caso de un llano túnel TE (sin el LDP/TDP habilitado en el túnel), el ingreso LSR (el LSR en el headend del túnel TE) utiliza la misma escritura de la etiqueta que se utiliza para alcanzar el tailend del túnel TE para todas las rutas que son doctas con a túnel TE.

Por ejemplo, hay un túnel TE de PE1 a P2 que está aprendiendo el prefijo 10.11.11.11/32 a través del túnel. La cola extremo del túnel del P2 es 10.5.5.5, y la etiqueta para alcanzar 10.5.5.5 en el PE1 es L3. El PE1 entonces utiliza el L3 para alcanzar el destino 10.11.11.11/32, aprendido sobre túnel TE.

En el escenario [arriba](#), cuando hay a túnel TE entre el PE1 y el P2, considere esos adelante datos PE1 a la frontera del cliente 2 (CE2). Si el L4 es la escritura de la etiqueta VPN, PE1 adelante los datos con la pila de etiquetas {L3 L4}. El P1 hace estallar el L3, y el P2 recibe el paquete con el L4. PE2 es el único LSR que puede reenviar correctamente el paquete con la etiqueta externa L4. El P2 no tiene una sesión MPBGP con el PE2, así que no recibe el L4 del PE2. Por lo tanto, el P2 no tiene ningún conocimiento del L2, y cae el paquete.

Las configuraciones y las salidas de la **demonstración** que siguen demuestran esto e ilustran una Solución posible a este problema.

## [Configuración de VPN inicial entre CE1 y CE2 sin un túnel TE](#)

### [Topología](#)



### [Configuración](#)

Solamente incluyen a las partes pertinentes de los archivos de configuración aquí:

```
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
!

```

## Verificación

El PE2 aprende el prefijo 172.16.1.0/24 del IPv4 PE1 VPN sobre el peering MPBGP entre el PE1 y el PE2. Esto se muestra aquí:

```
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
```

Semejantemente, el PE1 aprende el prefijo 172.16.13.0/24 del IPv4 PE2 VPN sobre el peering

## MPBGP entre el PE1 y el PE2. Esto se muestra aquí:

```
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.
```

Así, el CE1 puede alcanzar 172.16.13.13 en la red CE2 vía el caso “aguamarina” del VPN Routing and Forwarding (VRF), que se configura en el PE1 usando la pila de etiquetas {17 12308}, como se muestra arriba.

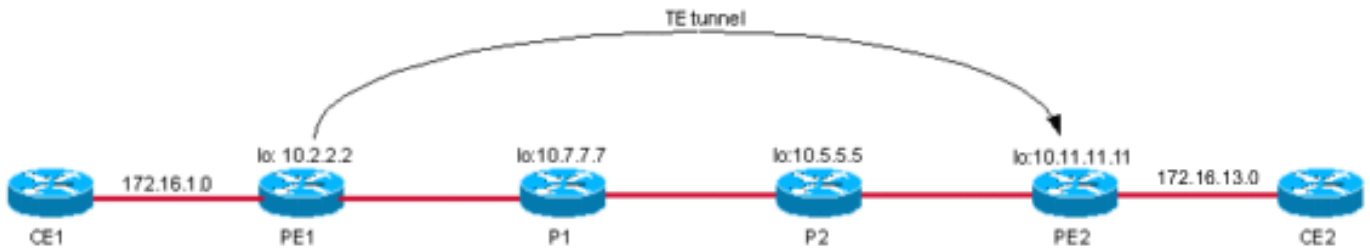
Esta salida del ping confirma la Conectividad:

```
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
```

# Caso 1: VPN sobre túnel TE cuando túnel TE es de PE1 al PE2

## Topología



Cuando túnel TE se construye entre el Routers PE con el autoroute anuncio utilizado, el salto siguiente de la salida PE BGP es accesible vía túnel TE la interfaz. Así, el PE1 utiliza la escritura de la etiqueta TE para alcanzar el PE2.

**Nota:** El MPLS TE es independiente del LDP, así que significa que, si usted tiene una interconexión total de los túneles del PE al PE, usted puede inhabilitar con eficacia el LDP en el Routers y no necesita ejecutar el LDP en túnel TE las interfaces. Sin embargo, usted debe construir todos los túneles al salto siguiente BGP de las rutas del VPN versión 4 (VPNv4). En el ejemplo en esta [configuración](#), usted puede ver que este salto siguiente BGP es el loopback0 en el PE2, 10.11.11.11. Este mismo loopback es también el destino del túnel para el túnel del PE1 al PE2. Esto explica porqué, en este ejemplo, si hay también un túnel del PE2 al PE1 para el tráfico de retorno, usted puede inhabilitar el LDP en la base. Entonces, remitiendo del CE a los trabajos CE con todo el tráfico del VPNv4 transportó los túneles TE. Si el salto siguiente BGP no es lo mismo que túnel TE el destino, el LDP se debe ejecutar en la base y en túnel TE.

## Configuración

La configuración adicional en el PE1 para establecer un túnel PE se muestra aquí:

```
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
```

## Verificación

```
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

```

Ahora, confirme la etiqueta exterior utilizada para alcanzar el siguiente salto 10.11.11.11 a través del túnel0.

```
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

```

Otra manera de ver esta información es rápidamente utilizar a los modificadores de resultado en los comandos show, como se muestra aquí:

```

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.

```

Observe la pila de etiquetas. Es 19, que corresponde a la etiqueta TE, utilizada para reenviar paquetes al salto siguiente 10.11.11.0 por Tunnel0.

```

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#

```

Así, el PE1 envía un paquete destinado a 172.16.13.13 con la pila de etiquetas {19 12308}. El P1 intercambia la escritura de la etiqueta 19. El paquete alcanza el P2, que hace estallar que externo etiqueta. Entonces, el paquete se remite al PE2 con solamente la escritura de la etiqueta 12308.

En el PE2, el paquete con la escritura de la etiqueta 12308 se recibe y se conmuta según la información en la tabla de reenvío. Esto se muestra aquí:

```

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#

```

**Nota:** No se muestra ninguna interfaz saliente porque la etiqueta saliente es agregado. Esto es porque el prefijo asociado a la escritura de la etiqueta es directamente el Routeconectad.

Los ping del CE1 a un host en el CE2 confirman la conectividad VPN sobre túnel TE:

```

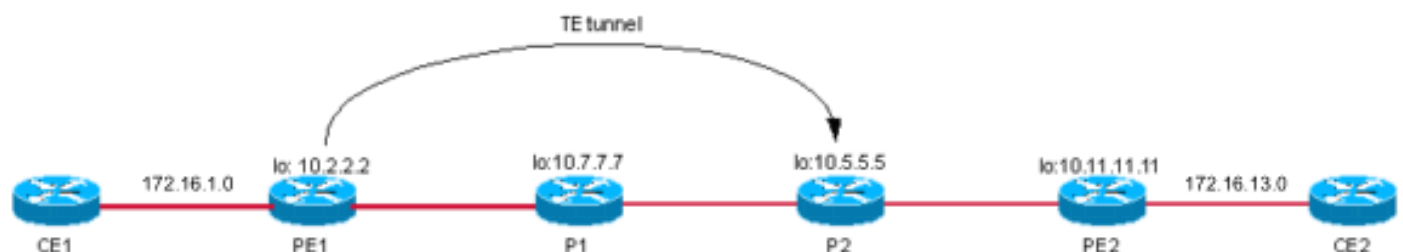
CE1# ping 172.16.13.13

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#

```

## Caso 2: VPN sobre túnel TE cuando túnel TE es de PE1 al P2

### Topología



### Configuración

La configuración adicional TE sobre la configuración básica en el PE1 se muestra aquí:

## 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  
!
```

## Verificación

Marque la ruta para prefijar 172.16.13.13 en los VRF aquums PE1. Señala al salto siguiente 10.11.11.11/32 (sobre el tunnel0) usando la pila de etiquetas {19 12308}.

```
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#
```

La escritura de la etiqueta 19, la escritura de la etiqueta externa, se utiliza para alcanzar el salto siguiente 10.11.11.11/32, como se muestra aquí:

```
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#

```

El paquete del PE1 se envía sobre túnel TE con la pila de etiquetas {19 12308}. Una vez que el P1 recibe el paquete, hace estallar (PHP) la etiqueta 19 y envía el paquete con la pila de etiquetas {12308}. El comando show confirma esto:

```

P1> show tag for tag 19
Local  Outgoing  Prefix          Bytes tag  Outgoing  Next Hop
tag    tag or VC  or Tunnel Id    switched   interface
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  Outgoing  Prefix          Bytes tag  Outgoing  Next Hop
tag    tag or VC  or Tunnel Id    switched   interface
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>

```

Cuando el P2 recibe el paquete con la pila de etiquetas {12308}, marca su LFIB y cae el paquete porque existe ninguna coincidencia. Ésta es la salida del comando show en el 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#
```

## Explicación

La solución a este problema es activar el TDP/LDP en el túnel TE y transformarlo en una interfaz conmutada por etiqueta. En el ejemplo mostrado en la [solución](#), el TDP se habilita en el tunnel0 del PE1. El P2 se configura para validar el hellos dirigido y formar a los vecinos TDP dirigidos. Así pues, el PE1 recibe la escritura de la etiqueta para 10.11.11.11 del P2 vía el LDP. Ahora que el tunnel0 se ha hecho una interfaz etiqueta-conmutada y el TDP se ha habilitado para el tráfico a 10.11.11.11, el PE1 utiliza ambas las escrituras de la etiqueta; utiliza la escritura de la etiqueta de RSVP para alcanzar el extremo de cola TE y la escritura de la etiqueta TDP para alcanzar 10.11.11.11.

En este escenario, el PE1 utiliza la pila de etiquetas {L2 L3 L1} para remitir los datos al CE2 si estos elementos son verdades:

- El L1 es la escritura de la etiqueta VPN.
- El L2 es la escritura de la etiqueta de RSVP para alcanzar el extremo de cola TE.
- El L3 es la escritura de la etiqueta TDP para alcanzar 10.11.11.11 (recibido del P2).

## Solución

La solución es habilitar TDP a través del túnel TE.

## Configuración

Se muestra aquí túnel TE la configuración en el PE1 con el TDP habilitado en ella. Las adiciones están en la negrilla.

```
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 !
```

Ésta es la configuración adicional en el tailend del túnel TE para validar el hellos dirigido TDP:

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

## Verificación

```
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.
```

Cuando el P1 recibe el paquete con la pila de etiquetas {19 18 12308}, hace estallar la etiqueta 19 y envía el paquete con la pila de etiquetas {18 12308} al P2. El P2 marca su LFIB para la escritura de la etiqueta 18, después hace estallar la etiqueta y la envía sobre la interfaz saliente PO2/0/0 hacia el PE1. PE1 recibe el paquete con la etiqueta 12308, y lo conmuta correctamente a CE2.

```
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
```

Un comando ping del CE1 a un host en el CE2 confirma la solución.

```
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:
```

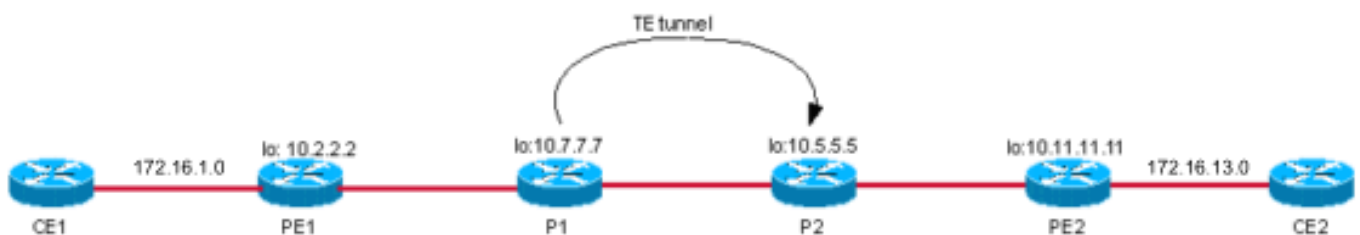
```
!!!!!
```

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

```
CE1#
```

## Caso 3: VPN entre el CE1 y el CE2 sobre a túnel TE del P1 al P2 cuando el TDP/LDP no se habilita

### Topología



### Configuración

La configuración del túnel en el PE1 se muestra aquí:

#### 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
```

## Verificación

Verifique cómo los paquetes destinados a CE2 172.16.13.13 consiguen conmutados aquí. La salida del comando **show ip cef** muestra que los paquetes al destino 172.16.13.13 están conmutados con la pila de etiquetas {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}
```

Cuando el P1 recibe este paquete, quita la escritura de la etiqueta externa 17 y conmuta el paquete después de mirar en la tabla de IP Routing al tunnel0. Note el `OutLabel` implícito-nulo en esta salida; significa que la interfaz saliente no es escritura de la etiqueta conmutada.

```
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
```

Una vez que P2 recibe el paquete con la etiqueta 12308, observa su tabla de reenvío. Porque hay ninguna manera que el P2 puede ser consciente de la etiqueta 12308 VPN del CE2, él cae el paquete.

```
P2# show tag for tag 12308 detail
```

```
Local Outgoing Prefix Bytes tag Outgoing Next Hop
tag tag or VC or Tunnel Id switched interface
```

Esto rompe el trayecto de paquetes VPN destinado al CE2. Es confirmada por el ping a CE2 172.16.13.13/32.

```
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)
```

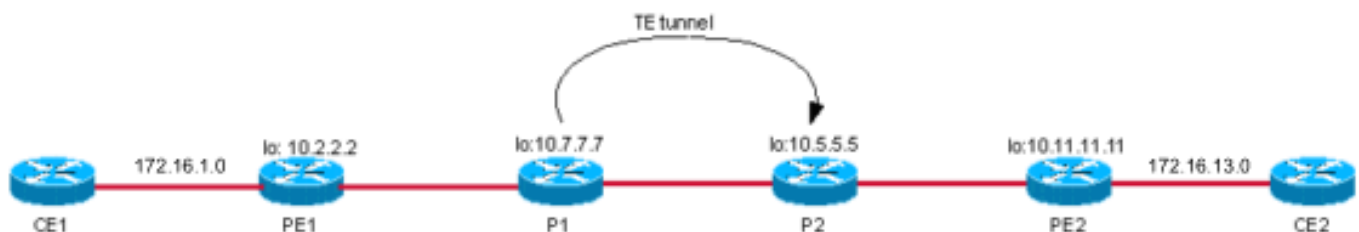
```
CE1#
```

## Solución

La solución es habilitar el LDP/TDP por el túnel. La próxima sección se refiere a esta solución.

## Caso 4: VPN sobre a túnel TE entre el P1 y el P2 con el LDP habilitado

### Topología



### Configuración

Con el LDP habilitado en el túnel, las configuraciones en el P1 aparecen como se muestra aquí. Las adiciones están en la negrilla.

#### 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
!
```

## Verificación

El PE1 envía los paquetes para prefijar 172.16.13.13/32 con la pila de etiquetas {17 12308}.

```

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}
```

El P1 recibe el paquete con la pila de etiquetas {17 12308} y mira su LFIB para la escritura de la etiqueta 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}

```

Muestra que la escritura de la etiqueta 17 se debe intercambiar para etiquetar 18. Por lo tanto, ese paquete se conmuta sobre la interfaz del túnel con la pila de etiquetas {18 12308}.

El P2 recibe el paquete sobre su interfaz del túnel con la pila de etiquetas {18 12308}. Elimina la etiqueta 18 (porque es el penúltimo router de saltos) y cambia el paquete a PE2 con etiqueta 12308.

```

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#

```

El PE2 recibe el paquete con la escritura de la etiqueta 12308, que conmuta el paquete al CE2 con éxito.

```

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

```

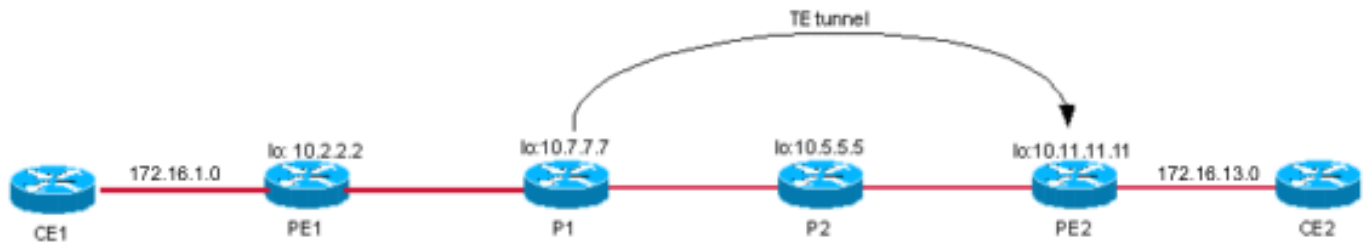
```

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#

```

## [Caso 5: MPLS VPN sobre un túnel entre el P1 y el PE2](#)

### [Topología](#)



## Configuración

### 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

```

## Verificación

El PE1 envía un paquete destinado a 172.16.13.13 a su salto siguiente 10.11.11.11 con la pila de etiquetas {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}

```

El P1 recibe el paquete con la pila de etiquetas {17 12308}. El P1 mira su tabla LFIB y marca el stack de la etiqueta {17} y conmuta el paquete con la escritura de la etiqueta {17} hacia el 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}
```

El P2 recibe el paquete con la pila de etiquetas {17 12308}. El P2, como es el penúltimo router de saltos, salta la etiqueta 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        PO2/0/0    point2point
      MAC/Encaps=4/4, MTU=4474, Tag Stack{}
      0F008847
      No output feature configured
```

P2#

El PE2 entonces recibe el paquete con la escritura de la etiqueta 12308. El P2 es consciente que el destino para la escritura de la etiqueta 12308 está conectado directamente. Por lo tanto, el ping del CE1 al CE2 es 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#

**Nota:** No se muestra ninguna interfaz saliente porque la etiqueta saliente es agregado. Esto es porque el prefijo asociado a la escritura de la etiqueta es directamente el Routeconectad.

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#

## Problemas conocidos

Consulte [Notificación: MPLS VPN con el Advisory TE y MPLS InterAS en el software de Cisco IOS®](#) para más detalles.

## Conclusión

Cuando se termina el túnel TE en el PE de egreso, la VPN de MPLS y el TE funcionan juntos sin necesidad de configuración adicional. Cuando túnel TE se termina en cualquier Routers P (antes del PE en la base), el reenvío de tráfico del MPLS VPN falla porque los paquetes llegan con las escrituras de la etiqueta VPN como las escrituras de la etiqueta externas, que no están en los LFIB de estos dispositivos. Por lo tanto, estos routers intermedios no pueden reenviar paquetes al destino final, la red VPN del cliente. En tal caso, el LDP/TDP se debe habilitar en túnel TE para solucionar el problema.

## Información Relacionada

- [Preguntas frecuentes sobre MPLS para principiantes](#)
- [Cómo solucionar problemas de VPN MPLS](#)
- [Ingeniería básica de tráfico MPLS utilizando el ejemplo de configuración OSPF](#)
- [Configuración de una VPN MPLS básica](#)
- [Solución de problemas de errores de LSP en MPLS VPN](#)
- [MPLS Support Page](#)
- [Soporte Técnico - Cisco Systems](#)