

Implementando uma VPN MPLS sobre túneis TE

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Conventions](#)

[Material de Suporte](#)

[Configuração VPN inicial entre CE1 e CE2 sem um túnel TE](#)

[Topologia](#)

[Configuração](#)

[Verificação](#)

[Caso 1: VPN sobre um túnel TE quando o túnel TE é de PE1 a PE2](#)

[Topologia](#)

[Configuração](#)

[Verificação](#)

[Caso 2: VPN sobre um túnel TE quando o túnel TE é de PE1 a P2](#)

[Topologia](#)

[Configuração](#)

[Verificação](#)

[Explicação](#)

[Solução](#)

[Caso 3: VPN entre CE1 e CE2 sobre um túnel TE de P1 para P2 quando TDP/LDP não está ativado](#)

[Topologia](#)

[Configuração](#)

[Verificação](#)

[Solução](#)

[Caso 4: VPN sobre um túnel TE entre P1 e P2 com LDP ativado](#)

[Topologia](#)

[Configuração](#)

[Verificação](#)

[Caso 5: VPN MPLS sobre um túnel entre P1 e PE2](#)

[Topologia](#)

[Configuração](#)

[Verificação](#)

[Problemas conhecidos](#)

[Conclusão](#)

[Informações Relacionadas](#)

Introduction

Este documento fornece configurações de exemplo para implantação de uma Multiprotocol Label Switching (MPLS) VPN sobre túneis da engenharia de tráfego (TE) em uma rede MPLS. Para obter os benefícios de uma VPN MPLS sobre túneis TE, ambos devem coexistir na rede. Este documento ilustra vários cenários que explicam por que o encaminhamento de pacotes dentro de uma VPN MPLS sobre túneis TE pode falhar. Também fornece uma possível solução.

Prerequisites

Requirements

Os leitores deste documento devem estar cientes destes tópicos:

- [Engenharia e aprimoramentos de tráfego MPLS](#)
- [Configurando uma VPN MPLS básica](#)

Componentes Utilizados

Este documento não se restringe a versões de software e hardware específicas.

Conventions

Consulte as [Convenções de Dicas Técnicas da Cisco para obter mais informações sobre convenções de documentos](#).

Material de Suporte



Como mostrado nesta topologia, em uma configuração simples de VPN MPLS, o Provider Edge 1 (PE1) aprende o rótulo de VPN (Label 1 [L1]) para o prefixo de VPN 172.16.13.0/24 via Multiprotocol Border Gateway Protocol (MPBGP) diretamente de PE2, com o próximo salto como o endereço de loopback PE2. O PE1 também aprende a etiqueta (L2) do endereço de loopback PE2 através do Protocolo de Distribuição de Rótulo (LDP - Label Distribution Protocol) de seu salto seguinte P1.

Ao encaminhar dados para o prefixo VPN 172.16.13.13, PE1 usa uma pilha de rótulos {L2 L1} com L2 como rótulo externo. A L2 é trocada pelo Roteador de Switch de Rótulo (LSR) de trânsito, P1. P2 abre o L2 externo e encaminha o pacote para PE2 com apenas um L1. Para entender melhor por que P2 estoura L2, consulte a seção 3.16 sobre o penúltimo salto (PHP) no [RFC 3031](#). Assim, os pacotes para o prefixo 172.16.13.0/24 do VPN IP versão 4 (IPv4) são comutados por uma rede MPLS.

A operação de encaminhamento VPN MPLS falha se qualquer roteador P receber o pacote com

L1 (rótulo VPN) como o único rótulo externo em vez da pilha de rótulos {L2 L1}. Isso ocorre porque nenhum dos roteadores P tem L1 em sua base de informações de encaminhamento de rótulo (LFIB) para comutar o pacote.

Um TE MPLS usa o Protocolo de Reserva de Recursos (RSVP - Resource Reservation Protocol) para trocar rótulos. Quando um roteador é configurado com TE e Protocolo TDP/LDP, o roteador recebe rótulos diferentes em LDP e RSVP para um determinado prefixo. Os rótulos do LDP e do RSVP não precisam ser os mesmos em todas as situações. O roteador instala um rótulo LDP na tabela de encaminhamento se o prefixo for aprendido por meio de uma interface LDP e instala o rótulo RSVP na tabela de encaminhamento se o prefixo for aprendido sobre uma interface de túnel TE.

No caso de um túnel TE simples (sem LDP/TDP ativado no túnel), o LSR de entrada (o LSR no headend do túnel TE) usa o mesmo rótulo que é usado para alcançar o tailend do túnel TE para todas as rotas aprendidas através de um túnel TE.

Por exemplo, há um túnel TE de PE1 para P2 aprendendo o prefixo 10.11.11.11/32 através do túnel. A extremidade do túnel em P2 é 10.5.5.5 e o rótulo para alcançar 10.5.5.5 em PE1 é L3. O PE1 então usa L3 para alcançar o destino 10.11.11.11/32, aprendido pelo túnel TE.

No cenário [acima](#), quando houver um túnel TE entre PE1 e P2, considere que PE1 encaminha dados para o Customer Edge 2 (CE2). Se L4 for o rótulo VPN, PE1 encaminhará os dados com a pilha de rótulos {L3 L4}. P1 exibe L3 e P2 recebe o pacote com L4. O PE2 é o único LSR que pode encaminhar corretamente o pacote com L4 de rótulo externo. P2 não tem uma sessão de MPBGP com PE2, portanto não recebe o L4 de PE2. Portanto, P2 não tem nenhum conhecimento de L2 e descarta o pacote.

As configurações e os **resultados de exibição** a seguir demonstram isso e ilustram uma possível solução para esse problema.

Configuração VPN inicial entre CE1 e CE2 sem um túnel TE

Topologia



Configuração

Apenas as partes relevantes dos arquivos de configuração são incluídas aqui:

```
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
!

```

Verificação

O PE2 aprende o prefixo 172.16.1.0/24 do PE1 VPN sobre o peering do MPBGP entre PE1 e PE2. Isso é mostrado aqui:

```
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
```

Da mesma forma, o PE1 aprende o prefixo 172.16.13.0/24 do PE2 VPN sobre o peering do MPBGP entre PE1 e PE2. Isso é mostrado aqui:

```
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.
```

Assim, CE1 pode acessar 172.16.13.13 na rede CE2 através da instância "aqua" de roteamento e encaminhamento de VPN (VRF), configurada em PE1 usando a pilha de rótulos {17 12308}, como mostrado acima.

Esta saída de ping confirma a conectividade:

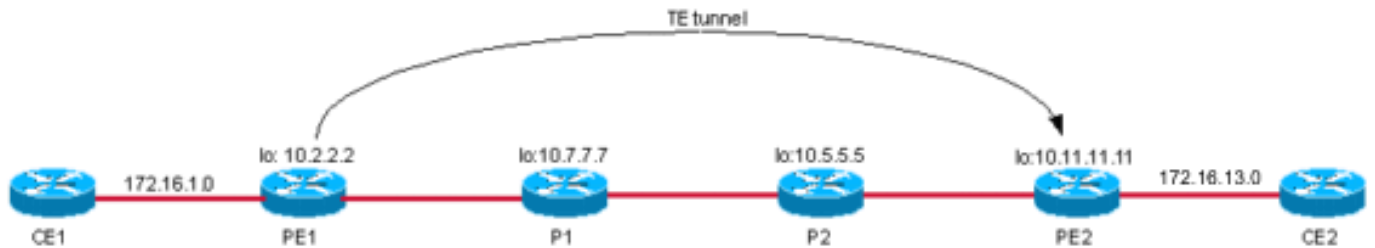
```
CE1# ping 172.16.13.13
```

```
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 um túnel TE quando o túnel TE é de PE1 a](#)

PE2

Topologia



Quando o túnel TE é construído entre os roteadores PE com anúncio de rota automática usado, o salto seguinte do PE de saída BGP pode ser alcançado através da interface de túnel TE. Assim, PE1 usa o rótulo TE para acessar PE2.

Observação: o TE MPLS é independente do LDP, o que significa que, se você tiver uma malha completa de túneis de PE para PE, você poderá efetivamente desativar o LDP nos roteadores e não precisará executar o LDP nas interfaces de túnel TE. No entanto, você deve criar todos os túneis para o próximo salto BGP das rotas VPN versão 4 (VPNv4). No exemplo nesta [configuração](#), você pode ver que esse salto seguinte do BGP é o Loopback0 em PE2, 10.11.11.11. Esse mesmo loopback também é o destino do túnel do PE1 ao PE2. Isso explica por que, neste exemplo, se também há um túnel de PE2 a PE1 para o tráfego de retorno, você pode desativar o LDP no núcleo. Em seguida, o encaminhamento de CE para CE funciona com todo o tráfego VPNv4 transportado pelos túneis TE. Se o salto seguinte do BGP não for o mesmo que o destino do túnel TE, o LDP deve ser executado no núcleo e no túnel TE.

Configuração

A configuração adicional no PE1 para estabelecer um túnel PE é mostrada aqui:

```
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
```

Verificação

```
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

```

Agora, confirme o rótulo externo usado para alcançar o próximo salto 10.11.11.11 por meio do Tunnel0.

```
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

```

Outra maneira de visualizar essas informações rapidamente é usar os modificadores de saída nos comandos **show**, como mostrado aqui:

```

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.

```

Veja a pilha de caracteres especiais. É 19, que é o rótulo do TE, usado para encaminhar pacotes para o Next Hop 10.11.11.0 sobre Tunnel0.

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

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


```

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#

```

Assim, o PE1 envia um pacote destinado a 172.16.13.13 com a pilha de rótulos {19 12308}. P1 troca o rótulo 19. O pacote chega a P2, o que abre o rótulo externo. Em seguida, o pacote é encaminhado para PE2 com apenas o rótulo 12308.

No PE2, o pacote com o rótulo 12308 é recebido e comutado de acordo com as informações na tabela de encaminhamento. Isso é mostrado aqui:

```

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#

```

Observação: nenhuma interface de saída é mostrada porque a tag de saída é Agregada. Isso porque o prefixo associado ao rótulo é a rota diretamente conectada.

Os pings do CE1 para um host no CE2 confirmam a conectividade VPN no túnel TE:

```

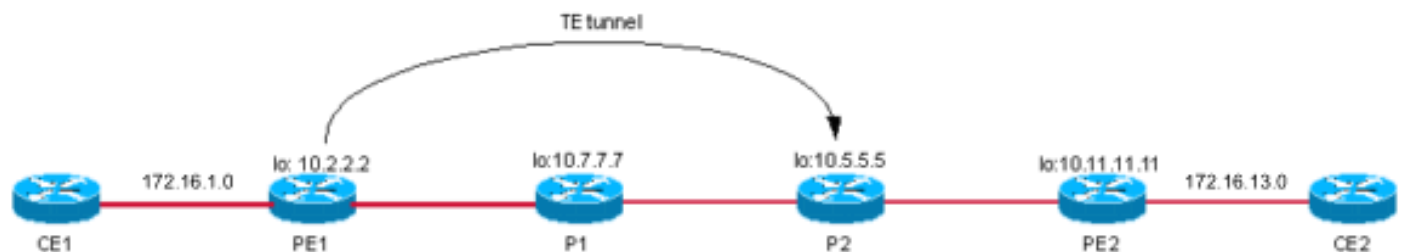
CE1# ping 172.16.13.13

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 um túnel TE quando o túnel TE é de PE1 a P2

Topologia



Configuração

A configuração TE adicional sobre a configuração básica em PE1 é mostrada aqui:

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  
!
```

Verificação

Verifique a rota para o prefixo 172.16.13.13 em PE1 VRF aqua. Ele aponta para o salto seguinte 10.11.11.11/32 (sobre Tunnel0) usando a pilha de rótulos {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#
```

A etiqueta 19, a etiqueta externa, é usada para alcançar o salto seguinte 10.11.11.11/32, como mostrado aqui:

```
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#

```

O pacote de PE1 é enviado pelo túnel TE com a pilha de rótulos {19 12308}. Quando o P1 recebe o pacote, ele abre a tag (PHP) 19 e envia o pacote com a pilha de rótulos {12308}. O comando **show** confirma isso:

```

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>

```

Quando o P2 recebe o pacote com a pilha de rótulos {12308}, ele verifica seu LFIB e descarta o pacote porque não há correspondência. Esta é a saída do comando **show** em 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#
```

Explicação

A solução para esse problema é habilitar TDP/LDP no túnel TE e torná-lo uma interface de caractere comutado. No exemplo mostrado na [Solução](#), o TDP está ativado no Tunnel0 de PE1. P2 está configurado para aceitar saudações direcionadas e formar vizinhos TDP direcionados. Assim, o PE1 recebe o rótulo de 10.11.11.11 de P2 via LDP. Agora que Tunnel0 foi transformado em uma interface comutada por tag e TDP foi ativado para o tráfego para 10.11.11.11, PE1 usa ambos os rótulos; ele usa o rótulo RSVP para acessar o tailend TE e o rótulo TDP para acessar 10.11.11.11.

Neste cenário, PE1 usa a pilha de rótulos {L2 L3 L1} para encaminhar dados para CE2 se esses itens forem verdadeiros:

- L1 é o rótulo da VPN.
- L2 é o rótulo RSVP para acessar o tailend TE.
- L3 é o rótulo TDP para acessar 10.11.11.11 (recebido de P2).

Solução

A solução deve ativar o TDP em um túnel TE.

Configuração

Aqui é exibida a configuração do túnel TE em PE1 com TDP ativado. As adições estão em negrito.

```
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 !
```

Esta é a configuração adicional na extremidade do túnel TE para aceitar saudações TDP direcionadas:

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

Verificação

```
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.
```

Quando P1 recebe o pacote com a pilha de rótulos {19 18 12308}, ele abre a marca 19 e envia o pacote com a pilha de rótulos {18 12308} para P2. O P2 verifica o LFIB em relação ao rótulo 18 e, em seguida, abre a marca e a envia pela interface de saída PO2/0/0 em direção ao PE1. PE1 recebe o pacote com o rótulo 12308 e comuta-o com êxito para 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
```

Um comando ping do CE1 para um host no CE2 confirma a solução.

```
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:
```

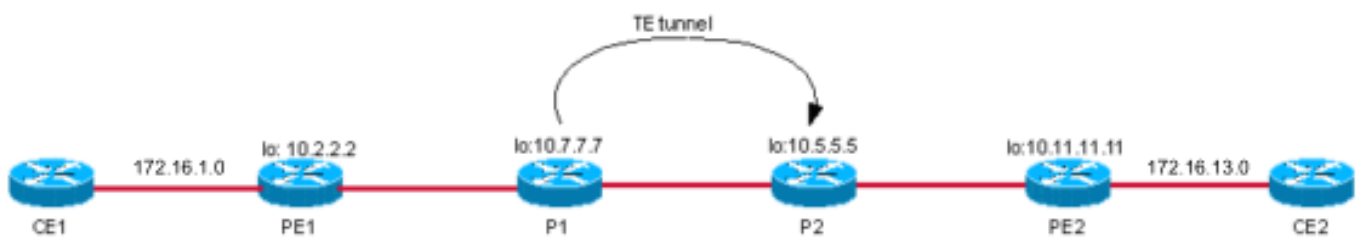
```
!!!!!
```

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

```
CE1#
```

Caso 3: VPN entre CE1 e CE2 sobre um túnel TE de P1 para P2 quando TDP/LDP não está ativado

Topologia



Configuração

A configuração do túnel em PE1 é mostrada aqui:

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
```

Verificação

Verifique como os pacotes destinados ao CE2 172.16.13.13 são comutados aqui. A saída do comando **show ip cef** mostra que os pacotes para o destino 172.16.13.13 são comutados com a pilha de rótulos {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}
```

Quando o P1 recebe esse pacote, ele remove o rótulo externo 17 e comuta o pacote depois de consultar a tabela de roteamento IP para Tunnel0. Observe o `implicit-null OutLabel` nesta saída; significa que a interface de saída não é comutada por rótulo.

```
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
```

Depois que o P2 recebe o pacote com o rótulo 12308, ele olha em sua tabela de encaminhamento. Como não há como o P2 estar ciente da marca VPN 12308 do CE2, ele descarta o pacote.

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

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

```

Isso quebra o caminho dos pacotes VPN destinados ao CE2. Ele é confirmado pelo ping para CE2 172.16.13.13/32.

```
PE1#
CE1# ping 172.16.13.13
```

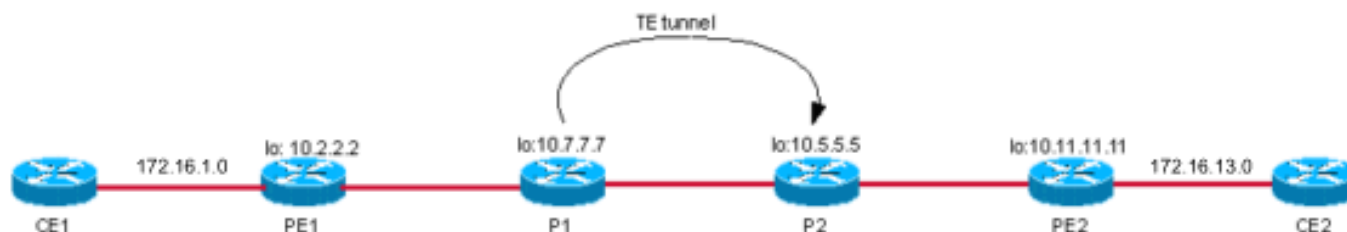
```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.13, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
CE1#
```

Solução

A solução é ativar o LDP/TDP no túnel. A próxima seção aborda essa solução.

Caso 4: VPN sobre um túnel TE entre P1 e P2 com LDP ativado

Topologia



Configuração

Com o LDP ativado no túnel, as configurações em P1 aparecem como mostrado aqui. As adições estão em negrito.

```
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
!
```

Verificação

PE1 envia pacotes para o prefixo 172.16.13.13/32 com a pilha de rótulos {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}
```

O P1 recebe o pacote com a pilha de rótulos {17 12308} e examina seu LFIB para o rótulo 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}

```

Mostra que o rótulo 17 deve ser trocado para o rótulo 18. Portanto, esse pacote é comutado pela interface de túnel com a pilha de rótulos {18 12308}.

P2 recebe o pacote sobre sua interface de túnel com a pilha de rótulos {18 12308}. Ele é exibido com a marca 18 (porque é o penúltimo roteador de salto) e comuta o pacote para o PE2 com o rótulo 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#
```

O PE2 recebe o pacote com o rótulo 12308, que comuta o pacote para CE2 com êxito.

```
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: VPN MPLS sobre um túnel entre P1 e PE2

Topologia



Configuração

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
```

Verificação

O PE1 envia um pacote destinado a 172.16.13.13 ao salto seguinte 10.11.11.11 com a pilha de rótulos {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 recebe o pacote com a pilha de rótulos {17 12308}. P1 examina sua tabela LFIB e verifica a pilha de tags {17} e comuta o pacote com rótulo {17} em direção a 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 recebe o pacote com a pilha de rótulos {17 12308}. P2, sendo o roteador de penúltimo salto, exibe o rótulo 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#
O PE2 recebe o pacote com o rótulo 12308. P2 está ciente de que o destino do rótulo 12308 está diretamente conectado. Portanto, o ping de CE1 para CE2 é 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#

Observação: nenhuma interface de saída é mostrada porque a tag de saída é Agregada. Isso porque o prefixo associado ao rótulo é a rota diretamente conectada.

```

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 conhecidos](#)

Consulte [Field Notice: VPN MPLS com consultivo TE e MPLS InterAS no software Cisco IOS®](#) para obter mais detalhes.

[Conclusão](#)

Quando o túnel do TE estiver terminado no PE de saída, a VPN MPLS e o TE funcionam juntos sem qualquer tipo de configuração adicional. Quando o túnel TE é terminado em qualquer

roteador P (antes do PE no núcleo), o encaminhamento de tráfego VPN MPLS falha porque os pacotes chegam com rótulos de VPN como rótulos de roteador, que não estão nos LFIBs desses dispositivos. Dessa forma, esses roteadores imediatos não são capazes de encaminhar pacotes para o destino final: a rede de cliente VPN. Nesse caso, o LDP/TDP deve ser ativado no túnel TE para resolver o problema.

[Informações Relacionadas](#)

- [MPLS FAQ For Beginners](#)
- [Como solucionar problemas de MPLS VPN](#)
- [Exemplo de engenharia básica de tráfego de MPLS usando configuração de OSPF](#)
- [Configurando uma VPN MPLS básica](#)
- [Troubleshooting de Falha de LSP em MPLS VPN](#)
- [MPLS Support Page](#)
- [Suporte Técnico - Cisco Systems](#)