

Executando um MPLS VPN sobre túneis TE

Índice

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Convenções](#)

[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 for do PE1 ao PE2](#)

[Topologia](#)

[Configuração](#)

[Verificação](#)

[Caso 2: VPN sobre um túnel TE quando o túnel TE for do PE1 ao P2](#)

[Topologia](#)

[Configuração](#)

[Verificação](#)

[Explicação](#)

[Solução](#)

[Caso 3: VPN entre o CE1 e o CE2 sobre um túnel TE do P1 ao P2 quando o TDP/LDP não for permitido](#)

[Topologia](#)

[Configuração](#)

[Verificação](#)

[Solução](#)

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

[Topologia](#)

[Configuração](#)

[Verificação](#)

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

[Topologia](#)

[Configuração](#)

[Verificação](#)

[Problemas conhecidos](#)

[Conclusão](#)

[Informações Relacionadas](#)

Introdução

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. A fim de ganhar os benefícios de um MPLS VPN sobre túneis TE, ambos devem coexistir na rede. Este documento ilustra as várias encenações que explicam porque o encaminhamento de pacote dentro de um MPLS VPN sobre túneis TE pôde falhar. Também fornece uma possível solução.

Pré-requisitos

Requisitos

Os leitores deste documento devem estar cientes destes tópicos:

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

Componentes Utilizados

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

Convenções

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

Material de Suporte



Segundo as indicações desta topologia, em uma configuração simples do MPLS VPN, a ponta de provedor 1 (PE1) aprende a etiqueta VPN (etiqueta 1 [L1]) para o prefixo 172.16.13.0/24 VPN através do protocolo de la puerta de enlace marginal (BGP) multiprotocol (MPBGP) do PE2 diretamente, com o salto seguinte como o endereço de loopback PE2. O PE1 igualmente aprende a etiqueta (L2) para o endereço de loopback PE2 através do protocolo de distribuição de rótulo (LDP) de seu salto seguinte P1.

Quando os dados de encaminhamento ao VPN prefixam 172.16.13.13, o PE1 usa uma pilha de rótulo {L2 L1} com o L2 como a etiqueta exterior. O L2 obtém trocado pelo Label Switch Router do trânsito (LSR), P1. O P2 estala o L2 exterior e para a frente o pacote ao PE2 com o somente um L1. Para compreender melhor porque o P2 estala o L2, refira a seção 3.16 sobre o Penultimate Hop Popping (PHP) no [RFC 3031](#). [Assim, os pacotes ao prefixo 172.16.13.0/24 da versão IP 4 VPN \(IPv4\) são etiqueta comutada sobre uma rede MPLS.](#)

A operação de encaminhamento do MPLS VPN falha se qualquer roteador P recebe o pacote

com L1 (etiqueta VPN) como a única etiqueta exterior em vez da pilha de rótulo {L2 L1}. Isto ocorre porque nenhum dos Roteadores P tem o L1 em sua base de informação de encaminhamento de rótulo (LFIB) para comutar o pacote.

Um MPLS TE usa o Resource Reservation Protocol (RSVP) para trocar etiquetas. Quando um roteador é configurado com TE e Protocolo TDP/LDP, o roteador recebe rótulos diferentes em LDP e RSVP para um determinado prefixo. As etiquetas do LDP e do RSVP não precisam de ser as mesmas em todas as situações. O roteador instala uma etiqueta LDP na tabela do forwarding se o prefixo é instruído através de uma relação LDP, e instala a etiqueta RSVP na tabela do forwarding se o prefixo é instruído sobre uma relação do túnel TE.

No caso de um túnel TE liso (sem LDP/TDP permitido no túnel), o ingresso LSR (o LSR no final do cabeçalho do túnel TE) usa a mesma etiqueta que é usado alcançando a extremidade final do túnel TE para todas as rotas que são instruídas 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 usa então o L3 para alcançar o destino 10.11.11.11/32, aprendido sobre o túnel TE.

Na encenação [acima](#), quando há um túnel TE entre o PE1 e o P2, considere esses para a frente dados PE1 ao edge de cliente 2 (CE2). Se o L4 é a etiqueta VPN, PE1 para a frente os dados com a pilha de rótulo {L3 L4}. O P1 estala o L3, e o P2 recebe o pacote com L4. O PE2 é o único LSR que pode encaminhar corretamente o pacote com L4 de rótulo externo. O P2 não tem uma sessão de MPBGP com PE2, assim que não recebe o L4 do PE2. Consequentemente, o P2 não tem nenhum conhecimento do L2, e deixa cair o pacote.

As configurações e as saídas da **mostra** que seguem demonstram esta e ilustram uma solução possível a este problema.

Configuração VPN inicial entre CE1 e CE2 sem um túnel TE

Topologia



Configuração

Somente 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 47.1234.2222.2222.2222.00
 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 IPv4 PE1 VPN sobre o MPBGP que espreita entre o PE1 e o PE2. Isto 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

```

Similarmente, o PE1 aprende o prefixo 172.16.13.0/24 do IPv4 PE2 VPN sobre o MPBGP que espreita entre o PE1 e o PE2. Isto 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.
```

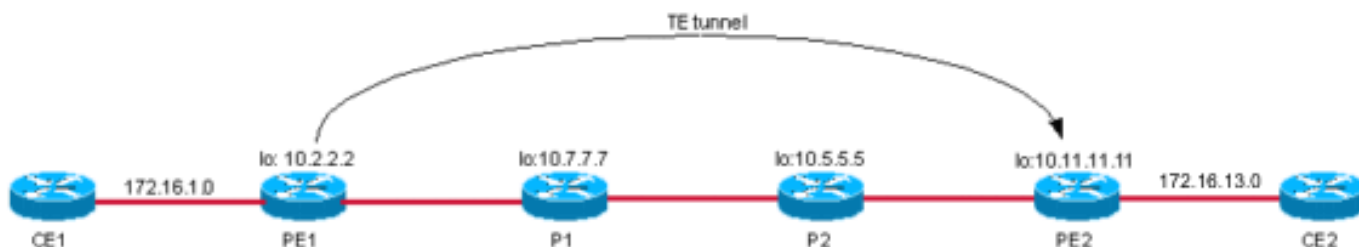
Assim, o CE1 pode alcançar 172.16.13.13 na rede CE2 através do exemplo “aqua” do VPN Routing and Forwarding (VRF), que é configurado no PE1 usando a pilha de rótulo {17 12308}, como mostrado acima.

Esta saída do **sibilo** 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 for do PE1 ao PE2

Topologia



Quando o túnel TE é construído entre os roteadores de PE com autoroute announce usado, o salto seguinte da saída PE BGP é alcançável através da relação do túnel TE. Assim, o PE1 usa a etiqueta TE para alcançar o PE2.

Nota: O MPLS TE é independente do LDP, assim que significa que, se você tem uma malha cheia dos túneis do PE ao PE, você pode eficazmente desabilitar o LDP no Roteadores e não precisa de executar o LDP nas relações do túnel TE. Contudo, você deve construir todos os túneis ao salto seguinte BGP das rotas da versão de VPN 4 (VPNv4). No exemplo nesta [configuração](#), você pode ver que este salto seguinte BGP é Loopback0 no PE2, 10.11.11.11. Este mesmo laço de retorno é igualmente o destino de túnel para o túnel do PE1 ao PE2. Isto explica porque, neste exemplo, se há igualmente um túnel do PE2 ao PE1 para o tráfego de retorno, você pode desabilitar o LDP no núcleo. Então, enviar do CE ao CE trabalha com todo o tráfego do VPNv4 levado sobre os túneis TE. Se o salto seguinte BGP não é 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
```

Uma outra maneira de ver esta informação é rapidamente 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 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#
```

Assim, o PE1 envia um pacote destinado a 172.16.13.13 com a pilha de rótulo {19 12308}. O P1 troca a etiqueta 19. O pacote alcança o P2, que estala que exterior etiqüete. Então, o pacote é enviado ao PE2 com somente a etiqueta 12308.

No PE2, o pacote com a etiqueta 12308 é recebido e comutado de acordo com a informação na tabela do forwarding. Isto 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#
```

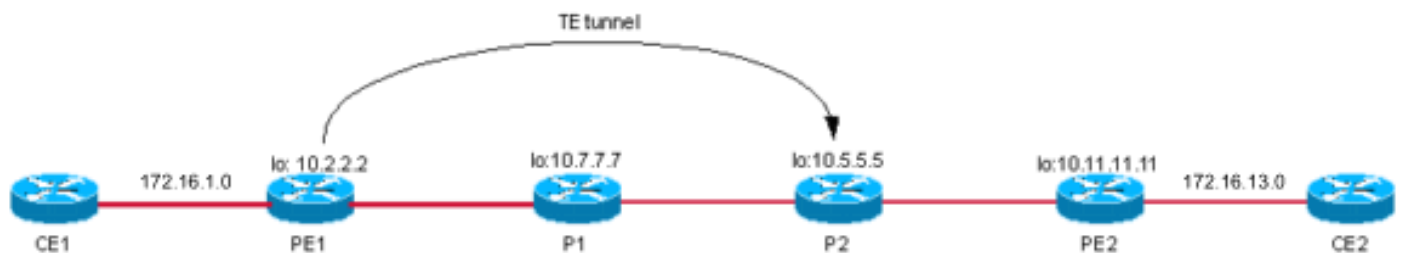
Nota: Nenhuma interface enviada é mostrada porque a etiqueta que parte é agregado. Isto é porque o prefixo associado com a etiqueta é diretamente a rota conectada.

Os sibilos do CE1 a um host no CE2 confirmam a conectividade de VPN sobre o 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 for do PE1 ao P2

Topologia



Configuração

A configuração adicional TE sobre a configuração básica no 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 prefixar 172.16.13.13 nos VRF aquum PE1. Aponta ao salto seguinte 10.11.11.11/32 (sobre o tunnel0) que usa a pilha de rótulo {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 exterior, 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 do PE1 é enviado sobre o túnel TE com a pilha de rótulo {19 12308}. Uma vez que o P1 recebe o pacote, estala (PHP) a etiqueta 19 e envia o pacote com pilha de rótulo {12308}. O comando **show** confirma este:

```
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 rótulo {12308}, verifica seu LFIB e deixa cair o pacote porque nenhum fósforo existe. Este é o **show command output (resultado do comando show)** no 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 é permitido no tunnel0 do PE1. O P2 é configurado aceitando hellos dirigidos e formando vizinhos de TDP dirigidos. Assim, o PE1 recebe a etiqueta para 10.11.11.11 do P2 através do LDP. Agora que o tunnel0 esteve feito a uma relação etiqueta-comutada e o TDP esteve permitido para o tráfego a 10.11.11.11, o PE1 usa ambas as etiquetas; usa a etiqueta RSVP para alcançar o TE tailend e a etiqueta TDP para alcançar 10.11.11.11.

Nesta encenação, o PE1 usa a pilha de rótulo {L2 L3 L1} para enviar dados ao CE2 se estes artigos são verdadeiros:

- O L1 é a etiqueta VPN.
- O L2 é a etiqueta RSVP para alcançar o TE tailend.
- O L3 é a etiqueta TDP para alcançar 10.11.11.11 (recebido do P2).

Solução

A solução deve ativar o TDP em um túnel TE.

Configuração

É mostrada aqui a configuração do túnel TE no PE1 com o TDP ativado nela. As adições estão no

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 final do túnel TE para aceitar hellos dirigidos TDP:

```
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 o P1 recebe o pacote com a pilha de rótulo {19 18 12308}, estala a etiqueta 19 e envia o pacote com a pilha de rótulo {18 12308} ao P2. O P2 verifica seu LFIB para ver se há a etiqueta 18, a seguir estala a etiqueta e envia-a sobre a interface enviada PO2/0/0 para o PE1. PE1 recebe o pacote com o rótulo 12308 e comuta-o com êxito para CE2.

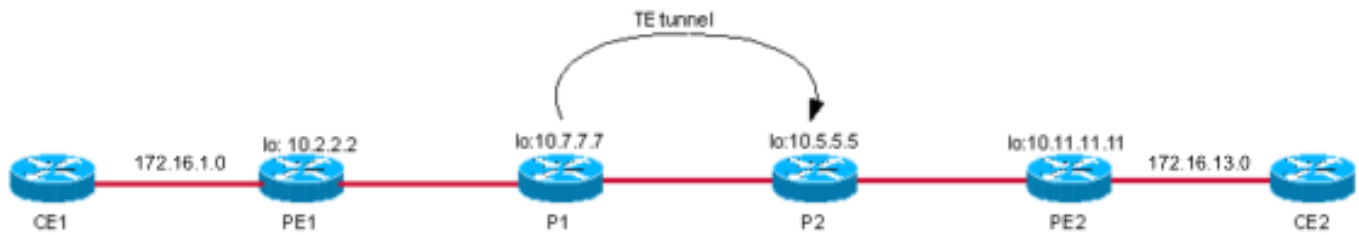
```
P2# show tag for tag 18 Local Outgoing Prefix Bytes tag Outgoing Next Hop tag tag or VC or
Tunnel Id switched interface 18 Pop tag 10.11.11.11/32 117496 POS2/0/0 point2point P2# show tag
tdp discovery Local TDP Identifier: 10.5.5.5:0 Discovery Sources: Interfaces: Ethernet0/3 (tdp):
xmit/rcv TDP 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 a 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 o CE1 e o CE2 sobre um túnel TE do P1 ao P2 quando o TDP/LDP não for permitido

Topologia



Configuração

A configuração de túnel no 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 a CE2 172.16.13.13 obtêm comutados aqui. A saída do comando `show ip cef` mostra que os pacotes ao destino 172.16.13.13 estão comutados com a pilha de rótulo {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 este pacote, remove a etiqueta exterior 17 e comuta o pacote após a vista na tabela de IP Routing ao tunnel0. Observe o OutLabel implícito-nulo nesta saída; significa que a interface enviada não é etiqueta comutada.

```
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. Porque há nenhuma maneira que o P2 pode estar ciente da etiqueta 12308 VPN do CE2, ele deixa cair 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
```

Isto quebra o trajeto dos pacotes de VPN destinados ao CE2. É confirmado pelo **sibilo a CE2 172.16.13.13/32**.

PE1#

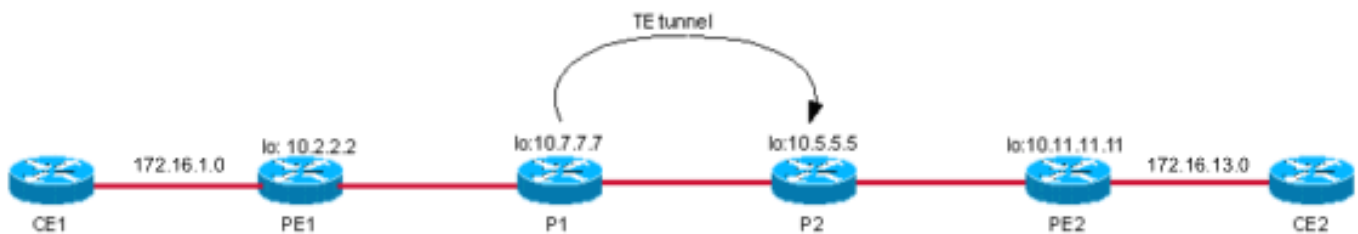
```
CE1# ping 172.16.13.13 Type escape sequence to abort. Sending 5, 100-byte ICMP Echos to 172.16.13.13, timeout is 2 seconds: ..... Success rate is 0 percent (0/5) CE1#
```

Solução

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

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

Topologia



Configuração

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

O PE1 envia pacotes para prefixar 172.16.13.13/32 com a pilha de rótulo {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 pilha de rótulo {17 12308} e olha seu LFIB para a 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}
```

Mostra que a etiqueta 17 deve ser trocada para etiquetar 18. Consequentemente, esse pacote é comutado sobre a interface de túnel com a pilha de rótulo {18 12308}.

O P2 recebe o pacote sobre sua interface de túnel com a pilha de rótulo {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.

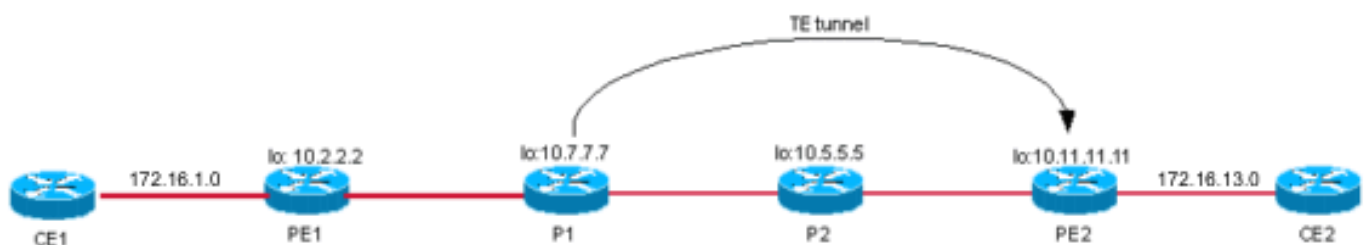
```
P2# show tag for tag 18 detail Local Outgoing Prefix Bytes tag Outgoing Next Hop tag tag or VC
or Tunnel Id switched interface 18 Pop tag 10.11.11.11/32 127645 PO2/0/0 point2point
MAC/Encaps=4/4, MTU=4474, Tag Stack{} 0F008847 No output feature configured Per-packet load-
sharing, slots: 0 1 2 3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

O PE2 recebe o pacote com etiqueta 12308, que comuta o pacote ao CE2 com sucesso.

```
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 um túnel entre o P1 e o 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 a seu salto seguinte 10.11.11.11 com a pilha de rótulo {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}
```

O P1 recebe o pacote com pilha de rótulo {17 12308}. O P1 olha sua tabela LFIB e verifica a pilha {17} da etiqueta e comuta o pacote com etiqueta {17} para o 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}
```

O P2 recebe o pacote com pilha de rótulo {17 12308}. P2, sendo o roteador de penúltimo salto, exhibe 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 P02/0/0 point2point
MAC/Encaps=4/4, MTU=4474, Tag Stack{} 0F008847 No output feature configured P2#
```

O PE2 recebe então o pacote com a etiqueta 12308. O P2 está ciente que o destino para a etiqueta 12308 está conectado diretamente. Consequentemente, o sibilo do CE1 ao 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#
```

Nota: Nenhuma interface enviada é mostrada porque a etiqueta que parte é agregado. Isto é porque o prefixo associado com a etiqueta é diretamente a rota 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: MPLS VPN com Advisory TE e MPLS InterAS no software de Cisco IOS®](#) para 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 estiver terminado em todo o Roteadores P (antes do PE no núcleo), o encaminhamento de tráfego do MPLS VPN falha porque os pacotes chegam com etiquetas VPN como as etiquetas exteriores, que não estão nos LFIB destes dispositivos. Dessa forma, esses roteadores imediatos não são capazes de encaminhar pacotes para o destino final: a rede de cliente VPN. Em tal caso, o LDP/TDP deve ser permitido no túnel TE de resolver o problema.

[Informações Relacionadas](#)

- [Perguntas mais freqüentes sobre MPLS para iniciantes](#)
- [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](#)
- [Página de suporte de MPLS](#)
- [Suporte Técnico - Cisco Systems](#)