

TE 터널을 통한 MPLS VPN 구현

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[표기 규칙](#)

[배경 이론](#)

[CE1과 CE2 간의 초기 VPN 설정\(TE 터널 없음\)](#)

[토폴로지](#)

[구성](#)

[확인](#)

[사례 1: PE1에서 PE2로 TE 터널의 VPN](#)

[토폴로지](#)

[구성](#)

[확인](#)

[사례 2: TE 터널이 PE1에서 P2로 설정된 경우 TE 터널을 통한 VPN](#)

[토폴로지](#)

[구성](#)

[확인](#)

[설명](#)

[솔루션](#)

[사례 3: TDP/LDP가 활성화되지 않은 경우 TE 터널을 통한 CE1과 CE2 간 VPN\(P1에서 P2로\)](#)

[토폴로지](#)

[구성](#)

[확인](#)

[솔루션](#)

[사례 4: LDP가 활성화된 P1과 P2 사이의 TE 터널을 통한 VPN](#)

[토폴로지](#)

[구성](#)

[확인](#)

[사례 5: P1과 PE2 사이의 터널을 통한 MPLS VPN](#)

[토폴로지](#)

[구성](#)

[확인](#)

[알려진 문제](#)

[결론](#)

[관련 정보](#)

소개

이 문서에서는 MPLS 네트워크에서 TE(Multiprotocol Label Switching) 터널을 구현하기 위한 샘플 컨피그레이션을 제공합니다. TE 터널을 통해 MPLS VPN의 이점을 얻으려면 두 가지가 네트워크에 공존해야 합니다. 이 문서에서는 MPLS VPN over TE 터널 내의 패킷 전달이 실패할 수 있는 이유를 설명하는 다양한 시나리오를 설명합니다. 가능한 솔루션도 제공합니다.

사전 요구 사항

요구 사항

이 문서의 독자는 다음 주제에 대해 알고 있어야 합니다.

- [MPLS 트래픽 엔지니어링 및 개선 사항](#)
- [기본 MPLS VPN 구성](#)

사용되는 구성 요소

이 문서는 특정 소프트웨어 및 하드웨어 버전으로 한정되지 않습니다.

표기 규칙

문서 규칙에 대한 자세한 내용은 [Cisco 기술 팁 표기 규칙을 참고하십시오](#).

배경 이론



이 토폴로지에 나와 있는 것처럼 간단한 MPLS VPN 컨피그레이션에서 PE1(Provider Edge 1)은 PE2에서 직접 MPBGP(Multiprotocol Border Gateway Protocol)를 통해 VPN 접두사 172.16.13.0/24에 대한 VPN 레이블(Label 1[L1])을 학습하고 다음 홉은 PE2 루프백 주소로 사용됩니다. PE1은 다음 홉의 P1에서 LDP(Label Distribution Protocol)를 통해 PE2 루프백 주소의 레이블(L2)을 학습합니다.

VPN 접두사 172.16.13.13에 데이터를 전달할 때 PE1은 L2가 있는 레이블 스택 {L2 L1}을 외부 레이블로 사용합니다. L2는 LSR(Transit Label Switch Router), P1, P2가 외부 L2를 팝업하고 하나의 L1만 사용하여 패킷을 PE2에 전달합니다. P2가 L2를 팝업하는 이유를 더 잘 이해하려면 [RFC 3031](#)에서 PHP(Penultimate hop) 포팅(Php)에 대한 섹션 3.16을 참조하십시오. 따라서 VPN IP 버전 4(IPv4) 접두사 172.16.13.0/24에 대한 패킷은 MPLS 네트워크를 통해 레이블로 전환됩니다.

P 라우터가 {L2 L1} 레이블 스택 대신 L1(VPN 레이블)을 유일한 외부 레이블로 받은 패킷을 받는 경우 MPLS VPN 전달 작업이 실패합니다. 이는 LFIB(Label Forwarding Information Base)에 패킷을 전환하기 위한 L1이 없는 IP 라우터이기 때문입니다.

MPLS TE는 RSVP(Resource Reservation Protocol)를 사용하여 레이블을 교환합니다. 라우터가

TE 및 TDP(Tag Distribution Protocol)/LDP에 대해 구성된 경우, 라우터는 지정된 접두사에 대해 LDP 및 RSVP에서 서로 다른 레이블을 받습니다. LDP와 RSVP의 레이블은 모든 상황에서 동일할 필요는 없습니다. LDP 인터페이스를 통해 접두사를 학습하는 경우 라우터는 전달 테이블에 LDP 레이블을 설치하고, 접두사가 TE 터널 인터페이스를 통해 학습되는 경우 전달 테이블에 RSVP 레이블을 설치합니다.

일반 TE 터널(터널에서 LDP/TDP가 활성화되지 않은 경우)의 경우, 인그레스 LSR(TE 터널의 헤드 엔드에 있는 LSR)은 TE 터널을 통해 학습된 모든 경로의 TE 터널에 도달하는 데 사용되는 것과 동일한 레이블을 사용합니다.

예를 들어, 터널을 통해 접두사 10.11.11.11/32을 학습하는 PE1에서 P2까지의 TE 터널이 있습니다. P2의 터널 끝은 10.5.5.5이고 PE1에서 10.5.5.5에 도달할 레이블은 L3입니다. 그런 다음 PE1은 L3을 사용하여 TE 터널을 통해 학습된 목적지 10.11.11.11/32에 연결합니다.

위 시나리오에서 PE1과 P2 간에 TE 터널이 있을 경우 PE1이 데이터를 Customer Edge 2(CE2)로 전달하는 것을 고려하십시오. L4가 VPN 레이블인 경우 PE1은 레이블 스택이 {L3 L4}인 데이터를 전달합니다. P1은 L3을 팝업하고 P2는 L4로 패킷을 수신합니다. PE2는 외부 레이블 L4로 패킷을 올바르게 전달할 수 있는 유일한 LSR입니다. P2는 PE2로 MPBGP 세션이 없으므로 PE2에서 L4를 수신하지 않습니다. 따라서 P2는 L2에 대한 지식이 없으므로 패킷을 삭제합니다.

다음 구성 및 show 출력은 이를 보여주고 이 문제에 대한 가능한 한 가지 해결책을 보여줍니다.

CE1과 CE2 간의 초기 VPN 설정(TE 터널 없음)

토폴로지



구성

구성 파일의 관련 부분만 여기에 포함됩니다.

```

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 47.1234.1010.1010.00
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
!

```

확인

PE2는 PE1과 PE2 간의 MPBGP 피어링을 통해 PE1 VPN IPv4 접두사 172.16.1.0/24을 학습합니다. 다음은 다음과 같습니다.

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

```

마찬가지로, PE1은 PE1과 PE2 간의 MPBGP 피어링을 통해 PE2 VPN IPv4 접두사 172.16.13.0/24을 학습합니다. 다음은 다음과 같습니다.

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.
```

따라서 CE1은 위에 표시된 것처럼 PE1에 레이블 스택 {17 12308}을 사용하여 구성된 VPN 라우팅 및 포워딩(VRF) 인스턴스 "aqua"를 통해 CE2 네트워크에서 172.16.13.13에 연결할 수 있습니다.

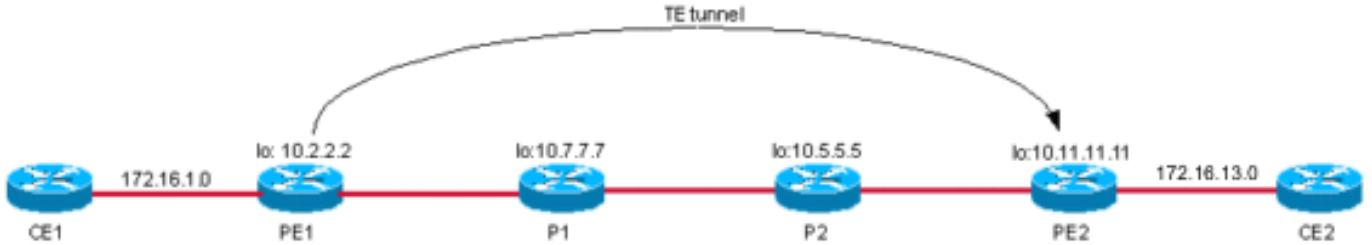
이 ping 출력은 연결을 확인합니다.

```
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
```

사례 1: PE1에서 PE2로 TE 터널의 VPN

토폴로지



PE 라우터 간에 TE 터널이 구축되어 자동 알림이 사용되는 경우 TE 터널 인터페이스를 통해 이그레스 PE BGP next hop에 연결할 수 있습니다. 따라서 PE1은 TE 레이블을 사용하여 PE2에 연결합니다.

참고: MPLS TE는 LDP와 독립적입니다. 즉, PE에서 PE로의 터널이 완전히 메쉬되어 있는 경우 라우터에서 LDP를 효과적으로 비활성화할 수 있으며 TE 터널 인터페이스에서 LDP를 실행할 필요가 없습니다. 그러나 VPNv4(VPN version 4) 경로의 BGP next hop에 대한 모든 터널을 구축해야 합니다. 이 구성의 예에서 이 BGP next hop이 PE2, 10.11.11.11의 Loopback0임을 확인할 수 있습니다. 이 루프백은 PE1에서 PE2로의 터널에 대한 터널 대상이기도 합니다. 이 예에서는 반환 트래픽에 대해 PE2에서 PE1까지의 터널이 있는 경우 코어에서 LDP를 비활성화할 수 있는 이유를 설명합니다. 그런 다음 CE에서 CE로 전달하면 TE 터널을 통해 전달되는 모든 VPNv4 트래픽에서 작동합니다. BGP next hop이 TE 터널 대상과 동일하지 않으면 LDP가 코어 및 TE 터널에서 실행되어야 합니다.

구성

PE 터널을 설정하기 위한 PE1의 추가 컨피그레이션은 다음과 같습니다.

```

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
  
```

확인

```

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
```

이제 Tunnel0을 통해 다음 홉에 연결하는 데 사용되는 외부 레이블을 확인합니다.

```
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
```

이 정보를 빠르게 볼 수 있는 또 다른 방법은 다음과 같이 **show** 명령의 출력 한정자를 사용하는 것입니다.

```
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.
```

태그 스택을 확인합니다. Tunnel0을 통해 next hop 10.11.11.0으로 패킷을 전달하는 데 사용되는 TE 레이블인 19입니다.

```
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#
따라서 PE1은 레이블 스택이 {19 12308}인 172.16.13.13으로 향하는 패킷을 전송합니다. P1은 레이블 19를 교체합니다. 패킷은 P2에 도달하며, 이 패킷은 외부 레이블을 팝업합니다. 그런 다음 패

킷이 레이블 12308과 함께 PE2로 전달됩니다.

PE2에서 12308이라는 레이블이 있는 패킷은 전달 테이블의 정보에 따라 수신 및 전환됩니다. 다음은 다음과 같습니다.

```
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
```

참고: 발신 태그가 Aggregate이므로 No Outgoing interface가 . 레이블과 연결된 접두사가 직접 연결된 경로이기 때문입니다.

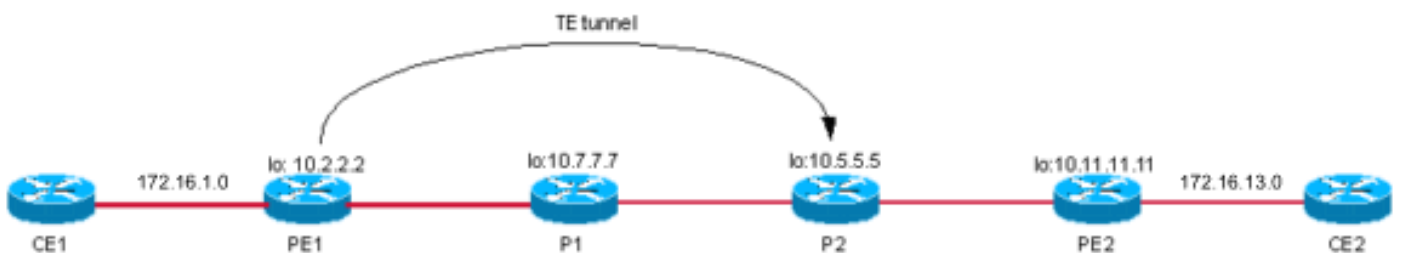
CE1에서 CE2의 호스트로 ping하는 작업은 TE 터널을 통한 VPN 연결을 확인합니다.

```
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#
```

사례 2: TE 터널이 PE1에서 P2로 설정된 경우 TE 터널을 통한 VPN

토폴로지



구성

PE1의 기본 컨피그레이션을 통한 추가 TE 컨피그레이션은 다음과 같습니다.

```
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
!
```

확인

PE1 VRF Aqua의 접두사 172.16.13.13 경로를 확인합니다. 레이블 스택 {19 12308}을 사용하여 next hop 10.11.11.11/32(over Tunnel0)을 가리킵니다.

```
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#

외부 레이블인 Label 19는 다음 홉의 10.11.11.11/32에 연결하는 데 사용됩니다.

```
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#
```

PE1의 패킷이 레이블 스택이 {19 12308}인 TE 터널을 통해 전송됩니다. P1은 패킷을 수신하면 태그 19를 팝업하고 레이블 스택이 {12308}인 패킷을 전송합니다. show 명령은 이를 확인합니다.

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

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

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

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

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

```
P1>
```

P2가 레이블 스택이 {12308}인 패킷을 받으면 LFIB를 확인하고 일치하는 항목이 없으므로 패킷을 삭제합니다. P2의 show 명령 출력입니다.

```
P2# show tag forwarding-table tags 12308 detail
```

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

```
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#
```

설명

이 문제의 해결책은 TE 터널에서 TDP/LDP를 활성화하고 태그 전환 인터페이스로 만드는 것입니다. [솔루션](#)에 표시된 예에서 TDP는 PE1의 Tunnel0에서 활성화됩니다. P2는 직접 헬로를 허용하고 직접 TDP 인접 디바이스를 형성하도록 구성됩니다. 따라서 PE1은 LDP를 통해 P2에서 10.11.11.11에 대한 레이블을 받습니다. 이제 Tunnel0이 태그 전환 인터페이스로 설정되었고 TDP가 10.11.11.11 트래픽에 대해 활성화되었으므로 PE1은 두 레이블을 모두 사용합니다. RSVP 레이블을 사용하여 TE Tailend에 연결하고 TDP 레이블을 사용하여 10.11.11.11에 연결합니다.

이 시나리오에서 PE1은 레이블 스택 {L2 L3 L1}을 사용하여 다음 항목이 true인 경우 데이터를 CE2로 전달합니다.

- L1은 VPN 레이블입니다.
- L2는 TE 맞춤형 엔드에 도달하기 위한 RSVP 레이블입니다.
- L3은 10.11.11.11(P2에서 수신)에 연결할 TDP 레이블입니다.

[솔루션](#)

이 솔루션은 TE 터널을 통해 TDP를 활성화하는 것입니다.

[구성](#)

다음은 TDP가 활성화된 PE1의 TE 터널 컨피그레이션입니다. 추가는 굵은 글씨로 되어 있다.

```
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 !
```

이것은 TE 터널의 끝에 지정 TDP Hello를 수락하기 위한 추가 컨피그레이션입니다.

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

[확인](#)

```
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.
```

P1은 레이블 스택이 {19 18 12308}인 패킷을 받으면 태그 19를 팝업하고 레이블 스택이 {18 12308}인 패킷을 P2로 전송합니다. P2는 레이블 18에 대해 LFIB를 확인한 다음 태그를 팝업하고 PE1에 대한 발신 인터페이스 PO2/0/0을 통해 전송합니다. PE1은 레이블 12308로 패킷을 수신하고 스위치에 성공적으로 전송합니다. CE2.

```
P2# show tag for tag 18
```

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
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/recv
TDP Id: 10.7.7.7:0
POS2/0/0 (tdp): xmit/recv
TDP Id: 10.11.11.11:0
Directed Hellos:
10.5.5.5 -> 10.2.2.2 (tdp): passive, xmit/recv
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

CE1에서 CE2의 호스트로 ping 명령을 실행하여 솔루션을 확인합니다.

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:

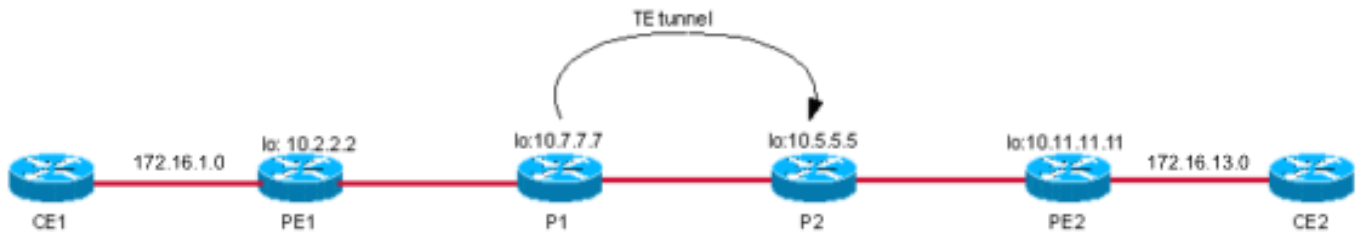
!!!!!

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

CE1#

사례 3: TDP/LDP가 활성화되지 않은 경우 TE 터널을 통한 CE1과 CE2 간 VPN(P1에서 P2로)

토폴로지



구성

PE1의 터널 컨피그레이션은 다음과 같습니다.

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
```

확인

CE2 172.16.13.13으로 향하는 패킷이 여기에서 어떻게 스위칭되는지 확인합니다. show ip cef 명령 출력에서는 대상 172.16.13.13에 대한 패킷이 레이블 스택 {17 12308}과(와) 스위칭되는 것을 보여

줍니다.

```
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}
```

P1은 이 패킷을 수신하면 외부 레이블 17을 제거하고 IP 라우팅 테이블을 검색한 후 패킷을 Tunnel0으로 전환합니다. 이 출력에서 implicit-nullOutLabel을 확인합니다. 이는 발신 인터페이스가 레이블 전환되지 않았음을 의미합니다.

```
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
```

P2가 레이블 12308의 패킷을 수신하면 포워딩 테이블을 살펴봅니다. P2는 CE2에서 VPN 태그 12308을 인식할 수 없으므로 패킷을 삭제합니다.

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

이렇게 하면 CE2로 향하는 VPN 패킷의 경로가 끊어집니다. CE2 172.16.13.13/32으로 ping을 통해 확인할 수 있습니다.

```

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#

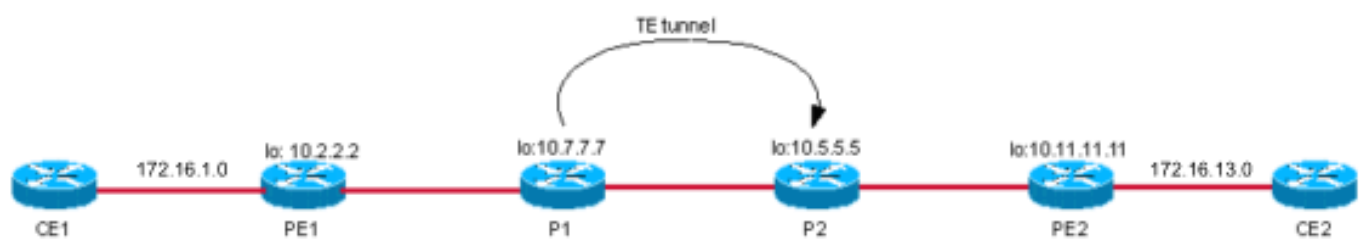
```

솔루션

그 해결책은 터널을 통해 LDP/TDP를 활성화하는 것이다. 다음 섹션에서는 이 솔루션에 대해 설명합니다.

사례 4: LDP가 활성화된 P1과 P2 사이의 TE 터널을 통한 VPN

토폴로지



구성

터널에서 LDP가 활성화되면 P1의 컨피그레이션이 여기에 표시된 것처럼 나타납니다. 추가는 굵은 글씨로 되어 있다.

```

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
!

```

확인

PE1은 레이블 스택이 {17 12308}인 접두사 172.16.13.13/32으로 패킷을 전송합니다.


```

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}

```

P1은 레이블 스택이 {17 12308}인 패킷을 수신하고 LFIB에서 레이블 17을 찾습니다.

```

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}

```

레이블 17을 레이블 18로 교체해야 함을 보여 줍니다. 따라서 해당 패킷은 터널 인터페이스를 통해 레이블 스택 {18 12308}으로 전환됩니다.

P2는 레이블 스택이 {18 12308}인 터널 인터페이스를 통해 패킷을 수신합니다. 태그 18을 팝업하고

(pendultimate hop 라우터이므로) 레이블을 12308로 사용하여 패킷을 PE2로 전환합니다.

```
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#
```

PE2는 레이블이 12308인 패킷을 수신하며, 패킷을 CE2로 성공적으로 전환합니다.

```
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:

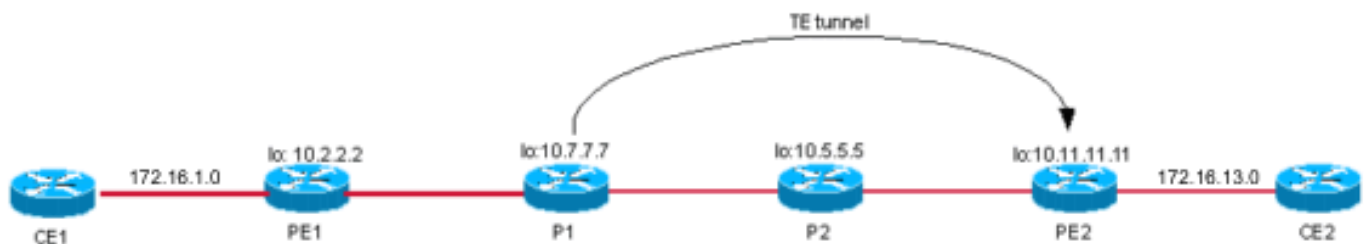
!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

```
CE1#
```

사례 5: P1과 PE2 사이의 터널을 통한 MPLS VPN

토폴로지



구성

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
```

확인

PE1은 레이블 스택 {17 12308}을 사용하여 다음 홉으로 172.16.13.13으로 향하는 패킷을 전송합니다.

```
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}
```

P1은 레이블 스택이 {17 12308}인 패킷을 받습니다. P1은 LFIB 테이블을 살펴보고 태그 스택 {17}을 확인하고 레이블이 {17}인 패킷을 P2로 전환합니다.

```
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}
```

P2는 레이블 스택이 {17 12308}인 패킷을 수신합니다. P2는 penultimate hop 라우터이며 레이블

17을 표시합니다.

```
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#
그런 다음 PE2는 레이블 12308의 패킷을 수신합니다. P2는 레이블 12308의 대상이 직접 연결되어 있음을 인식합니다. 따라서 CE1에서 CE2로의 ping은 10입니다.

```
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#
참고: 발신 태그가 Aggregate이므로 No Outgoing interface가 . 레이블과 연결된 접두사가 직접 연결된 경로이기 때문입니다.

```
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#
```

알려진 문제

[필드 알림](#)을 참조하십시오. [자세한 내용은 MPLS VPN with TE 및 MPLS InterAS Advisory on Cisco IOS® Software](#)를 참조하십시오.

결론

이그레스 PE에서 TE 터널이 종료되면 추가 컨피그레이션 없이 MPLS VPN과 TE가 함께 작동합니다. IP 라우터에서 TE 터널이 종료될 때(코어의 PE가 시작되기 전) 패킷이 VPN 레이블을 외부 레이블로 가지고 도달하기 때문에 MPLS VPN 트래픽 전달이 실패합니다. 이 레이블은 이러한 디바이스의 LFIB에 없습니다. 따라서 이러한 중간 라우터는 최종 목적지인 VPN 고객 네트워크로 패킷을 전달할 수 없습니다. 이러한 경우 TE 터널에서 LDP/TDP를 활성화하여 문제를 해결해야 합니다.

관련 정보

- [초보자용 MPLS FAQ](#)
- [MPLS VPN 문제 해결 방법](#)
- [OSPF 컨피그레이션을 사용하는 MPLS 기본 트래픽 엔지니어링 예](#)

- [기본 MPLS VPN 구성](#)
- [MPLS VPN에서 LSP 장애 문제 해결](#)
- [MPLS 지원 페이지](#)
- [Technical Support - Cisco Systems](#)