

# 实现在流量工程隧道的MPLS VPN

## 目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[规则](#)

[背景理论](#)

[没有 TE 隧道时 CE1 与 CE2 之间 VPN 的初始设置](#)

[拓扑](#)

[配置](#)

[验证](#)

[第 1 种情况：在TE隧道的VPN，当TE隧道是从PE1到PE2](#)

[拓扑](#)

[配置](#)

[验证](#)

[第 2 种情况：在TE隧道的VPN，当TE隧道是从PE1到P2](#)

[拓扑](#)

[配置](#)

[验证](#)

[说明](#)

[解决方案](#)

[实例3：其它WRR加权修改在CE1与CE2之间的VPN在一TE隧道从P1到P2，当TDP/LDP没有启用](#)

[拓扑](#)

[配置](#)

[验证](#)

[解决方案](#)

[实例4：修改队列极限缓冲区分配在一TE隧道的VPN在P1和P2之间与启用的LDP](#)

[拓扑](#)

[配置](#)

[验证](#)

[案例5：在一个通道的MPLS VPN在P1和PE2之间](#)

[拓扑](#)

[配置](#)

[验证](#)

[已知问题](#)

[结论](#)

[相关信息](#)

## 简介

本文为实现在流量工程(TE)通道的一多协议标签交换(MPLS)提供配置示例VPN在MPLS网络。为了得到MPLS VPN的好处在流量工程隧道的，两个在网络应该共存。本文说明解释的多种方案在MPLS VPN内的信息包转发在流量工程隧道为什么也许发生故障。它也提供一个可能的解决方案。

## 先决条件

### 要求

本文档的读者应掌握以下这些主题的相关知识：

- [MPLS流量工程和增强](#)
- [配置基本 MPLS VPN](#)

### 使用的组件

本文档不限于特定的软件和硬件版本。

### 规则

有关文档规则的详细信息，请参阅 [Cisco 技术提示规则](#)。

### 背景理论



如此拓扑所显示，在一简单MPLS VPN配置里，运营商边缘1 (PE1)通过多协议学习VPN标签(标签1 [L1]) VPN前缀的172.16.13.0/24边界网关协议(MPBGP)从直接PE2，与下一跳作为PE2环回地址。PE1通过从其下一跳P1的标签转发协议(LDP)也学习标签(L2) PE2环回地址的。

当对VPN的转发数据加前缀172.16.13.13时，PE1以L2使用标签栈{L2 L1}作为外面标签。L2由传输标签交换路由器(LSR)得到交换，P1。P2只弹出外面L2并且转发数据包对与一个L1的PE2。要改善请知道P2为什么弹出L2，参考关于Penultimate Hop Popping (PHP)的第3.16部分在[RFC 3031](#)。因此，对VPN IP版本4 (IPv4)前缀172.16.13.0/24的数据包是在MPLS网络交换的标签。

MPLS VPN转发操作发生故障，如果任何P路由器收到有L1的(VPN标签)数据包作为唯一的外面标签而不是{L2 L1}标签栈。因为P路由器都没有L1在其转换数据包的标签转发信息基础(LFIB)这发生。

MPLS TE使用资源预留协议(RSVP)交换标签。当路由器为两TE和标签发行协议(TDP) /LDP配置，从LDP和RSVP的路由器接收不同的标签一个给的前缀的。从LDP和RSVP的在所有情况下标签不需要是相同的。路由器在转发表里安装一个LDP标签，如果前缀通过LDP接口了解，并且在转发表里安装RSVP标签，如果前缀在TE隧道接口了解。

一旦一无格式TE隧道(没有在通道启用的LDP/TDP)，入口LSR (在TE隧道的头端的LSR)使用标签和

一样使用到达TE隧道的尾端通过TE隧道了解的所有路由的。

例如，有一TE隧道从PE1到学习在通道的P2前缀10.11.11.11/32。在P2的通道尾端是10.5.5.5，并且到达在PE1的10.5.5.5的标签是L3。PE1然后使用L3到达目的地10.11.11.11/32，了解在TE隧道。

在以上的方案中，当有在PE1和P2之间的一TE隧道，请考虑PE1转递数据到用户边缘2 (CE2)。如果L4是VPN标签，PE1转递与标签栈{L3 L4}的数据。P1弹出L3，并且P2收到有L4的数据包。PE2是能正确地转发数据包用外面标签L4的唯一的LSR。P2没有有PE2的一MPBGP会话，因此不接收从PE2的L4。所以，P2没有L2任何知识，并且丢弃数据包。

配置和显示跟随展示此并且说明一个可能的解决方案对此问题的输出。

## 没有 TE 隧道时 CE1 与 CE2 之间 VPN 的初始设置

### 拓扑



### 配置

配置文件的仅相关部分包括此处：

```
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
!

```

## 验证

PE2了解PE1 VPN在并列在PE1及PE2之间的MPBGP的IPv4前缀172.16.1.0/24。这显示此处：

```
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
```

同样地，PE1了解PE2 VPN在并列在PE1及PE2之间的MPBGP的IPv4前缀172.16.13.0/24。这显示此处：

```
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.

```

因此，CE1能通过VPN路由与转发(VRF)实例“水色”到达在CE2网络的172.16.13.13，使用标签栈{17 12308}，在PE1配置，如上所述。

此ping输出确认连接：

```

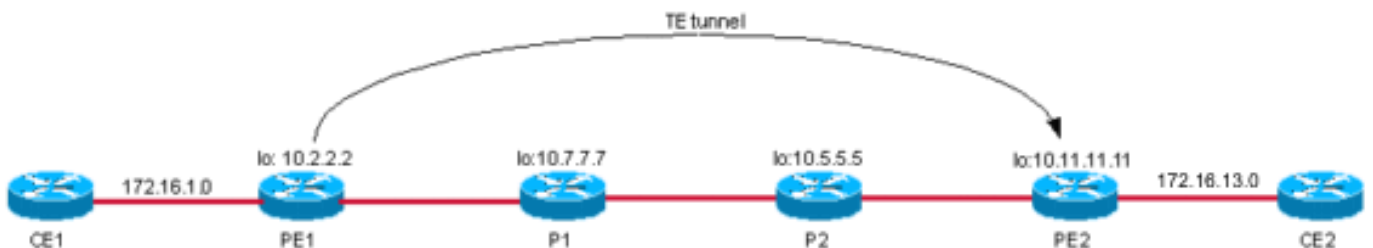
CE1# ping 172.16.13.13

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

```

## 第 1 种情况：在TE隧道的VPN，当TE隧道是从PE1到PE2

### 拓扑



当TE隧道被构件在有使用时的autoroute announce的PE路由器之间，出口PE BGP下一跳通过TE隧道接口是可及的。因此，PE1使用TE标签到达PE2。

**注意：** MPLS TE对立干LDP，因此意味着，如果有通道全网状从PE到PE，您能有效禁用在路由器的LDP，并且不需要运行在TE隧道接口的LDP。然而，您必须构建所有通道到VPN版本4 (Vpvn4)路由的BGP下一跳。在示例中在此[配置方面](#)，您能看到此BGP下一跳是在PE2的Loopback0，

10.11.11.11。此同样环回也是通道的隧道目的地从PE1到PE2。这解释，在本例中，如果也有一个通道从PE2到回程数据流的PE1，您为什么能禁用在核心的LDP。然后，转发从CE到CE与所有Vpnv4流量一起使用把流量工程隧道转入。如果BGP下一跳不是相同的象TE隧道目的地，必须运行LDP在核心和在TE隧道。

## 配置

在设立PE通道的PE1的更多的配置显示此处：

```
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
```

## 验证

```
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
现在，请确认用于的外面标签通过隧道0到达下一跳10.11.11.11。
```

```
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
```

另一个方式查看此信息迅速将使用输出修正值在显示命令，如显示此处：

```
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.
```

查看标记堆叠。它是19，是TE标签，过去常常转发数据包到在隧道0的下一跳10.11.11.0。

```
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#
```

因此，PE1发送被注定的数据包对与标签栈{19 12308}的172.16.13.13。P1交换标签19。数据包到达P2，弹出外面请标记。然后，数据包转发对PE2用仅标签12308。

在PE2，数据包用标签12308根据信息在转发表里接收并且交换。这显示此处：

```
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#
```

注意：因为流出的标记显示。这是因为用标签关联的前缀直接地是已连接路由。

从CE1的Ping到在CE2的一台主机确认在TE隧道的VPN连接：

```
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:
```

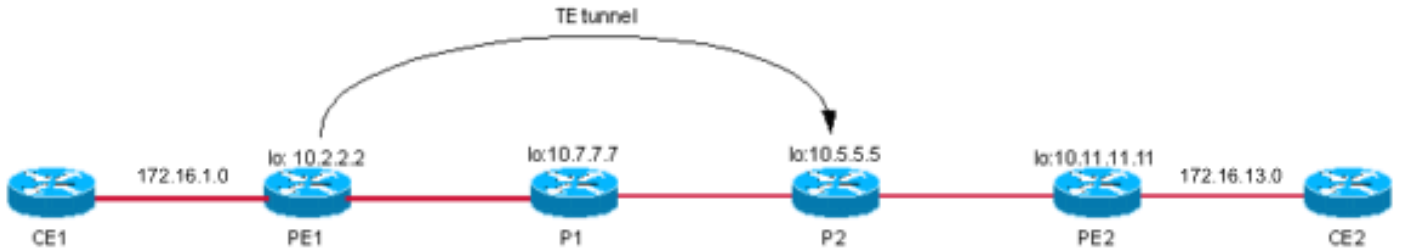
```
!!!!
```



Success rate is 100 percent (5/5), round-trip min/avg/max = 4/13/36 ms  
CE1#

## 第 2 种情况：在TE隧道的VPN，当TE隧道是从PE1到P2

### 拓扑



### 配置

在基本配置的另外的TE配置在PE1显示此处：

```
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
!
```

### 验证

检查路由加前缀在PE1 VRF水色的172.16.13.13。它指向下一跳10.11.11.11/32 (在隧道0)使用标签栈{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#  
标签19，外面标签，用于到达下一跳10.11.11.11/32，如显示此处：

```
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#
```

从PE1的数据包在与标签栈{19 12308}的TE隧道发送。一旦P1收到数据包，弹出(PHP)标记19并且发送有标签栈的{12308}数据包。show命令确认此：

```
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>

当P2收到有标签栈的{12308}时数据包，检查其LFIB并且丢弃数据包，因为匹配不存在。这是在P2的show命令输出：

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

```

P2#

P2#

## 说明

对此问题的解决方案将启用在TE隧道的TDP/LDP和做它一个标记交换的接口。在[解决方案](#)显示的示例中，TDP在PE1隧道0启用。P2为接受处理的hello和形成处理的TDP邻居配置。因此，PE1通过LDP收到10.11.11.11的标签从P2。即然隧道0做一个标记交换的接口，并且TDP为对10.11.11.11的流量启用，PE1使用两个标签;它使用RSVP标签到达TE尾端和TDP标签到达10.11.11.11。

在此方案中，如果这些项目是真的，PE1使用标签栈{L2 L3 L1}转递数据到CE2：

- L1是VPN标签。
- L2是到达TE尾端的RSVP标签。
- L3是到达10.11.11.11的TDP标签(接收从P2)。

## 解决方案

解决方案将启用在TE隧道间的TDP。

## 配置

显示此处PE1的TE隧道配置与对此的开启TDP时。新增内容黑体字。

```

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 !

```

这是在接收处理的TDP hello的TE隧道的尾端的更多的配置：

```

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

```

## 验证

```

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.

```

当P1收到有标签栈的{19 18 12308}时数据包，弹出标记19并且发送有标签栈的{18 12308}数据包对P2。P2检查其LFIB标签18，然后弹出标记并且在流出接口PO2/0/0发送它往PE1。PE1收到数据包用标签12308并且顺利地换成它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/recv

```

```
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

一ping命令从CE1到在CE2的一台主机确认解决方案。

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:

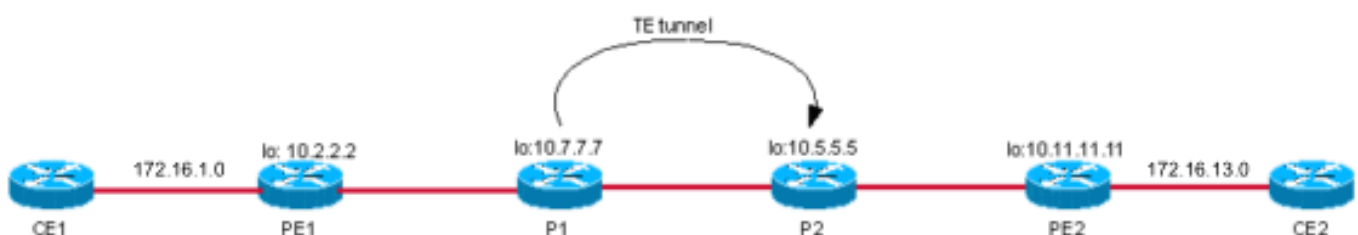
!!!!!

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

CE1#

## 实例3：其它WRR加权修改在CE1与CE2之间的VPN在一TE隧道从P1到P2，当TDP/LDP没有启用

### 拓扑



## 配置

在PE1的隧道配置显示此处：

```
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
```

## 验证

验证数据包被注定对CE2 172.16.13.13如何得到交换此处。**show ip cef**命令输出显示对目的地172.16.13.13的数据包交换与标签栈{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}
```

当P1收到此数据包时，删除外面标签17并且在查找转换数据包在IP路由表以后对隧道0。注意implicit-null OutLabel在此输出中;意味着流出接口不是交换的标签。

```
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

```

一旦P2收到数据包用标签12308，查看其转发表。由于没有办法P2可以知道从CE2的VPN标记12308，丢弃数据包。

```

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

```

这中断VPN数据包路径被注定对CE2。它由对CE2 172.16.13.13/32的ping确认。

```

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#

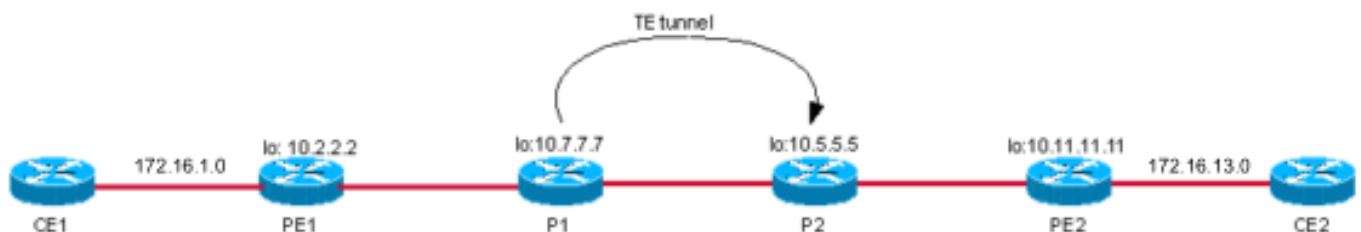
```

## 解决方案

解决方案将启用在通道的LDP/TDP。下一部分讨论此解决方案。

## 实例4：修改队列极限缓冲区分配在一TE隧道的VPN在P1和P2之间与启用的LDP

### 拓扑



### 配置

使用在通道启用的LDP，在P1的配置出现如显示此处。新增内容黑体字。

## PE1

```
PE1# 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
!
```

## 验证

PE1发送数据包加前缀172.16.13.13/32与标签栈{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}
```

P1收到有标签栈的{17 12308}数据包并且查看其LFIB为标签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}

```

它显示应该交换标签17标记18。所以，该数据包在与标签栈{18 12308}的隧道接口交换。

P2收到数据包经过其与标签栈{18 12308}的隧道接口。它弹出标记18 (因为它是倒数第二的跳跃路由器)并且转换数据包对PE2用标签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#

```

PE2收到数据包用标签12308，顺利地转换数据包对CE2。

```

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#

```

## 案例5：在一个通道的MPLS VPN在P1和PE2之间

### 拓扑



## 配置

### 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
```

## 验证

PE1发送数据包被注定为172.16.13.13对其与标签栈{17 12308}的下一跳10.11.11.11。

```
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收到有标签栈的{17 12308}数据包。P1看看其LFIB表和检查标记堆叠{17}并且转换数据包用标签{17}往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收到有标签栈的{17 12308}数据包。P2，是倒数第二的跳跃路由器，pops标签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#

```

PE2然后收到数据包用标签12308。P2知道标签的12308目的地直接地连接。所以，ping从CE1到CE2是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#

```

**注意：** 因为流出的标记显示。这是因为用标签关联的前缀直接地是已连接路由。

```

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#

```

## 已知问题

有关此问题的示例，请参阅 [Field Notice : 与TE和MPLS InterAS建议的MPLS VPN在Cisco IOS软](#)

[件](#)欲了解更详细的信息。

## [结论](#)

当TE隧道在出口PE、MPLS VPN和TE工作一起终止，不用任何更多的配置。当TE隧道在所有P路由器终止(在核心的PE)前，MPLS VPN流量转发发生故障，因为数据包用VPN标签到达作为外面标签，不在这些设备LFIB。所以，这些中间路由器不能转发数据包到最终目的地，VPN客户网络。在这种情况下，在TE隧道应该启用LDP/TDP解决问题。

## [相关信息](#)

- [对初学者的MPLS 常见问题](#)
- [如何排除 MPLS VPN 故障](#)
- [使用 OSPF 的 MPLS 基本流量工程配置示例](#)
- [配置基本 MPLS VPN](#)
- [MPLS VPN 上 LSP 故障排除](#)
- [MPLS 支持页](#)
- [技术支持 - Cisco Systems](#)