

Применение VPN MPLS по туннелям TE

Содержание

[Введение](#)

[Предварительные условия](#)

[Требования](#)

[Используемые компоненты](#)

[Условные обозначения](#)

[Теоретические сведения](#)

[Начальная установка VPN между CE1 и CE2 без туннеля TE](#)

[Топология](#)

[!--- конфигурацию](#)

[Проверка](#)

[Пример 1: VPN по туннелю TE, когда туннель TE от PE1 до PE2](#)

[Топология](#)

[!--- конфигурацию](#)

[Проверка](#)

[Случай 2: VPN по туннелю TE, когда туннель TE от PE1 до P2](#)

[Топология](#)

[!--- конфигурацию](#)

[Проверка](#)

[Пояснение](#)

[Решение](#)

[Случай 3: VPN между CE1 и CE2 по туннелю TE от P1 до P2, когда не включен TDP/LDP](#)

[Топология](#)

[!--- конфигурацию](#)

[Проверка](#)

[Решение](#)

[Пример 4: VPN по туннелю TE между P1 и P2 с включенным LDP](#)

[Топология](#)

[!--- конфигурацию](#)

[Проверка](#)

[Случай 5: MPLS VPN по туннелю между P1 и PE2](#)

[Топология](#)

[!--- конфигурацию](#)

[Проверка](#)

[Типичные ошибки](#)

[Заключение](#)

[Дополнительные сведения](#)

Введение

Этот документ содержит примеры конфигурации для внедрения VPN-сети многопротокольной коммутации по меткам (MPLS) по туннелям с регулированием трафика (TE) в сети MPLS. Для получения выгоды от MPLS VPN по Туннелям TE оба должны сосуществовать в сети. Этот документ иллюстрирует различные сценарии, которые объясняют, почему могла бы отказать пересылка пакетов в MPLS VPN по Туннелям TE. Также приводится возможное решение.

Предварительные условия

Требования

Читатели данного документа должны обладать знаниями по следующим темам:

- [Регулирование трафика MLPS и усовершенствования](#)
- [Выбор базовой конфигурации для MPLS VPN](#)

Используемые компоненты

Настоящий документ не имеет жесткой привязки к каким-либо конкретным версиям программного обеспечения и оборудования.

Условные обозначения

[Дополнительные сведения об условных обозначениях см. в документе Условные обозначения технических терминов Cisco.](#)

Теоретические сведения

Как показано в этой топологии, в простой конфигурации MPLS VPN, Граница провайдера 1 (PE1) узнает, что метка VPN (Метка 1 [L1]) для VPN снабжает префиксом 172.16.13.0/24 через многопротокольный Border Gateway Protocol (MPBGP) от PE2 непосредственно со следующим переходом как адрес обратной связи PE2. PE1 также изучает метку (L2) для адреса обратной связи PE2 через Протокол распределения меток (LDP) от его P1 следующего перехода.

Когда данные переадресации к VPN снабжают префиксом 172.16.13.13, PE1 использует стек меток {L2 L1} с L2 как внешняя метка. L2 заменен транзитным маршрутизатором на основе коммутации пакетов, P1. P2 выталкивает внешний L2 и передает пакет к PE2 только с одним L1. Чтобы лучше понять, почему P2 выталкивает L2, обратитесь для разделения 3.16 о вытеснении предпоследней пересылки (PHP) в [RFC 3031](#). Таким образом пакеты к IP версии 4 VPN 172.16.13.0/24 префикса (IPv4) с коммутацией по меткам по сети MPLS.

Если какой-либо маршрутизатор P получает пакет с L1 (метка VPN) как единственная внешняя метка вместо {L2 L1} стек меток, функционирование переадресации MPLS VPN отказывает. Это происходит, потому что ни один из маршрутизаторов P не имеет L1 в его Обозначении базы данных переадресации (LFIB) для коммутации пакета.

MPLS TE использует Протокол RSVP для обмена метками. Когда маршрутизатор настроен и на TE, и на протокол распределения тегов (TDP)/LDP, он получает различные метки как от LDP, так и от RSVP для данного префикса. Метки от LDP и RSVP не должны быть тем же во всех ситуациях. Маршрутизатор устанавливает метку LDP в таблице пересылки, если префикс изучен через интерфейс LDP, и это устанавливает метку RSVP в таблице пересылки, если префикс изучен по интерфейсу Туннеля TE.

В случае простого Туннеля TE (без LDP/TDP включил на туннеле), входной LSR (LSR на головном узле Туннеля TE) использует как есть ту же метку, используемую для достижения окончания Туннеля TE для всех маршрутов, которые изучены через Туннель TE.

Например, существует туннель TE от PE1 до PE2, и префикс 10.11.11.11/32 получается по туннелю. Окончание туннеля на P2 - 10.5.5.5, а метка для доступа к 10.5.5.5 в PE1 - L3. PE1 тогда использует L3 для достижения целевого 10.11.11.11/32, изученного по Туннелю TE.

В сценарии [выше](#), когда существует Туннель TE между PE1 и P2, полагают, что PE1 передает данные к Порту заказчика Customer Edge 2 (CE2). Если L4 является меткой VPN, PE1 передает данные со стеком меток {L4 L3}. L3 популярности P1 и P2 получают пакет с L4. PE2 – единственный LSR, который может правильно переадресовать пакет с внешней меткой L4. P2 не имеет сеанса MPBGP с PE2, таким образом, это не получает L4 от PE2. Поэтому P2 не имеет никакого знания L2, и это отбрасывает пакет.

Конфигурации и **выходные данные show**, которые придерживаются, демонстрируют это и иллюстрируют одно возможное решение к этой проблеме.

Начальная установка VPN между CE1 и CE2 без туннеля TE

Топология

!--- конфигурацию

Только соответствующие части файлов конфигурации включены здесь:

```
PE 1
hostname PE1
ip cef
!
ip vrf aqua
  rd 100:1
  route-target export 1:1
  route-target import 1:1
!
mpls traffic-eng tunnels
!
interface Loopback0
  ip address 10.2.2.2 255.255.255.255
  no ip directed-broadcast
!
interface Ethernet2/0/1
  ip vrf forwarding aqua
  ip address 172.16.1.2 255.255.255.0
!
interface Ethernet2/0/2
  ip address 10.7.7.2 255.255.255.0
```

```
ip router isis
mpls traffic-eng tunnels
tag-switching ip
!
router isis
passive-interface Loopback0
net 47.1234.2222.2222.00
is-type level-1
metric-style wide
mpls traffic-eng router-id Loopback0
mpls traffic-eng level-1
!
router bgp 1
bgp log-neighbor-changes
neighbor 10.11.11.11 remote-as 1
neighbor 10.11.11.11 update-source Loopback0
!
address-family vpnv4
neighbor 10.11.11.11 activate
neighbor 10.11.11.11 send-community extended
exit-address-family
!
address-family ipv4
neighbor 10.11.11.11 activate
no auto-summary
no synchronization
exit-address-family
!
address-family ipv4 vrf aqua
redistribute connected
no auto-summary
no synchronization
exit-address-family
```

PE 2

```
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
!

```

Проверка

PE2 узнает, что IPv4 VPN PE1 снабжает префиксом 172.16.1.0/24 по пирингу MPBGP между PE1 и PE2. Это показывают здесь:

```

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

```

Точно так же PE1 узнает, что IPv4 VPN PE2 снабжает префиксом 172.16.13.0/24 по пирингу MPBGP между PE1 и PE2. Это показывают здесь:

```

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.

```

Таким образом CE1 может достигнуть 172.16.13.13 в сети CE2 через VPN Routing и Forwarding (VRF) экземпляр "вода", которая настроена на PE1 с помощью стека меток {17

12308}, как показано выше.

Эти **выходные данные ping** подтверждают подключение:

```
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
```

Пример 1: VPN по туннелю TE, когда туннель TE от PE1 до PE2

Топология

Когда Туннель TE создан между Периферийными маршрутизаторами с используемым autoroute announce, выходной следующий переход BGP PE достижим через интерфейс Туннеля TE. Таким образом PE1 использует метку TE для достижения PE2.

Примечание: MPLS TE независим от LDP, что означает, что, если у вас есть полная сетка туннелей от PE до PE, вы можете эффективно отключить LDP в маршрутизаторах и не должны выполнять LDP на интерфейсах Туннеля TE. Однако необходимо создать все туннели к следующему переходу BGP маршрутов версии VPN 4 (VPNv4). В примере в этой [Конфигурации](#) вы видите, что этим следующим переходом BGP является Loopback0 на PE2, 10.11.11.11. Этот тот же loopback является также назначением туннеля для туннеля от PE1 до PE2. Это объясняет, почему, в данном примере, если существует также туннель от PE2 до PE1 для ответного трафика, можно отключить LDP в ядре. Затем передача от CE до CE работает со всем трафиком VPNv4, перенес Туннели TE. Если следующий переход BGP не является тем же как назначением Туннеля TE, LDP должен быть выполнен в ядре и на Туннеле TE.

!--- конфигурацию

Дополнительную настройку на PE1 для установления туннеля PE показывают здесь:

PE 1
<pre>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</pre>

Проверка

```
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
```

Теперь подтвердите внешнюю метку, используемую для достижения следующего перехода

10.11.11.11 с помощью 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
```

Другой способ посмотреть эту информацию быстро состоит в том, чтобы использовать модификаторы вывода в командах показа, как показано здесь:

```
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.
```

Посмотрите на стек тегов. 19, метка TE, используется для перенаправления пакетов до следующего перехода 10.11.11.0 через туннель 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#
```

Таким образом PE1 передает пакет, предназначенный к 172.16.13.13 со стеком меток {19 12308}. P1 подкачивает метку 19. Пакет достигает P2, который выталкивает ту внешнюю метку. Затем пакет передан к PE2 с только меткой 12308.

На PE2 пакет с меткой 12308 получен и коммутирован согласно информации в таблице пересылки. Это показывают здесь:

```
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#
```

Примечание: Никакой не показывают, потому что исходящий тег является . Это вызвано тем, что префикс, привязанный к метке, является маршрутом прямого соединения.

Эхо-запросы от CE1 до хоста на CE2 подтверждают возможность VPN - подключения по Туннелю 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#
```

[Случай 2: VPN по туннелю TE, когда туннель TE от PE1 до P2](#)

[Топология](#)

[!--- конфигурацию](#)

Дополнительную конфигурацию TE по базовой конфигурации на PE1 показывают здесь:

PE 1

```
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 !
```

Проверка

Проверьте маршрут для добавления префикса 172.16.13.13 на VRF aquim PE1. Это указывает к следующему переходу 10.11.11.11/32 (по Tunnel0) использование стека меток {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#
```

Метка 19, внешняя метка, используется для достижения следующего перехода 10.11.11.11/32, как показано здесь:

```
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#
```

Пакет от PE1 передан по Туннелю ТЕ со стеком меток {19 12308}. Как только P1 получает пакет, он выталкивает (PHP) метку 19 и передает пакет со стеком меток {12308}. Команда показа подтверждает это:

```
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>
```

Когда P2 получает пакет со стеком меток {12308}, это проверяет свой LFIB и отбрасывает пакет, потому что не существует никакого соответствия. Это - выходные данные команды show на 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#
```

Пояснение

Решением данной проблемы является включение TDP/LDP для туннеля TE и интерфейса с переключением тегов. В примере, показанном в [Решении](#), TDP включен на Tunnel0 PE1. P2 настроен для принятия направленного hellos и формирования направленных соседей TDP. Так, PE1 получает метку для 10.11.11.11 от P2 через LDP. Теперь, когда Tunnel0 был сделан коммутируемым интерфейсом метки, и TDP был включен для трафика к 10.11.11.11, PE1 использует обоим метки; это использует метку RSVP для достижения окончательного устройства TE и метки TDP для достижения 10.11.11.11.

Если эти элементы истинны, в этом сценарии PE1 использует стек меток {L2 L3 L1} для передачи данных к CE2:

- L1 является меткой VPN.
- L2 является меткой RSVP для достижения окончательного устройства TE.
- L3 является меткой TDP для достижения 10.11.11.11 (полученный от P2).

[Решение](#)

Решением является включение TDP в туннеле TE.

[!--- конфигурацию](#)

Показанный здесь конфигурация Туннеля TE на PE1 с Доступным tdp на нем. Добавления находятся в полужирном шрифте.

```
PE 1
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 !
```

Это - дополнительная настройка на окончании Туннеля TE для принятия направленного hellos TDP:

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

[Проверка](#)

```
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.
```

Когда P1 получает пакет со стеком меток {19 18 12308}, это выталкивает метку 19 и передает пакет со стеком меток {18 12308} к P2. P2 проверяет свой LFIВ для метки 18, затем выталкивает метку и передает его по PO2/0/0 исходящего интерфейса к PE1. PE1 получает пакет с меткой 12308 и успешно переключает его на 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
```

Команда ping от CE1 до хоста на CE2 подтверждает решение.

```
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#
```

Случай 3: VPN между CE1 и CE2 по туннелю TE от P1 до P2, когда не включен TDP/LDP

Топология

!--- конфигурацию

Конфигурацию туннеля на PE1 показывают здесь:

PE 1
<pre>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</pre>

Проверка

Проверьте, как пакеты, предназначенные к CE2 172.16.13.13, коммутированы здесь. Выходные данные команды **show ip cef** показывают, что пакеты назначению 172.16.13.13 коммутированы со стеклом меток {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}
```

Когда P1 получает этот пакет, он удаляет внешнюю метку 17 и переключает пакет после взгляда в таблице IP-маршрутизации к Tunnel0. Заметьте **implicit-null OutLabel** в этих выходных данных; это означает, что исходящий интерфейс не с коммутацией по меткам.

```
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
```

Как только P2 получает пакет с меткой 12308, он сверяется с таблицей переадресации. Поскольку нет никакого способа, которым P2 может знать о метке VPN 12308 от CE2, это отбрасывает пакет.

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

Это ломает путь пакетов VPN, предназначенный к CE2. Это подтверждено эхо-запросом к 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#
```

Решение

Решение включает LDP/TDP через туннель. Это решение обсуждается в следующем разделе.

Пример 4: VPN по туннелю TE между P1 и P2 с включенным LDP

Топология

!--- конфигурацию

С LDP, включенным на туннеле, конфигурации на P1 появляются как показано здесь. Добавления находятся в полужирном шрифте.

```
PE 1
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 !
```

Проверка

PE1 передает пакеты для добавления префикса 172.16.13.13/32 стек меток {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 получает пакет со стеком меток {17 12308} и посмотрел на его LFIB для метки 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}
```

Это показывает, что метка 17 должна быть подкачана к метке 18. Поэтому тот пакет переключен туннельный интерфейс со стеком меток {18 12308}.

P2 получает пакет по своему туннельному интерфейсу со стеком меток {18 12308}. Он отображает метку 18 (так как является предпоследним маршрутизатором перехода) и направляет пакет на PE2 с меткой 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 получает пакет с меткой 12308, которая переключает пакет на CE2 успешно.

```
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#
```

Случай 5: MPLS VPN по туннелю между P1 и PE2

Топология

!--- конфигурацию

PE 1

```
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
```

Проверка

PE1 передает пакет, предназначенный к 172.16.13.13 к его следующему переходу 10.11.11.11 со стеком меток {17 12308}.

```
PE1# show ip cef vrf aqua 172.16.13.13 172.16.13.0/24, version 18, cached adjacency 10.7.7.7 0
packets, 0 bytes tag information set local tag: VPN route head fast tag rewrite with Et2/0/2,
10.7.7.7, tags imposed {17 12308} via 10.11.11.11, 0 dependencies, recursive next hop 10.7.7.7,
Ethernet2/0/2 via 10.11.11.11/32 valid cached adjacency tag rewrite with Et2/0/2, 10.7.7.7, tags
imposed {17 12308}
```

P1 получает пакет со стеком меток {17 12308}. P1 посмотрел на свою таблицу LFIB и проверяет стек метки {17} и переключает пакет с меткой {17} к 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 получает пакет со стеком меток {17 12308}. P2, являясь предпоследним маршрутизатором перехода, отбрасывает метку 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#
```

PE2 тогда получает пакет с меткой 12308. P2, знает, что напрямую подключается назначение для метки 12308. Поэтому эхо-запрос от CE1 до CE2 равняется 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#
```

Примечание: Никакой не показывают, потому что исходящий тег является . Это вызвано тем, что префикс, привязанный к метке, является маршрутом прямого соединения.

```
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#
```

Типичные ошибки

[Для получения дополнительной информации см. следующее уведомление: MPLS VPN с TE и MPLS InterAS, Консультативным на программном обеспечении Cisco IOS](#) для получения дополнительной информации.

Заключение

Когда туннель TE заканчивается на выходе PE, MPLS VPN и TE работают вместе без какой-либо дополнительной настройки. Когда Туннель TE завершен на любых маршрутизаторах P (перед PE в ядре), сбой перенаправления трафика MPLS VPN, потому что пакеты поступают

с метками VPN как внешние метки, которые не находятся в LFIB этих устройств. Таким образом, эти промежуточные маршрутизаторы не способны переадресовать пакеты в пункт конечного назначения, клиентскую сеть VPN. В таком случае LDP/TDP должен быть позволен на Туннеле TE решить проблему.

[Дополнительные сведения](#)

- [Ответы на вопросы по MPLS для начинающих](#)
- [Способы обнаружение неполадок в MPLS VPN](#)
- [Проектирование основного трафика MPLS с использованием примера конфигурации OSPF](#)
- [Выбор базовой конфигурации для MPLS VPN](#)
- [Устранение неполадок LSP в виртуальной частной сети с многопротокольной коммутацией по меткам \(MPLS VPN\)](#)
- [Страница поддержки MPLS](#)
- [Техническая поддержка - Cisco Systems](#)