

Implémentation d'un VPN MPLS sur les tunnels TE

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Conventions](#)

[Théorie générale](#)

[VPN initial installé entre CE1 et CE2 sans tunnel TE](#)

[Topologie](#)

[Configuration](#)

[Vérification](#)

[Cas 1 : VPN au-dessus d'un tunnel TE quand le tunnel TE est de PE1 à PE2](#)

[Topologie](#)

[Configuration](#)

[Vérification](#)

[Cas 2 : VPN au-dessus d'un tunnel TE quand le tunnel TE est de PE1 à P2](#)

[Topologie](#)

[Configuration](#)

[Vérification](#)

[Explication](#)

[Solution](#)

[Affaire 3 : VPN entre CE1 et CE2 au-dessus d'un tunnel TE de P1 à P2 quand TDP/LDP n'est pas activé](#)

[Topologie](#)

[Configuration](#)

[Vérification](#)

[Solution](#)

[Affaire 4 : VPN au-dessus d'un tunnel TE entre P1 et P2 avec le LDP activé](#)

[Topologie](#)

[Configuration](#)

[Vérification](#)

[Affaire 5 : MPLS VPN au-dessus d'un tunnel entre P1 et PE2](#)

[Topologie](#)

[Configuration](#)

[Vérification](#)

[Problèmes identifiés](#)

[Conclusion](#)

[Introduction](#)

Ce document propose des exemples de configurations pour mettre en œuvre un VPN à Commutation multiprotocole par étiquette (MPLS) dans des tunnels d'ingénierie de trafic (TE) dans un réseau MPLS. Afin de gagner les avantages d'un MPLS VPN au-dessus des tunnels TE, chacun des deux devraient coexister dans le réseau. Ce document montre les divers scénarios qui expliquent pourquoi le transfert de paquet dans un MPLS VPN au-dessus des tunnels TE pourrait échouer. Il propose également une solution possible.

[Conditions préalables](#)

[Conditions requises](#)

Les lecteurs de ce document devraient avoir connaissance des sujets suivants :

- [Ingénierie de trafic MPLS et améliorations](#)
- [Configuration d'un VPN MPLS de base](#)

[Composants utilisés](#)

Ce document n'est pas limité à des versions de matériel et de logiciel spécifiques.

[Conventions](#)

Pour plus d'informations sur les conventions utilisées dans ce document, reportez-vous à [Conventions relatives aux conseils techniques Cisco](#).

[Théorie générale](#)



Suivant les indications de cette topologie, en configuration du VPN simple MPLS, le Provider Edge 1 (PE1) apprend l'étiquette VPN (étiquette 1 [L1]) pour le préfixe 172.16.13.0/24 VPN par l'intermédiaire du protocole BGP multiprotocole (MPBGP) de PE2 directement, avec le prochain saut comme adresse de bouclage PE2. PE1 apprend également l'étiquette (L2) pour l'adresse de bouclage PE2 par l'intermédiaire du protocole de distribution d'étiquette (LDP) de son prochain saut P1.

En expédiant des données au VPN préfixez 172.16.13.13, PE1 utilise une pile d'étiquette {L2 L1} avec L2 comme étiquette externe. L2 obtient troqué par le routeur de commutateur d'étiquette de transit (LSR), P1. P2 saute le L2 externe et en avant le paquet à PE2 avec seulement un L1. Pour comprendre mieux pourquoi P2 saute L2, référez-vous à la section 3.16 au sujet du saut pénultième en sautant (PHP) dans [RFC 3031](#) . [Ainsi, les paquets au préfixe 172.16.13.0/24 de la](#)

[version d'IP 4 VPN \(ipv4\) sont étiquette commutée au-dessus d'un réseau MPLS.](#)

L'opération à terme MPLS VPN échoue si n'importe quel routeur P reçoit le paquet avec L1 (étiquette VPN) comme seule étiquette externe au lieu de la pile d'étiquette {L2 L1}. Ceci se produit parce qu'aucun des Routeurs P n'a L1 dans son Forwarding Information Base d'étiquette (LFIB) pour commuter le paquet.

Un MPLS TE emploie le Protocole RSVP (Resource Reservation Protocol) pour permuter des étiquettes. Quand un routeur est configuré pour TE et protocole de distribution de balise (Protocole TDP) /LDP, le routeur reçoit les étiquettes différentes du LDP et du RSVP pour un préfixe donné. Les étiquettes du LDP et du RSVP n'ont pas besoin d'être identiques dans toutes les situations. Le routeur installe une étiquette LDP dans la table d'expédition si le préfixe est appris par une interface LDP, et il installe l'étiquette de RSVP dans la table d'expédition si le préfixe est appris au-dessus d'une interface de tunnel TE.

Dans le cas d'un tunnel de la brute TE (sans LDP/TDP activé sur le tunnel), le d'entrée LSR (le LSR sur le headend du tunnel TE) utilise la même étiquette qu'est utilisé pour atteindre le tailend du tunnel TE pour toutes les artères qui sont apprises par un tunnel TE.

Par exemple, il y a un tunnel TE de PE1 à P2 apprenant le préfixe 10.11.11.11/32 au-dessus du tunnel. Le tailend de tunnel sur P2 est 10.5.5.5, et l'étiquette pour atteindre 10.5.5.5 dans PE1 est L3. PE1 emploie alors L3 pour atteindre la destination 10.11.11.11/32, apprise au-dessus du tunnel TE.

Dans le scénario [ci-dessus](#), quand il y a un tunnel TE entre PE1 et P2, considérez ces en avant données PE1 au Customer Edge 2 (CE2). Si L4 est l'étiquette VPN, PE1 en avant les données avec la pile d'étiquette {L3 L4}. P1 saute L3, et P2 reçoit le paquet avec L4. PE2 est le seul LSR qui peut correctement expédier le paquet avec l'étiquette externe L4. P2 n'a pas une session MPBGP avec PE2, ainsi il ne reçoit pas le L4 de PE2. Par conséquent, P2 n'a aucune connaissance de L2, et il relâche le paquet.

Les configurations et les sorties d'**exposition** qui suivent expliquent ceci et illustrent une solution au problème possible.

[VPN initial installé entre CE1 et CE2 sans tunnel TE](#)

[Topologie](#)



[Configuration](#)

Seulement les éléments pertinents des fichiers de configuration sont inclus ici :

```
PE1
hostname PE1
```

```
ip cef
!
ip vrf aqua
  rd 100:1
  route-target export 1:1
  route-target import 1:1
!
mpls traffic-eng tunnels
!
interface Loopback0
  ip address 10.2.2.2 255.255.255.255
  no ip directed-broadcast
!
interface Ethernet2/0/1
  ip vrf forwarding aqua
  ip address 172.16.1.2 255.255.255.0
!
interface Ethernet2/0/2
  ip address 10.7.7.2 255.255.255.0
  ip router isis
  mpls traffic-eng tunnels
  tag-switching ip
!
router isis
  passive-interface Loopback0
  net 47.1234.2222.2222.2222.00
  is-type level-1
  metric-style wide
  mpls traffic-eng router-id Loopback0
  mpls traffic-eng level-1
!
router bgp 1
  bgp log-neighbor-changes
  neighbor 10.11.11.11 remote-as 1
  neighbor 10.11.11.11 update-source Loopback0
  !
  address-family vpnv4
  neighbor 10.11.11.11 activate
  neighbor 10.11.11.11 send-community extended
  exit-address-family
  !
  address-family ipv4
  neighbor 10.11.11.11 activate
  no auto-summary
  no synchronization
  exit-address-family
  !
  address-family ipv4 vrf aqua
  redistribute connected
  no auto-summary
  no synchronization
  exit-address-family
```

PE2

```
hostname PE2
!
ip vrf aqua
  rd 100:1
  route-target export 1:1
  route-target import 1:1
!
mpls traffic-eng tunnels
```

```

!
interface Loopback0
 ip address 10.11.11.11 255.255.255.255
!
interface POS0/1
 ip address 10.12.12.10 255.255.255.0
 ip router isis
 mpls traffic-eng tunnels
 tag-switching ip
 crc 16
 clock source internal
!
interface POS5/1
 ip vrf forwarding aqua
 ip address 172.16.13.11 255.255.255.0
 crc 32
 clock source internal
!
router isis
 passive-interface Loopback0
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng level-1
 net 47.1234.1010.1010.1010.00
 is-type level-1
 metric-style wide
!
router bgp 1
 bgp log-neighbor-changes
 neighbor 10.2.2.2 remote-as 1
 neighbor 10.2.2.2 update-source Loopback0
 no auto-summary
!
 address-family vpnv4
 neighbor 10.2.2.2 activate
 neighbor 10.2.2.2 send-community extended
 exit-address-family
!
 address-family ipv4 vrf aqua
 redistribute connected
 no auto-summary
 no synchronization
 exit-address-family
!

```

Vérification

PE2 apprend le préfixe 172.16.1.0/24 MPBGP fini d'ipv4 PE1 VPN scrutant entre PE1 et PE2.
Ceci est affiché ici :

```
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

De même, PE1 apprend le préfixe 172.16.13.0/24 MPBGP fini d'ipv4 PE2 VPN scrutant entre PE1 et PE2. Ceci est affiché ici :

```
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.
```

Ainsi, CE1 peut atteindre 172.16.13.13 sur le réseau CE2 par l'intermédiaire de l'exemple « aqua » de routage et d'expédition VPN (VRF), qui est configuré sur PE1 utilisant la pile d'étiquette {17 12308}, comme affiché ci-dessus.

Cette sortie de ping confirme la Connectivité :

```
CE1# ping 172.16.13.13
```

```
Type escape sequence to abort.
```

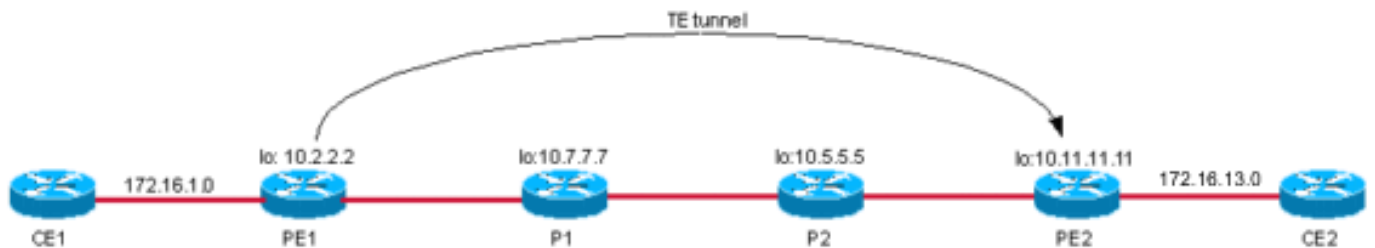
Sending 5, 100-byte ICMP Echos to 172.16.13.13, timeout is 2 seconds:

!!!!!

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

Cas 1 : VPN au-dessus d'un tunnel TE quand le tunnel TE est de PE1 à PE2

Topologie



Quand le tunnel TE est construit entre les Routeurs de PE avec l'autoroute annonce utilisée, saut BGP de PE de sortie le prochain est accessible par l'intermédiaire de l'interface de tunnel TE. Ainsi, PE1 emploie l'étiquette TE pour atteindre PE2.

Remarque: MPLS TE est indépendant de LDP, ainsi il signifie que, si vous avez un maillage complet des tunnels du PE au PE, vous pouvez efficacement désactiver le LDP dans les Routeurs et n'avez pas besoin d'exécuter le LDP sur les interfaces de tunnel TE. Cependant, vous devez construire tous les tunnels au prochain saut BGP des artères de la version 4 VPN (VPNv4). Dans l'exemple dans cette [configuration](#), vous pouvez voir que ce prochain saut BGP est le Loopback0 sur PE2, 10.11.11.11. Ce même bouclage est également la destination de tunnel pour le tunnel de PE1 à PE2. Ceci explique pourquoi, dans cet exemple, s'il y a également un tunnel de PE2 à PE1 pour le trafic de retour, vous pouvez désactiver le LDP au centre. Puis, expédiant du CE aux travaux de la CE avec tout le trafic VPNv4 a reporté les tunnels TE. Si le prochain saut BGP n'est pas identique que la destination de tunnel TE, LDP doit être exécuté au centre et sur le tunnel TE.

Configuration

La configuration supplémentaire sur PE1 pour établir un tunnel de PE est affichée ici :

```
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
```

Vérification

```

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

```

Maintenant, confirmez l'étiquette externe utilisée pour atteindre le prochain saut 10.11.11.11 par l'intermédiaire de 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

```

Une autre manière de visualiser ces informations est rapidement d'utiliser les modificateurs de sortie dans les commandes show, comme affiché ici :

```

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.

```

Regardez la pile de balise. Il est 19, qui est l'étiquette TE, utilisé pour expédier des paquets au prochain saut 10.11.11.0 au-dessus de 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#
Ainsi, PE1 envoie un paquet destiné à 172.16.13.13 avec la pile d'étiquette {19 12308}. P1 permute l'étiquette 19. Le paquet atteint P2, qui saute qu'externe étiquetez. Puis, le paquet est expédié à PE2 avec seulement l'étiquette 12308.

Sur PE2, le paquet avec l'étiquette 12308 est reçu et commuté selon les informations dans la table d'expédition. Ceci est affiché ici :

```

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#
Remarque: Aucune interface sortante n'est affichée parce que la balise sortante est d'agrégat. C'est parce que le préfixe associé avec l'étiquette est directement la route connectée.

Les pings de CE1 à un hôte sur CE2 confirment la connectivité VPN au-dessus du tunnel TE :

```

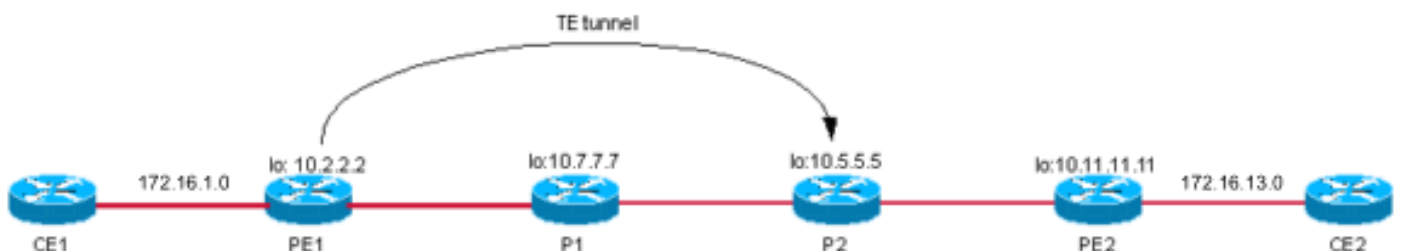
CE1# ping 172.16.13.13

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#

```

Cas 2 : VPN au-dessus d'un tunnel TE quand le tunnel TE est de PE1 à P2

Topologie



Configuration

La configuration supplémentaire TE au-dessus de la configuration de base sur PE1 est affichée ici :

```
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
!
```

Vérification

Vérifiez l'artère pour préfixer 172.16.13.13 sur l'aqua du VRF PE1. Il indique le prochain saut 10.11.11.11/32 (au-dessus de Tunnel0) utilisant la pile d'étiquette {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#
```

L'étiquette 19, l'étiquette externe, est utilisée pour atteindre le prochain saut 10.11.11.11/32, comme affiché ici :

```
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#
```

Le paquet de PE1 est envoyé au-dessus du tunnel TE avec la pile d'étiquette {19 12308}. Une fois que P1 reçoit le paquet, il saute (PHP) la balise 19 et envoie le paquet avec la pile d'étiquette {12308}. La commande show confirme ceci :

```
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>
```

Quand P2 reçoit le paquet avec la pile d'étiquette {12308}, il vérifie son LFIB et relâche le paquet parce qu'aucune correspondance n'existe. C'est la sortie de commande show sur 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#

```

Explication

La solution au problème est d'activer TDP/LDP sur le tunnel TE et de lui faire une interface balise-commutée. Dans l'exemple présenté dans la [solution](#), le Protocole TDP est activé sur le Tunnel0 de PE1. P2 est configuré pour recevoir des hellos dirigés et former les voisins dirigés de Protocole TDP. Ainsi, PE1 reçoit l'étiquette pour 10.11.11.11 de P2 par l'intermédiaire du LDP. Maintenant que Tunnel0 a été fait à une interface balise-commutée et le Protocole TDP a été activé pour le trafic à 10.11.11.11, PE1 utilise chacun des deux les étiquettes ; il emploie l'étiquette de RSVP pour atteindre le tailend TE et l'étiquette de Protocole TDP pour atteindre 10.11.11.11.

Dans ce scénario, PE1 utilise la pile d'étiquette {L2 L3 L1} pour expédier des données à CE2 si ces éléments sont vrais :

- L1 est l'étiquette VPN.
- L2 est l'étiquette de RSVP pour atteindre le tailend TE.
- L3 est l'étiquette de Protocole TDP pour atteindre 10.11.11.11 (reçu de P2).

Solution

La solution est d'activer le Protocole TDP à travers le tunnel TE.

Configuration

Affichée ici est la configuration de tunnel TE sur PE1 avec le Protocole TDP activé là-dessus. Les ajouts sont en caractères gras.

```

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 !

```

C'est la configuration supplémentaire sur le tailend du tunnel TE pour recevoir des hellos dirigés

de Protocole TDP :

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

Vérification

```
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.
```

Quand P1 reçoit le paquet avec la pile d'étiquette {19 18 12308}, il saute la balise 19 et envoie le paquet avec la pile d'étiquette {18 12308} à P2. P2 vérifie son LFIB pour l'étiquette 18, puis saute la balise et l'envoie au-dessus de l'interface sortante PO2/0/0 vers PE1. PE1 reçoit le paquet avec l'étiquette 12308 et le commute avec succès à 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/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
```

Une commande ping de CE1 à un hôte sur CE2 confirme la solution.

```
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:
```

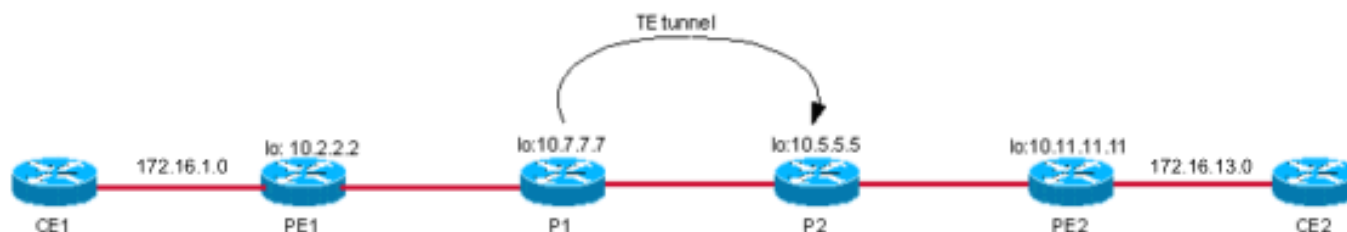
```
!!!!!
```

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

```
CE1#
```

Affaire 3 : VPN entre CE1 et CE2 au-dessus d'un tunnel TE de P1 à P2 quand TDP/LDP n'est pas activé

Topologie



Configuration

La configuration de tunnel sur PE1 est affichée ici :

PE1
PE1# 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

```

Vérification

Vérifiez comment les paquets destinés à CE2 172.16.13.13 obtiennent commuté ici. La sortie de commande de **show ip cef** prouve que des paquets à la destination 172.16.13.13 sont commutés avec la pile d'étiquette {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}

```

Quand P1 reçoit ce paquet, il retire l'étiquette externe 17 et commute le paquet après regard dans la table de Routage IP à Tunnel0. Notez l'implicite-null OutLabel dans cette sortie ; il signifie que l'interface sortante n'est pas étiquette commutée.

```

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

```

Une fois que P2 reçoit le paquet avec l'étiquette 12308, il regarde sa table d'expédition. Puisqu'il y a aucune manière que P2 ne peut se rendre compte de la balise 12308 VPN de CE2, il relâche le paquet.

```

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

```

Ceci casse le chemin des paquets VPN destinés à CE2. Il est confirmé par le ping à 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#

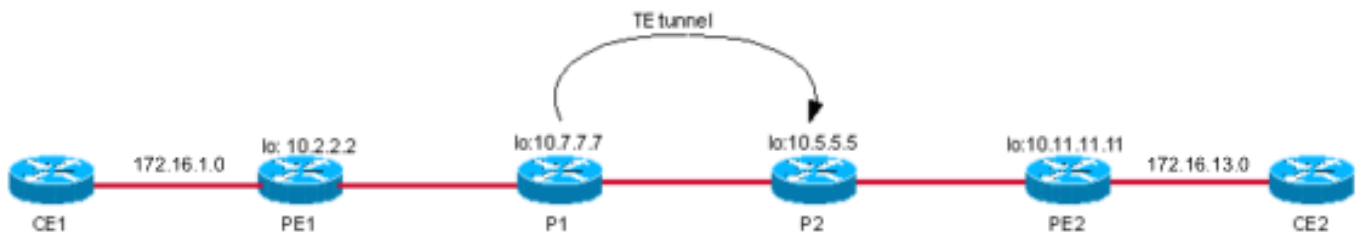
```

Solution

La solution est d'activer LDP/TDP au-dessus du tunnel. La section suivante discute cette solution.

Affaire 4 : VPN au-dessus d'un tunnel TE entre P1 et P2 avec le LDP activé

Topologie



Configuration

Le LDP étant activé sur le tunnel, les configurations sur P1 apparaissent comme affiché ici. Les ajouts sont en caractères gras.

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
!

```

Vérification

PE1 envoie des paquets pour préfixer 172.16.13.13/32 avec la pile d'étiquette {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 reçoit le paquet avec la pile d'étiquette {17 12308} et regarde son LFIB pour l'étiquette 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}

```

Il prouve que l'étiquette 17 devrait être permutée pour étiqueter 18. Par conséquent, ce paquet est commuté au-dessus de l'interface de tunnel avec la pile d'étiquette {18 12308}.

P2 reçoit le paquet au-dessus de son interface de tunnel avec la pile d'étiquette {18 12308}. Il saute la balise 18 (parce que c'est le routeur pénultième de saut) et commute le paquet à PE2 avec l'étiquette 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 reçoit le paquet avec l'étiquette 12308, qui commute le paquet à CE2 avec succès.

```

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

```

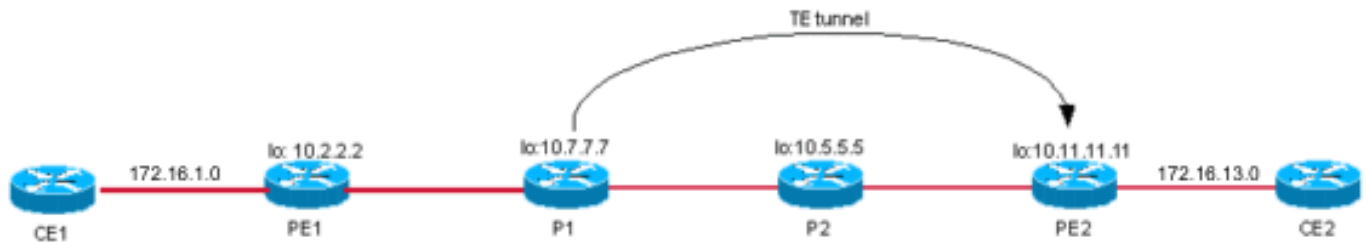
```

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#

```

[Affaire 5 : MPLS VPN au-dessus d'un tunnel entre P1 et PE2](#)

[Topologie](#)



Configuration

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

```

Vérification

PE1 envoie un paquet destiné à 172.16.13.13 à son prochain saut 10.11.11.11 avec la pile d'étiquette {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 reçoit le paquet avec la pile d'étiquette {17 12308}. P1 regarde sa table LFIB et vérifie la pile {17} de balise et commute le paquet avec l'étiquette {17} vers 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 reçoit le paquet avec la pile d'étiquette {17 12308}. P2, étant le routeur pénultième de saut, étiquette 17 de bruits.

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 reçoit alors le paquet avec l'étiquette 12308. P2 se rend compte que la destination pour l'étiquette 12308 soit directement connectée. Par conséquent, le ping de CE1 à CE2 est 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#

Remarque: Aucune interface sortante n'est affichée parce que la balise sortante est d'agrégat. C'est parce que le préfixe associé avec l'étiquette est directement la route connectée.

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#

Problèmes identifiés

Référez-vous aux [notes de terrain : MPLS VPN avec le bulletin de renseignements TE et MPLS InterAS sur le logiciel de Cisco IOS®](#) pour plus de détails.

Conclusion

Quand le tunnel TE est terminé sur le PE de sortie, le MPLS VPN et le travail TE ensemble sans n'importe quelle configuration supplémentaire. Quand le tunnel TE est terminé sur tous les Routeurs P (avant le PE au centre), l'expédition du trafic VPN MPLS échoue parce que les paquets arrivent avec des étiquettes VPN comme étiquettes externes, qui ne sont pas dans le LFIBs de ces périphériques. Par conséquent, ces routeurs intermédiaires ne peuvent pas expédier des paquets à la destination définitive, le réseau client VPN. En pareil cas, LDP/TDP devrait être activé sur le tunnel TE résoudre le problème.

Informations connexes

- [MPLS FAQ For Beginners](#)
- [Comment dépanner le VPN MPLS](#)
- [Exemple de configuration d'ingénierie du trafic MPLS de base à l'aide de la configuration OSPF](#)
- [Configuration d'un VPN MPLS de base](#)
- [Dépannage des défaillances LSP dans VPN MPLS](#)
- [MPLS Support Page](#)
- [Support technique - Cisco Systems](#)