

# Implementazione di una VPN MPLS sui tunnel TE

## Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Convenzioni](#)

[Nozioni di base](#)

[Configurazione VPN iniziale tra CE1 e CE2 senza tunnel TE](#)

[Topologia](#)

[Configurazione](#)

[Verifica](#)

[Caso 1: VPN su tunnel TE quando il tunnel TE viene da PE1 a PE2](#)

[Topologia](#)

[Configurazione](#)

[Verifica](#)

[Caso 2: VPN su tunnel TE quando il tunnel TE viene da PE1 a P2](#)

[Topologia](#)

[Configurazione](#)

[Verifica](#)

[Spiegazione](#)

[Soluzione](#)

[Caso 3: VPN tra CE1 e CE2 su un tunnel TE da P1 a P2 quando TDP/LDP non è abilitato](#)

[Topologia](#)

[Configurazione](#)

[Verifica](#)

[Soluzione](#)

[Caso 4: VPN su tunnel TE tra P1 e P2 con LDP abilitato](#)

[Topologia](#)

[Configurazione](#)

[Verifica](#)

[Caso 5: VPN MPLS su tunnel tra P1 e PE2](#)

[Topologia](#)

[Configurazione](#)

[Verifica](#)

[Problemi noti](#)

[Conclusioni](#)

[Informazioni correlate](#)

## Introduzione

In questo documento vengono fornite configurazioni di esempio per l'implementazione di una VPN MPLS (Multiprotocol Label Switching) su tunnel TE (Traffic Engineering) di una rete MPLS. Per ottenere i vantaggi di una VPN MPLS sui tunnel TE, entrambi devono coesistere nella rete. Questo documento illustra vari scenari che spiegano perché l'inoltro di pacchetti all'interno di una VPN MPLS sui tunnel TE potrebbe non riuscire. Offre inoltre una possibile soluzione.

## Prerequisiti

### Requisiti

Questo documento è utile per conoscere i seguenti argomenti:

- [Progettazione e miglioramenti del traffico MPLS](#)
- [Configurazione di una VPN MPLS di base](#)

### Componenti usati

Il documento può essere consultato per tutte le versioni software o hardware.

### Convenzioni

Per ulteriori informazioni sulle convenzioni usate, consultare il documento [Cisco sulle convenzioni nei suggerimenti tecnici](#).

### Nozioni di base



Come illustrato in questa topologia, in una configurazione VPN MPLS semplice, il provider Edge 1 (PE1) apprende direttamente l'etichetta VPN (etichetta 1 [L1]) per il prefisso VPN 172.16.13.0/24 tramite il protocollo MPBGP (Multiprotocol Border Gateway Protocol) da PE2, con l'hop successivo come indirizzo di loopback PE2. PE1 apprende inoltre l'etichetta (L2) per l'indirizzo di loopback PE2 tramite il protocollo LDP (Label Distribution Protocol) dall'hop successivo P1.

Quando si inoltrano i dati al prefisso VPN 172.16.13.13, PE1 utilizza uno stack di etichette {L2 L1} con L2 come etichetta esterna. L2 viene scambiato dal router LSR (transit label switch router), P1. P2 apre l'L2 esterno e inoltra il pacchetto al PE2 con un solo L1. Per comprendere meglio il motivo per cui P2 fa saltare L2, fare riferimento alla sezione 3.16 sul php (penultimate hop popping) nella [RFC 3031](#). Pertanto, i pacchetti al prefisso VPN IP versione 4 (IPv4) 172.16.13.0/24 hanno un'etichetta scambiata su una rete MPLS.

L'operazione di inoltro VPN MPLS non riesce se un router IP riceve il pacchetto con l'etichetta L1 (etichetta VPN) come unica etichetta esterna anziché lo stack di etichette {L2/L1}. Questo si

verifica perché nessuno dei router IP dispone di L1 nella base LFIB (Label Forwarding Information Base) per commutare il pacchetto.

Un TE MPLS utilizza il protocollo RSVP (Resource Reservation Protocol) per scambiare le etichette. Quando un router è configurato sia per TE che per TDP (Tag Distribution Protocol)/LDP, riceve etichette diverse da LDP e RSVP per un dato prefisso. Le etichette LDP e RSVP non devono essere le stesse in tutte le situazioni. Il router installa un'etichetta LDP nella tabella di inoltro se il prefisso viene appreso tramite un'interfaccia LDP e installa l'etichetta RSVP nella tabella di inoltro se il prefisso viene appreso tramite un'interfaccia del tunnel TE.

Nel caso di un tunnel TE normale (senza LDP/TDP abilitato sul tunnel), l'LSR in entrata (l'LSR sull'headend del tunnel TE) usa la stessa etichetta utilizzata per raggiungere la coda del tunnel TE per tutti i percorsi appresi attraverso un tunnel TE.

Ad esempio, esiste un tunnel TE da PE1 a P2 che apprende il prefisso 10.11.11.11/32 sul tunnel. La coda del tunnel su P2 è 10.5.5.5 e l'etichetta per raggiungere 10.5.5.5 in PE1 è L3. PE1 quindi utilizza L3 per raggiungere la destinazione 10.11.11.11/32, appresa attraverso il tunnel TE.

Nello scenario [precedente](#), quando è presente un tunnel TE tra PE1 e P2, tenere presente che PE1 inoltra i dati al Customer Edge 2 (CE2). Se L4 è l'etichetta VPN, PE1 inoltra i dati con lo stack di etichette {L3 L4}. P1 apre L3 e P2 riceve il pacchetto con L4. PE2 è l'unico LSR in grado di inoltrare correttamente il pacchetto con l'etichetta esterna L4. P2 non dispone di una sessione MPBGP con PE2, quindi non riceve L4 da PE2. Pertanto, P2 non ha alcuna conoscenza di L2 e scarta il pacchetto.

Le configurazioni e gli output **mostrati di seguito** lo dimostrano e illustrano una possibile soluzione al problema.

## Configurazione VPN iniziale tra CE1 e CE2 senza tunnel TE

### Topologia



### Configurazione

Sono incluse solo le parti pertinenti dei file di configurazione:

```
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
!

```

## Verifica

PE2 apprende il prefisso 172.16.1.0/24 della VPN IPv4 PE1 sul peering MPBGP tra PE1 e PE2. Di seguito è illustrato:

```
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
```

Analogamente, PE1 apprende il prefisso 172.16.13.0/24 IPv4 della VPN PE2 sul peering MPBGP tra PE1 e PE2. Di seguito viene illustrato:

```
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.*

Pertanto, CE1 può raggiungere la versione 172.16.13.13 sulla rete CE2 tramite l'istanza "aqua" di VRF (VPN routing and forwarding), configurata su PE1 utilizzando lo stack di etichette {17 12308}, come mostrato sopra.

Questo output del comando **ping** conferma la connettività:

CE1# **ping 172.16.13.13**

Type escape sequence to abort.

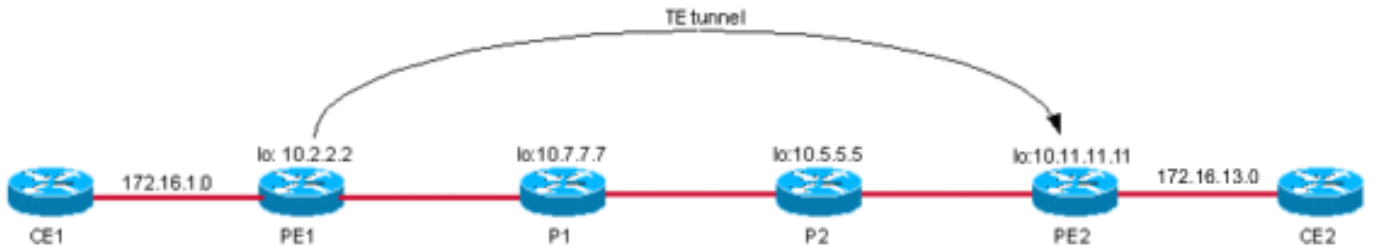
Sending 5, 100-byte ICMP Echos to 172.16.13.13, timeout is 2 seconds:

!!!!!

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

**Caso 1: VPN su tunnel TE quando il tunnel TE viene da PE1 a PE2**

## Topologia



Quando il tunnel TE viene creato tra i router PE con l'annuncio di routing automatico utilizzato, l'hop successivo PE in uscita è raggiungibile tramite l'interfaccia del tunnel TE. PE1 utilizza pertanto l'etichetta TE per raggiungere PE2.

**Nota:** MPLS TE è indipendente da LDP, il che significa che, se si dispone di una rete completa di tunnel da PE a PE, è possibile disabilitare LDP nei router in modo efficace e non è necessario eseguirlo sulle interfacce del tunnel TE. Tuttavia, è necessario compilare tutti i tunnel per l'hop successivo BGP delle route VPN versione 4 (VPNv4). Nell'esempio di questa [configurazione](#), è possibile vedere che l'hop successivo BGP è il loopback0 su PE2, 10.11.11.11. Lo stesso loopback è anche la destinazione del tunnel da PE1 a PE2. Questo spiega perché, in questo esempio, se esiste anche un tunnel da PE2 a PE1 per il traffico di ritorno, è possibile disabilitare LDP nel core. Quindi, l'inoltro da CE a CE funziona con tutto il traffico VPNv4 trasportato sui tunnel TE. Se l'hop successivo BGP non è la stessa destinazione del tunnel TE, LDP deve essere eseguito nel core e sul tunnel TE.

## Configurazione

La configurazione aggiuntiva in PE1 per stabilire un tunnel PE è illustrata di seguito:

```
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
```

## Verifica

```
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

```

A questo punto, confermare l'etichetta esterna usata per raggiungere l'hop successivo 10.11.11.11 tramite 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

```

Un altro modo per visualizzare rapidamente queste informazioni è utilizzare i modificatori di output nei comandi **show**, come mostrato di seguito:

```

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.

```

Osservare la pila di tag. Il valore 19, che è l'etichetta TE, viene usata per inoltrare i pacchetti all'hop successivo 10.11.11.0 su 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#
```

Pertanto, PE1 invia un pacchetto destinato alla versione 172.16.13.13 con lo stack di etichette {19.12308}. P1 scambia l'etichetta 19. Il pacchetto raggiunge P2, che apre l'etichetta esterna. Il pacchetto viene quindi inoltrato a PE2 solo con l'etichetta 12308.

In PE2, il pacchetto con etichetta 12308 viene ricevuto e commutato in base alle informazioni nella tabella di inoltro. Di seguito viene illustrato come eseguire questa operazione:

```
PE2# show tag for tags 12308 detail
Local   Outgoing   Prefix          Bytes tag   Outgoing   Next Hop
tag     tag or VC  or Tunnel Id    switched   interface
12308   Aggregate  172.16.13.0/24[V] 12256
        MAC/Encaps=0/0, MTU=0, Tag Stack{}
        VPN route: aqua
        No output feature configured
Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
PE2#
```

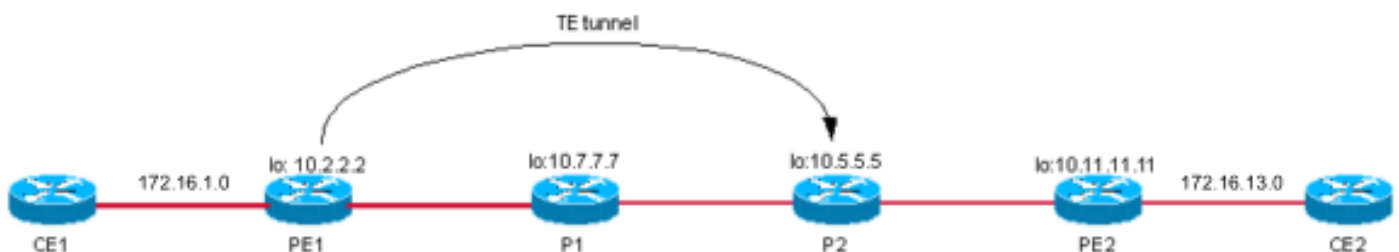
**Nota:** non viene visualizzata alcuna interfaccia in uscita perché il tag in uscita è Aggregate. Infatti il prefisso associato all'etichetta è la route connessa direttamente.

I ping tra CE1 e un host su CE2 confermano la connettività VPN sul 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#
```

## Caso 2: VPN su tunnel TE quando il tunnel TE viene da PE1 a P2

### Topologia



### Configurazione

Di seguito è riportata la configurazione TE aggiuntiva rispetto alla configurazione di base in PE1:

```
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
!
```

## Verifica

Controllare la route al prefisso 172.16.13.13 su PE1 VRF aqua. Fa riferimento all'hop successivo 10.11.11.11/32 (su Tunnel0) con stack di etichette {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'etichetta 19, l'etichetta esterna, viene usata per raggiungere l'hop successivo 10.11.11.11/32, come mostrato di seguito:

```
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#

```

Il pacchetto da PE1 viene inviato tramite il tunnel TE con lo stack di etichette {19 12308}. Una volta ricevuto il pacchetto, P1 scarta (PHP) il tag 19 e invia il pacchetto con lo stack di etichette {12308}. Il comando **show** conferma:

```
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>
```

```
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>

```

Quando P2 riceve il pacchetto con lo stack di etichette {12308}, controlla il relativo LFIB e scarta il pacchetto perché non esiste alcuna corrispondenza. Di seguito viene riportato l'output del comando **show** su P2:

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

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

```
P2#
```

```
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#
```

## Spiegazione

Per risolvere questo problema, abilitare TDP/LDP sul tunnel TE e trasformarlo in un'interfaccia a commutazione di tag. Nell'esempio illustrato nella [soluzione](#), il protocollo TDP è abilitato sul tunnel 0 di PE1. Il protocollo P2 è configurato per accettare hellos diretti e formare router adiacenti diretti. Pertanto, PE1 riceve l'etichetta per 10.11.11.11 da P2 tramite LDP. Ora che Tunnel0 è stata resa un'interfaccia a commutazione di tag e TDP è stato abilitato per il traffico fino alla versione 10.11.11.11, PE1 utilizza entrambe le etichette; usa l'etichetta RSVP per raggiungere la coda TE e l'etichetta TDP per raggiungere 10.11.11.11.

In questo scenario, PE1 utilizza lo stack di etichette {L2 L3 L1} per inoltrare i dati a CE2 se questi elementi sono veri:

- L1 è l'etichetta VPN.
- L2 è l'etichetta RSVP che consente di raggiungere la coda TE.
- L3 è l'etichetta TDP che raggiunge 10.11.11.11 (ricevuta da P2).

## Soluzione

La soluzione è abilitare il TDP attraverso il tunnel TE.

## Configurazione

Di seguito è riportata la configurazione del tunnel TE su PE1 con TDP abilitato. Le aggiunte sono in grassetto.

```
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 !
```

Questa è la configurazione aggiuntiva sulla parte terminale del tunnel TE per accettare gli hellop TDP diretti:

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

## Verifica

```
PE1# show tag tdp neighbor | i Peer
Peer TDP Ident: 10.7.7.7:0; Local TDP Ident 10.2.2.2:0
Peer TDP Ident: 10.5.5.5:0; Local TDP Ident 10.2.2.2:0

PE1#
PE1# show ip cef vrf aqua 172.16.13.13
172.16.13.0/24, version 11
0 packets, 0 bytes
tag information set
local tag: VPN route head
fast tag rewrite with Tu0, point2point, tags imposed {19 18 12308}
via 10.11.11.11, 0 dependencies, recursive
next hop 10.5.5.5, Tunnel0 via 10.11.11.11/32
valid adjacency
tag rewrite with Tu0, point2point, tags imposed {19 18 12308}
PE1#

PE1# show mpls traffic-eng tunnels tunnel 0 | i Label
InLabel : -
OutLabel : Ethernet2/0/2, 19
!--- This is the TE label learned via RSVP. PE1# PE1# show tag tdp bind 10.11.11.11 32
tib entry: 10.11.11.11/32, rev 20
local binding: tag: 21
remote binding: tsr: 10.7.7.7:0, tag: 17
remote binding: tsr: 10.5.5.5:0, tag: 18
!--- This is the TDP label from P2.
```

Quando P1 riceve il pacchetto con lo stack di etichette {19 18 12308}, estrae il tag 19 e invia il pacchetto con lo stack di etichette {18 12308} a P2. P2 controlla l'LFIB per l'etichetta 18, quindi estrae il tag e lo invia tramite l'interfaccia in uscita PO2/0/0 verso PE1. PE1 riceve il pacchetto con etichetta 12308 e lo passa correttamente a 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
```

Un comando ping inviato da CE1 a un host su CE2 conferma la soluzione.

```
CE1# ping 172.16.13.13
```

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

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

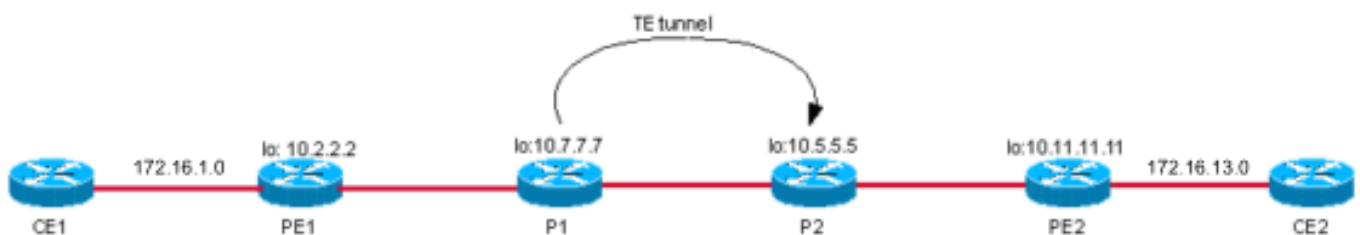
```
!!!!!
```

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

```
CE1#
```

## Caso 3: VPN tra CE1 e CE2 su un tunnel TE da P1 a P2 quando TDP/LDP non è abilitato

### Topologia



### Configurazione

La configurazione del tunnel in PE1 è illustrata di seguito:

#### 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
```

## Verifica

Verificare in che modo i pacchetti destinati a CE2 172.16.13.13 vengono commutati qui. L'output del comando **show ip cef** mostra che i pacchetti verso la destinazione 172.16.13.13 sono commutati con lo stack di etichette {17.12308}:

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

Quando P1 riceve questo pacchetto, rimuove l'etichetta esterna 17 e passa il pacchetto dopo aver cercato nella tabella di routing IP su Tunnel0. Notare l'implicit-null OutLabel in questo output; significa che l'interfaccia in uscita non è a commutazione di etichetta.

```
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
```

Una volta che P2 riceve il pacchetto con l'etichetta 12308, guarda la sua tabella di inoltro. Poiché non c'è modo per P2 di riconoscere il tag VPN 12308 da CE2, il pacchetto viene scartato.

```
P2# show tag for tag 12308 detail
```

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

```

In questo modo viene interrotto il percorso dei pacchetti VPN destinati a CE2. Viene confermato dal ping a CE2 172.16.13.13/32.

```
PE1#
CE1# ping 172.16.13.13
```

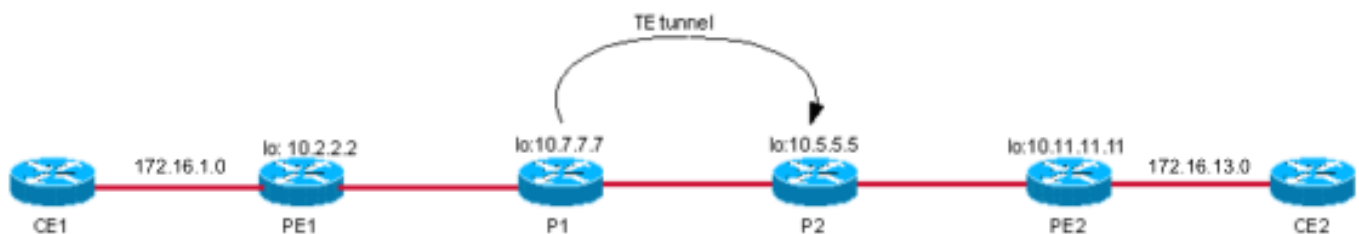
```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.13, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
CE1#
```

## Soluzione

La soluzione è abilitare LDP/TDP sul tunnel. Nella sezione successiva viene illustrata questa soluzione.

## Caso 4: VPN su tunnel TE tra P1 e P2 con LDP abilitato

### Topologia



### Configurazione

Con LDP abilitato sul tunnel, le configurazioni su P1 vengono visualizzate come mostrato di seguito. Le aggiunte sono in grassetto.

```
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
!
```

## Verifica

PE1 invia i pacchetti al prefisso 172.16.13.13/32 con lo stack di etichette {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 riceve il pacchetto con stack di etichette {17 12308} e cerca l'etichetta 17 nel relativo LFIB.

```
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}

```

Indica che l'etichetta 17 deve essere sostituita con l'etichetta 18. Pertanto, il pacchetto viene spostato sull'interfaccia del tunnel con lo stack di etichette {18 12308}.

P2 riceve il pacchetto tramite l'interfaccia del tunnel con stack di etichette {18 12308}. Viene aperto il tag 18 (in quanto è il penultimo router hop) e il pacchetto viene commutato in PE2 con l'etichetta 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 riceve il pacchetto con l'etichetta 12308, che passa il pacchetto a 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#

```

## Caso 5: VPN MPLS su tunnel tra P1 e PE2

### Topologia



## Configurazione

### 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
```

## Verifica

PE1 invia un pacchetto destinato alla versione 172.16.13.13 all'hop successivo 10.11.11.11 con lo stack di etichette {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 riceve il pacchetto con stack di etichette {17 12308}. P1 esamina la tabella LFIB e controlla lo stack di tag {17} e cambia il pacchetto con etichetta {17} in 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 riceve il pacchetto con stack di etichette {17 12308}. P2, essendo il penultimo router hop, si apre con l'etichetta 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 riceve quindi il pacchetto con l'etichetta 12308. P2 è consapevole del fatto che la destinazione dell'etichetta 12308 è collegata direttamente. Pertanto, il ping tra CE1 e CE2 è 10.

```

PE2# show tag for tag 12308 detail
Local  Outgoing  Prefix          Bytes tag  Outgoing  Next Hop
tag    tag or VC  or Tunnel Id    switched   interface
12308  Aggregate 172.16.13.0/24[V] 12776
      MAC/Encaps=0/0, MTU=0, Tag Stack{
      VPN route: aqua
      No output feature configured
      Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

```

PE2#  
**Nota:** non viene visualizzata alcuna interfaccia in uscita perché il tag in uscita è Aggregate. Infatti il prefisso associato all'etichetta è la route connessa direttamente.

```

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#

```

## Problemi noti

Per ulteriori informazioni, fare riferimento al documento [sulla comunicazione dei prodotti: MPLS VPN con TE e MPLS InterAS Advisory sul software Cisco IOS®](#) per ulteriori dettagli.

## Conclusioni

Quando il tunnel TE viene terminato sul PE in uscita, la VPN MPLS e il TE funzionano insieme senza alcuna configurazione aggiuntiva. Quando il tunnel TE viene terminato su qualsiasi router

IP (prima del PE nel core), l'inoltro del traffico VPN MPLS ha esito negativo perché i pacchetti arrivano con etichette VPN come etichette esterne, che non si trovano nei LFIB di questi dispositivi. Pertanto, questi router intermedi non sono in grado di inoltrare i pacchetti alla destinazione finale, la rete del cliente VPN. In tal caso, per risolvere il problema è necessario abilitare LDP/TDP sul tunnel TE.

## [Informazioni correlate](#)

- [Domande frequenti su MPLS per principianti](#)
- [Come risolvere i problemi della VPN MPLS](#)
- [Progettazione di base del traffico MPLS con esempio di configurazione OSPF](#)
- [Configurazione di una VPN MPLS di base](#)
- [Risoluzione dei problemi di errore LSP nella VPN MPLS](#)
- [Pagina di supporto MPLS](#)
- [Supporto tecnico – Cisco Systems](#)