

# Verificar o encaminhamento de VPN de Camada 3 do MPLS

## Contents

---

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Informações de Apoio](#)

[Conventions](#)

[Topologia](#)

[Troubleshooting](#)

[Informações iniciais](#)

[Verificação](#)

[Comandos de verificação do Cisco IOS XE](#)

[Comandos de verificação do Cisco IOS XR](#)

[Informações Relacionadas](#)

---

## Introdução

Este documento descreve o processo para verificar a conectividade fim-a-fim através de uma rede de núcleo VPN de Camada 3 MPLS.

## Pré-requisitos

### Requisitos

A Cisco recomenda que você tenha conhecimento destes tópicos:

- Conhecimento de roteamento IP básico
- Conhecimento da linha de comando do Cisco IOS® XE e do Cisco IOS® XR

### Componentes Utilizados

As informações neste documento são baseadas nestas versões de software e hardware:

- Roteador com software Cisco IOS XR
- Roteador com software Cisco IOS XE

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a rede estiver ativa, certifique-se de que você entenda o impacto

potencial de qualquer comando.

## Informações de Apoio

A finalidade deste documento é demonstrar as etapas básicas de verificação e solução de problemas para verificar a conectividade e o encaminhamento entre dois roteadores CE (Customer Edge) interconectados com o BGP (Border Gateway Protocol) por uma rede de núcleo VPN de Camada 3 MPLS com uma combinação de roteadores Cisco IOS XE e Cisco IOS XR que atuam como roteadores PE (Provider Edge) e P (Provider).

## Conventions

Para obter mais informações sobre convenções de documento, consulte as [Convenções de dicas técnicas Cisco](#).

## Topologia

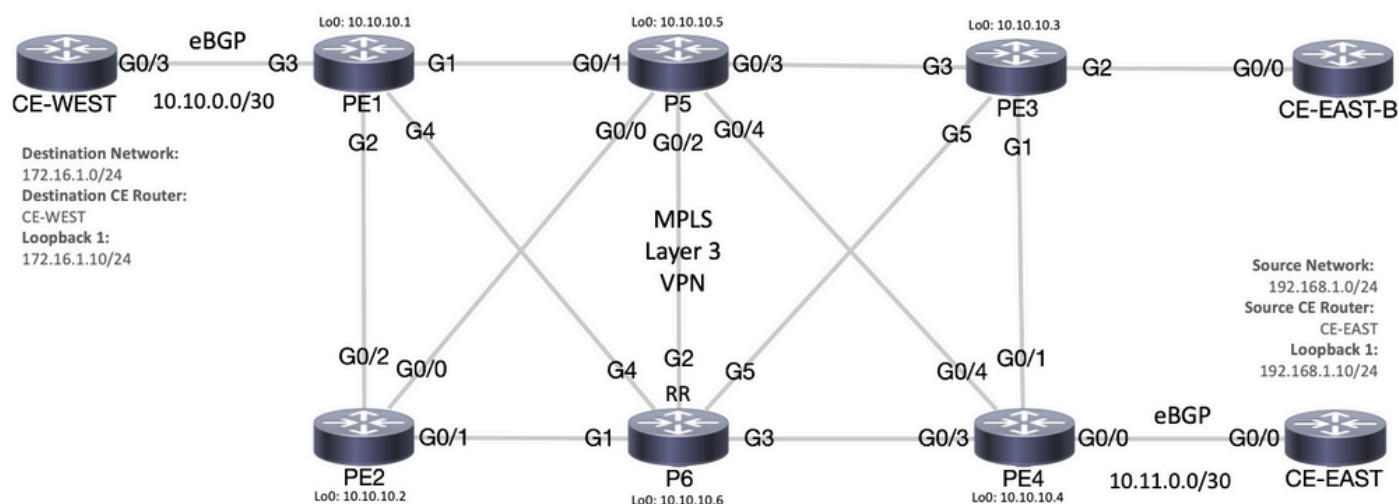


Diagrama de topologia MPLS

## Troubleshooting

### Informações iniciais

Rede de Origem: 192.168.1.0/24

Roteador CE de origem: CE-EAST

Rede de destino: 172.16.1.0/24

Roteador CE de destino: CE-WEST

Com base nas informações e na topologia iniciais, a acessibilidade deve ser bem-sucedida entre o endereço de origem 192.168.1.10 representado por Loopback1 no roteador CE-EAST e o

endereço de destino 172.16.1.10 representado por Loopback1 no roteador CE-WEST:

```
<#root>
```

```
CE-EAST#
```

```
show run interface loopback1
```

```
Building configuration...
```

```
Current configuration : 66 bytes
```

```
!  
interface Loopback1  
 ip address 192.168.1.10 255.255.255.0  
end
```

```
CE-WEST#
```

```
show run interface loopback 1
```

```
Building configuration...
```

```
Current configuration : 65 bytes
```

```
!  
interface Loopback1  
 ip address 172.16.1.10 255.255.255.0  
end
```

A acessibilidade ICMP e um traceroute foram usados para começar a verificar a conectividade entre esses endereços de origem e destino, no entanto, a partir das próximas saídas, pode-se ver que isso não foi bem-sucedido:

```
<#root>
```

```
CE-EAST#
```

```
ping 172.16.1.10 source loopback1
```

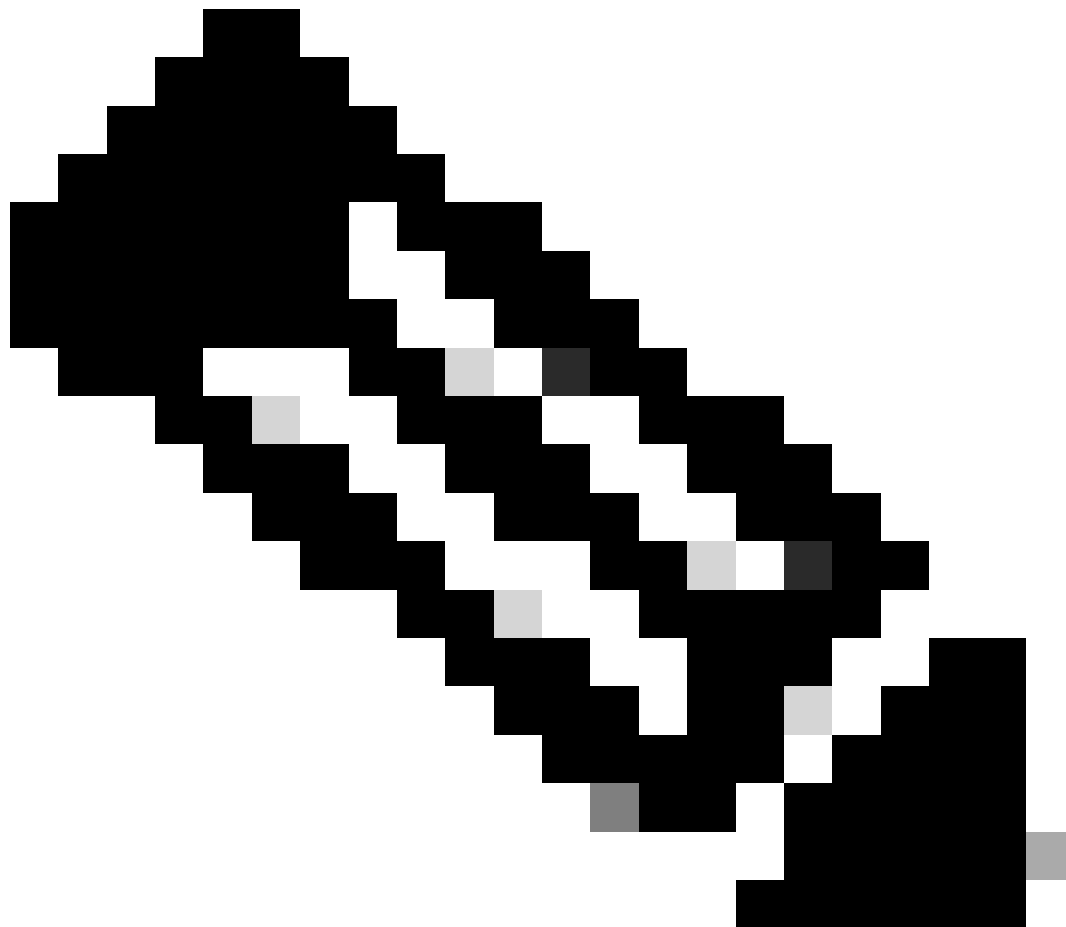
```
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 172.16.1.10, timeout is 2 seconds:  
Packet sent with a source address of 192.168.1.10  
.....  
Success rate is 0 percent (0/5)
```

```
CE-EAST#
```

```
traceroute 172.16.1.10 source loop1 probe 1 numeric
```

```
Type escape sequence to abort.  
Tracing the route to 172.16.1.10  
VRF info: (vrf in name/id, vrf out name/id)  
 1 10.11.0.2 2 msec  
 2 *  
 3 10.10.0.2 [MPLS: Label 16 Exp 0] 9 msec  
 4 *  
 5 *  
 6 *  
 7 *
```

8 \*  
9 \*  
10 \*  
11 \*  
12 \*  
13 \*  
14 \*  
15 \*  
16 \*  
17 \*  
18 \*  
19 \*  
20 \*  
21 \*  
22 \*  
23 \*  
24 \*  
25 \*  
26 \*  
27 \*  
28 \*  
29 \*  
30 \*  
CE-EAST#



Observação: durante a solução de problemas, o uso de um traceroute quando conectado a uma rede MPLS pode ser menos eficaz, pois alguns provedores de serviços tendem a configurar o comando no `mpls ip propagate-ttl forward` no Cisco IOS XE ou o comando `mpls ip-ttl-propagate disable forward` no Cisco IOS XR para ocultar todos os LSRs (Label Switch Router) no núcleo (isso, no entanto, com exceção dos roteadores PE de entrada e saída).

---

Ao revisar o status do roteador CE de origem, como este roteador não tem nenhum VRF (Virtual Route Forwarding) e não reconhece o MPLS, você precisa verificar o RIB (Routing Information Base), o CEF (Cisco Express Forwarding) e o BGP. Nas próximas saídas, pode-se observar que há uma entrada de roteamento conhecida via BGP para a sub-rede de destino 172.16.1.0/24 e que pode ser alcançada via interface GigabitEthernet0/0:

```
<#root>
```

```
CE-EAST#
```

```
show ip route 172.16.1.10
```

Routing entry for 172.16.1.0/24

Known via "bgp 65001", distance 20, metric 0

<<<<<

Tag 65500, type external  
Last update from 10.11.0.2 3d01h ago  
Routing Descriptor Blocks:  
\* 10.11.0.2, from 10.11.0.2, 3d01h ago  
Route metric is 0, traffic share count is 1  
AS Hops 2  
Route tag 65500  
MPLS label: none

CE-EAST#

show ip cef 172.16.1.10

172.16.1.0/24

nexthop 10.11.0.2 GigabitEthernet0/0

<<<<<

CE-EAST#

Como o roteador CE-EAST de origem tem uma rota para o destino instalada na RIB, é hora de examinar o roteador de borda do provedor PE4 (PE de ingresso), como mostrado na topologia. Neste ponto, o VRF e os diferenciadores de rota são configurados, bem como a importação e exportação de destino de rota, como visto nas próximas saídas:

<#root>

RP/0/0/CPU0:PE4#

show run vrf EAST

Mon Sep 11 20:01:54.454 UTC  
vrf EAST  
address-family ipv4 unicast

import route-target 65000:1 65001:1 65001:2 ! export route-target 65001:1

!  
!  
!

RP/0/0/CPU0:PE4#

show run router bgp

Mon Sep 11 20:06:48.164 UTC  
router bgp 65500  
address-family ipv4 unicast  
!  
address-family vpnv4 unicast  
!

```
neighbor 10.10.10.6
  remote-as 65500
  update-source Loopback0
  address-family vpnv4 unicast
  !
!
vrf EAST
```

```
rd 65001:1
```

```
  address-family ipv4 unicast
  !
  neighbor 10.11.0.1
    remote-as 65001
    address-family ipv4 unicast
      route-policy PASS in
      route-policy PASS out
    !
  !
!
!
```

```
RP/0/0/CPU0:PE4#
```

A partir da saída anterior, pode-se ver que o nome VRF "EAST" foi definido com a importação de rota de destino para 65000:1. Agora, a tabela de roteamento VRF pode ser verificada e isso ajuda a determinar se o PE4 tem uma rota para o endereço IP de destino 172.16.1.10:

```
<#root>
```

```
RP/0/0/CPU0:PE4#
```

```
show route vrf EAST 172.16.1.10
```

```
Mon Sep 11 19:58:28.128 UTC
```

```
Routing entry for 172.16.1.0/24
```

```
  Known via "bgp 65500", distance 200, metric 0
```

```
  Tag 65000, type internal
```

```
  Installed Sep  8 18:28:46.303 for 3d01h
```

```
  Routing Descriptor Blocks
```

```
    10.10.10.1, from 10.10.10.6
```

```
      Nexthop in Vrf: "default", Table: "default", IPv4 Unicast, Table Id: 0xe0000000
```

```
      Route metric is 0
```

```
  No advertising protos.
```

```
RP/0/0/CPU0:PE4#
```

Como esse PE é um dispositivo Cisco IOS XR, a palavra-chave "detail" pode ser usada no final do comando `show route vrf <name>` para examinar algumas informações adicionais, como o rótulo VPNv4 imposto pelo MP-BGP (Multiprotocol BGP) e o RD de origem (Route Distinguisher) do prefixo:

```
<#root>
```

RP/0/0/CPU0:PE4#

```
show route vrf EAST 172.16.1.10 detail
```

Mon Sep 11 20:21:48.492 UTC

Routing entry for 172.16.1.0/24

Known via "bgp 65500", distance 200, metric 0

Tag 65000, type internal

Installed Sep 8 18:28:46.303 for 3d01h

Routing Descriptor Blocks

10.10.10.1, from 10.10.10.6

Nexthop in Vrf: "default", Table: "default", IPv4 Unicast, Table Id: 0xe0000000

Route metric is 0

Label: 0x10 (16)

<<<<<

Tunnel ID: None

Binding Label: None

Extended communities count: 0

Source RD attributes: 0x0000:65000:1

<<<<<

NHID:0x0(Ref:0)

Route version is 0x5 (5)

No local label

IP Precedence: Not Set

QoS Group ID: Not Set

Flow-tag: Not Set

Fwd-class: Not Set

Route Priority: RIB\_PRIORITY\_RECURSIVE (12) SVD Type RIB\_SVD\_TYPE\_REMOTE

Download Priority 3, Download Version 36

No advertising protos.

RP/0/0/CPU0:PE4#

Agora, vamos dar uma olhada no prefixo VPNv4 BGP que foi importado para o VRF, observe que este é o mesmo rótulo 16 da saída anterior e também tem a comunidade estendida 65000:1.

Também é importante observar que 10.10.10.1 é o próximo salto que o PE4 precisa para poder executar uma recursão de rota para ele, o próximo endereço "de 10.10.10.6" é o par BGP que o PE4 usou para aprender esse prefixo (nesse cenário, é o Refletor de Rota P6):

<#root>

RP/0/0/CPU0:PE4#

```
show bgp vpnv4 unicast vrf EAST 172.16.1.10
```

Mon Sep 11 22:42:28.114 UTC

BGP routing table entry for 172.16.1.0/24, Route Distinguisher: 65001:1

Versions:

Process	bRIB/RIB	SendTblVer
---------	----------	------------

Speaker	48	48
---------	----	----

Last Modified: Sep 8 18:28:46.314 for 3d04h

Paths: (1 available, best #1)



```
Not advertised to any peer
Path #1: Received by speaker 0
Not advertised to any peer
65000
```

```
10.10.10.1 (metric 20) from 10.10.10.6 (10.10.10.1)
```

```
<<<<<
```

```
Received Label 16
```

```
Origin IGP, metric 0, localpref 100, valid, internal, best, group-best, import-candidate, imported
Received Path ID 0, Local Path ID 0, version 48
```

```
Extended community: RT:65000:1
```

```
<<<<<
```

```
Originator: 10.10.10.1, Cluster list: 10.10.10.6
```

```
Source AFI: VPNv4 Unicast, Source VRF: default, Source Route Distinguisher: 65000:1
```

```
<<<<<
```

Ao revisar o CEF com a palavra-chave de rota exata no nível do VRF, você pode ter uma ideia da interface de saída dos pacotes. Esse comando também pode fornecer alguns detalhes importantes, pois mostra os dois rótulos impostos ao prefixo, 24001 e 16, o motivo é que o rótulo 16 está vindo do BGP VPNv4 e o rótulo 24001 está vindo do LDP (Label Distribution Protocol):

```
<#root>
```

```
RP/0/0/CPU0:PE4#
```

```
show cef vrf EAST exact-route 192.168.1.10 172.16.1.10
```

```
Mon Sep 11 22:48:15.241 UTC
```

```
172.16.1.0/24, version 36, internal 0x5000001 0x0 (ptr 0xa12dc74c) [1], 0x0 (0x0), 0x208 (0xa155b1b8)
```

```
Updated Sep 8 18:28:46.323
```

```
local adjacency 10.0.0.16
```

```
Prefix Len 24, traffic index 0, precedence n/a, priority 3
```

```
via GigabitEthernet0/0/0/4
```

```
via 10.10.10.1/32, 3 dependencies, recursive [flags 0x6000]
```

```
path-idx 0 NHID 0x0 [0xa15c3f54 0x0]
```

```
recursion-via-/32
```

```
next hop VRF - 'default', table - 0xe0000000
```

```
next hop 10.10.10.1/32 via 24010/0/21
```

```
next hop 10.0.0.16/32 Gi0/0/0/4 labels imposed {24001 16}
```

```
<<<<<
```

Como próxima etapa, o comando `show bgp vpnv4 unicast` é usado para verificar as rotas VPNv4

que estão sendo aprendidas por esse PE. Esta saída mostra as informações antes do prefixo VPNv4 ser importado para o VRF, lembre-se de que o RT (Route Target) configurado (para este exemplo, os RTs importados são 65000:1, 65001:1, 65001:2) indica quais rotas e para quais VRF são importadas:

<#root>

RP/0/0/CPU0:PE4#

show bgp vpnv4 unicast

```
Fri Sep 15 02:15:15.463 UTC
BGP router identifier 10.10.10.4, local AS number 65500
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0 RD version: 0
BGP main routing table version 85
BGP NSR Initial initsync version 1 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs
```

Status codes: s suppressed, d damped, h history, \* valid, > best  
i - internal, r RIB-failure, S stale, N Nexthop-discard

Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
---------	----------	--------	--------	--------	------

Route Distinguisher: 65000:1

```
*>i172.16.1.0/24 10.10.10.1 0 100 0 65000 i
```

<<<<<

```
*>i172.16.2.0/24 10.10.10.1 0 100 0 65000 i
```

Route Distinguisher: 65001:1 (default for vrf EAST)

```
* i0.0.0.0/0 10.10.10.3 0 100 0 65001 i
```

```
*> 10.11.0.1 0 65001 i
```

```
*>i172.16.1.0/24 10.10.10.1 0 100 0 65000 i
```

```
*>i172.16.2.0/24 10.10.10.1 0 100 0 65000 i
```

```
*> 192.168.1.0/24 10.11.0.1 0 65001 i
```

```
*>i192.168.2.0/24 10.10.10.3 0 100 0 65001 i
```

```
*> 192.168.3.0/24 10.11.0.1 0 65001 i
```

Route Distinguisher: 65001:2

```
*>i0.0.0.0/0 10.10.10.3 0 100 0 65001 i
```

```
*>i192.168.2.0/24 10.10.10.3 0 100 0 65001 i
```

Processed 10 prefixes, 11 paths

Neste exemplo, a tabela VPNv4 pode ser pequena, mas em um ambiente de produção, em vez de examinar todos os prefixos VPNv4, você pode restringir a verificação para o RD específico e o prefixo com o próximo comando:

<#root>

RP/0/0/CPU0:PE4#

```
show bgp vpnv4 unicast rd 65000:1 172.16.1.10
```

```
Mon Sep 11 22:54:04.967 UTC
```

```
BGP routing table entry for 172.16.1.0/24, Route Distinguisher: 65000:1
```

```
Versions:
```

```
Process          bRIB/RIB  SendTblVer
Speaker          46        46
```

```
Last Modified: Sep  8 18:28:46.314 for 3d04h
```

```
Paths: (1 available, best #1)
```

```
Not advertised to any peer
Path #1: Received by speaker 0
Not advertised to any peer
```

```
65000
```

```
10.10.10.1 (metric 20) from 10.10.10.6 (10.10.10.1)
```

```
Received Label 16
```

```
Origin IGP, metric 0, localpref 100, valid, internal, best, group-best, import-candidate, not-in-
```

```
Received Path ID 0, Local Path ID 0, version 46
```

```
Extended community: RT:65000:1
```

```
Originator: 10.10.10.1, Cluster list: 10.10.10.6
```

Neste ponto, o plano de controle do MP-BGP tem o prefixo de destino e os rótulos LDP e VPNv4 {24001 16} respectivamente, a interface de saída para esse tráfego parece ser Gi0/0/0/4 e o próximo salto onde o tráfego precisa ser encaminhado é 10.10.10.1. Mas, há outra opção para verificar a interface de saída preferida? É hora de examinar a tabela de encaminhamento MPLS ou LFIB (Label Forwarding Information Base, Base de informações de encaminhamento de rótulo). Com o uso do comando show mpls forwarding, duas entradas são exibidas em direção ao destino 10.10.10.1 (Loopback0 de PE1), um caminho com uma Interface de Saída de Gi0/0/0/4 e um próximo salto 10.0.0.16 (roteador P5) onde o Rótulo de Saída imposto é 24001 e outro caminho através de Gi0/0/0/3 com um próximo salto 10.0.0.13 (roteador P6) e um Rótulo de Saída de 23:

```
<#root>
```

```
RP/0/0/CPU0:PE4#
```

```
show mpls forwarding
```

```
Mon Sep 11 23:28:33.425 UTC
```

Local Label	Outgoing Label	Prefix or ID	Outgoing Interface	Next Hop	Bytes Switched
24000	Unlabelled	192.168.1.0/24[V]	Gi0/0/0/0	10.11.0.1	1096
24001	Unlabelled	192.168.3.0/24[V]	Gi0/0/0/0	10.11.0.1	56056
24002	Unlabelled	0.0.0.0/0[V]	Gi0/0/0/0	10.11.0.1	0
24003	Pop	10.10.10.6/32	Gi0/0/0/3	10.0.0.13	7778512
24004	Pop	10.0.0.4/31	Gi0/0/0/3	10.0.0.13	0
24005	Pop	10.0.0.8/31	Gi0/0/0/3	10.0.0.13	0
24006	Pop	10.10.10.5/32	Gi0/0/0/4	10.0.0.16	3542574
24007	Pop	10.0.0.10/31	Gi0/0/0/3	10.0.0.13	0
	Pop	10.0.0.10/31	Gi0/0/0/4	10.0.0.16	0
24008	Pop	10.0.0.6/31	Gi0/0/0/4	10.0.0.16	0
24009	Pop	10.0.0.0/31	Gi0/0/0/4	10.0.0.16	0
24010	23	10.10.10.1/32	Gi0/0/0/3	10.0.0.13	22316

```
<<<<<
```

```
24001 10.10.10.1/32 Gi0/0/0/4 10.0.0.16 42308
```

```
<<<<<
```

```
24011 18      10.10.10.2/32    Gi0/0/0/3    10.0.0.13    0
      24003      10.10.10.2/32    Gi0/0/0/4    10.0.0.16    0
24012 17      10.0.0.2/31     Gi0/0/0/3    10.0.0.13    0
      24005      10.0.0.2/31     Gi0/0/0/4    10.0.0.16    0
24013 Pop      10.10.10.3/32    Gi0/0/0/1    10.0.0.20    3553900
24014 Pop      10.0.0.14/31    Gi0/0/0/1    10.0.0.20    0
      Pop      10.0.0.14/31    Gi0/0/0/4    10.0.0.16    0
24015 Pop      10.0.0.18/31    Gi0/0/0/1    10.0.0.20    0
      Pop      10.0.0.18/31    Gi0/0/0/3    10.0.0.13    0
```

```
RP/0/0/CPU0:PE4#
```

```
show mpls forwarding prefix 10.10.10.1/32
```

```
Mon Sep 11 23:30:54.685 UTC
```

Local Label	Outgoing Label	Prefix or ID	Outgoing Interface	Next Hop	Bytes Switched
24010	23	10.10.10.1/32	Gi0/0/0/3	10.0.0.13	3188
	24001	10.10.10.1/32	Gi0/0/0/4	10.0.0.16	6044

```
RP/0/0/CPU0:PE4#
```

```
show mpls forwarding prefix 10.10.10.1/32 detail hardware egress
```

```
Mon Sep 11 23:36:06.504 UTC
```

Local Label	Outgoing Label	Prefix or ID	Outgoing Interface	Next Hop	Bytes Switched
24010	23	10.10.10.1/32	Gi0/0/0/3	10.0.0.13	N/A
Updated: Sep 8 20:27:26.596 Version: 39, Priority: 3 Label Stack (Top -> Bottom): { 23 } NHID: 0x0, Encap-ID: N/A, Path idx: 0, Backup path idx: 0, Weight: 0 MAC/Encaps: 14/18, MTU: 1500 Outgoing Interface: GigabitEthernet0/0/0/3 (ifhandle 0x000000a0) Packets Switched: 0					
	24001	10.10.10.1/32	Gi0/0/0/4	10.0.0.16	N/A
Updated: Sep 8 20:27:26.596 Version: 39, Priority: 3 Label Stack (Top -> Bottom): { 24001 } NHID: 0x0, Encap-ID: N/A, Path idx: 1, Backup path idx: 0, Weight: 0 MAC/Encaps: 14/18, MTU: 1500 Outgoing Interface: GigabitEthernet0/0/0/4 (ifhandle 0x000000c0) Packets Switched: 0					

A partir das saídas anteriores, fica claro que há duas opções de caminho em que o tráfego pode ter a carga balanceada, mas há algumas maneiras que podem ajudar a determinar qual é o caminho preferido. Uma maneira é com o uso do comando `show cef exact-route <source IP> <destination IP>` adicionando o Loopback0 do PE de Origem e o Loopback0 do PE de Destino. Como mostrado na próxima saída, o caminho preferido é através de Gi0/0/0/4:

<#root>

RP/0/0/CPU0:PE4#

show cef exact-route 10.10.10.4 10.10.10.1

Mon Sep 11 23:49:44.558 UTC

10.10.10.1/32, version 39, internal 0x1000001 0x0 (ptr 0xa12dbdbc) [1], 0x0 (0xa12c18c0), 0xa28 (0xa185

Updated Sep 8 20:27:26.596

Local adjacency 10.0.0.16

Prefix Len 32, traffic index 0, precedence n/a, priority 3

via GigabitEthernet0/0/0/4

via 10.0.0.16/32, GigabitEthernet0/0/0/4, 9 dependencies, weight 0, class 0 [flags 0x0]

<<<<<

path-idx 1 NHID 0x0 [0xa16765bc 0x0]

next hop 10.0.0.16/32

local adjacency

local label 24010 labels imposed {24001}

Outra opção é primeiro verificar o LIB (Label Information Base) e obter a vinculação LDP do destino Loopback0 (10.10.10.1 que pertence ao PE de saída) usando o comando show mpls ldp bindings <prefix/mask>, e uma vez que o rótulo de vinculação local seja encontrado a partir dessa saída, use esse valor de rótulo no comando show mpls forwarding exact-route label <label> ipv4 <source IP> <destination IP> detail para encontrar o caminho preferido:

<#root>

RP/0/0/CPU0:PE4#

show mpls ldp bindings 10.10.10.1/32

Wed Sep 13 17:18:43.007 UTC

10.10.10.1/32, rev 29

Local binding: label: 24010

<<<<<

Remote bindings: (3 peers)

Peer	Label
10.10.10.3:0	24
10.10.10.5:0	24001
10.10.10.6:0	23

RP/0/0/CPU0:PE4#

show mpls forwarding exact-route label 24010 ipv4 10.10.10.4 10.10.10.1 detail

Wed Sep 13 17:20:06.342 UTC

Local Label	Outgoing Label	Prefix or ID	Outgoing Interface	Next Hop	Bytes Switched
24010	24001	10.10.10.1/32	Gi0/0/0/4	10.0.0.16	N/A

<<<<<

```

Updated: Sep 12 14:15:37.009
Version: 198, Priority: 3
Label Stack (Top -> Bottom): { 24001 }
NHID: 0x0, Encap-ID: N/A, Path idx: 1, Backup path idx: 0, Weight: 0
Hash idx: 1
MAC/Encaps: 14/18, MTU: 1500
Outgoing Interface: GigabitEthernet0/0/0/4 (ifhandle 0x000000c0)
Packets Switched: 0

```

```

Via: Gi0/0/0/4, Next Hop: 10.0.0.16
Label Stack (Top -> Bottom): { 24001 }
NHID: 0x0, Encap-ID: N/A, Path idx: 1, Backup path idx: 0, Weight: 0
Hash idx: 1
MAC/Encaps: 14/18, MTU: 1500
Outgoing Interface: GigabitEthernet0/0/0/4 (ifhandle 0x000000c0)

```

Em seguida, é importante verificar o roteador do próximo salto que está no dataplane, para este exemplo específico, o roteador a verificar é P5 (que tem a interface 10.0.0.16). O primeiro lugar para começar a examinar é a tabela de encaminhamento MPLS, onde o rótulo local para o prefixo 10.10.10.1 deve ser 24001:

```
<#root>
```

```
RP/0/0/CPU0:P5#
```

```
show mpls forwarding
```

```
Thu Sep 14 20:07:16.455 UTC
```

Local Label	Outgoing Label	Prefix or ID	Outgoing Interface	Next Hop	Bytes Switched
24000	Pop	10.10.10.6/32	Gi0/0/0/2	10.0.0.11	361906
24001	Pop	10.10.10.1/32	Gi0/0/0/1	10.0.0.0	361002
		<<<<<			
24002	Pop	10.0.0.4/31	Gi0/0/0/1	10.0.0.0	0
	Pop	10.0.0.4/31	Gi0/0/0/2	10.0.0.11	0
24003	Pop	10.10.10.2/32	Gi0/0/0/0	10.0.0.6	360940
24004	Pop	10.0.0.8/31	Gi0/0/0/0	10.0.0.6	0
	Pop	10.0.0.8/31	Gi0/0/0/2	10.0.0.11	0
24005	Pop	10.0.0.2/31	Gi0/0/0/0	10.0.0.6	0
	Pop	10.0.0.2/31	Gi0/0/0/1	10.0.0.0	0
24006	Pop	10.10.10.4/32	Gi0/0/0/4	10.0.0.17	361230
24007	Pop	10.0.0.12/31	Gi0/0/0/2	10.0.0.11	0
	Pop	10.0.0.12/31	Gi0/0/0/4	10.0.0.17	0
24008	Pop	10.10.10.3/32	Gi0/0/0/3	10.0.0.15	361346
24009	Pop	10.0.0.20/31	Gi0/0/0/3	10.0.0.15	0
	Pop	10.0.0.20/31	Gi0/0/0/4	10.0.0.17	0
24010	Pop	10.0.0.18/31	Gi0/0/0/2	10.0.0.11	0
	Pop	10.0.0.18/31	Gi0/0/0/3	10.0.0.15	0

```
RP/0/0/CPU0:P5#
```

```
show mpls forwarding labels 24001
```

```
Thu Sep 14 20:07:42.584 UTC
```

Local Label	Outgoing Label	Prefix or ID	Outgoing Interface	Next Hop	Bytes Switched
24001	Pop	10.10.10.1/32	Gi0/0/0/1	10.0.0.0	361060

RP/0/0/CPU0:P5#

A partir da saída anterior, pode-se ver que a entrada LFIB para o prefixo 10.10.10.1/32 mostra "Pop" como o rótulo de saída, significando que este roteador é o Penultimate Hop Popping (PHP). Ele também mostra que o tráfego deve ser enviado por meio de Gi0/0/0/1 com base nas informações de LFIB, e isso também pode ser verificado ao examinar o CEF. A próxima saída de rota exata CEF mostra o Rótulo Nulo Implícito como o rótulo imposto, isso novamente, é devido ao fato de que o próximo salto conectado em Gi0/0/0/1 é o último roteador no caminho do switch de rótulo e também é o PE voltado para o local de destino (CE-WEST). Essa também é a razão pela qual o roteador P5 está removendo e não impondo outro rótulo ao pacote, graças a esse processo, o roteador de saída PE1 receberá um pacote sem um rótulo LDP:

<#root>

RP/0/0/CPU0:P5#

**show cef exact-route 10.10.10.4 10.10.10.1**

Thu Sep 14 20:25:57.269 UTC

10.10.10.1/32, version 192, internal 0x1000001 0x0 (ptr 0xa1246394) [1], 0x0 (0xa122b638), 0xa20 (0xa15

Updated Sep 12 14:15:38.009

local adjacency 10.0.0.0

Prefix Len 32, traffic index 0, precedence n/a, priority 3

via GigabitEthernet0/0/0/1

via 10.0.0.0/32, GigabitEthernet0/0/0/1, 9 dependencies, weight 0, class 0 [flags 0x0]

path-idx 0 NHID 0x0 [0xa166e280 0xa166e674]

next hop 10.0.0.0/32

local adjacency

local label 24001 labels imposed {ImplNull}

<<<<<

O último ponto para verificar o caminho do switch de rótulo é PE1. Ao examinar a tabela de encaminhamento MPLS, observe que não há entrada para o prefixo 10.10.10.1/32 no LFIB:

<#root>

PE1#

**show mpls forwarding-table**

Local Label	Outgoing Label	Prefix or Tunnel Id	Bytes Switched	Outgoing interface	Next Hop
-------------	----------------	---------------------	----------------	--------------------	----------

16	No Label	172.16.1.0/24[V]	12938	Gi3	10.10.0.1
17	No Label	172.16.2.0/24[V]	0	Gi3	10.10.0.1
18	Pop Label	10.0.0.6/31	0	Gi1	10.0.0.1
	Pop Label	10.0.0.6/31	0	Gi2	10.0.0.3
19	Pop Label	10.0.0.8/31	0	Gi2	10.0.0.3
	Pop Label	10.0.0.8/31	0	Gi4	10.0.0.5
20	Pop Label	10.0.0.10/31	0	Gi1	10.0.0.1
	Pop Label	10.0.0.10/31	0	Gi4	10.0.0.5
21	Pop Label	10.0.0.12/31	0	Gi4	10.0.0.5
22	Pop Label	10.0.0.14/31	0	Gi1	10.0.0.1
23	Pop Label	10.0.0.16/31	0	Gi1	10.0.0.1
24	Pop Label	10.0.0.18/31	0	Gi4	10.0.0.5
25	24009	10.0.0.20/31	0	Gi1	10.0.0.1
	22	10.0.0.20/31	0	Gi4	10.0.0.5
26	Pop Label	10.10.10.2/32	0	Gi2	10.0.0.3
27	24008	10.10.10.3/32	0	Gi1	10.0.0.1
	24	10.10.10.3/32	0	Gi4	10.0.0.5
28	24006	10.10.10.4/32	0	Gi1	10.0.0.1
	25	10.10.10.4/32	0	Gi4	10.0.0.5
29	Pop Label	10.10.10.5/32	0	Gi1	10.0.0.1
Local	Outgoing	Prefix	Bytes Label	Outgoing	Next Hop
Label	Label	or Tunnel Id	Switched	interface	
30	Pop Label	10.10.10.6/32	0	Gi4	10.0.0.5
31	[T] Pop Label	1/1[TE-Bind]	0	drop	

[T] Forwarding through a LSP tunnel.  
View additional labelling info with the 'detail' option

Como você descobriu, a razão desse comportamento é que o prefixo (10.10.10.1/32) pertence a PE1 e o roteador também atribuiu um rótulo nulo implícito a esse prefixo conectado. Isso pode ser verificado com o uso do comando `show mpls ldp bindings`:

```
<#root>
```

```
PE1#
```

```
show run interface loopback 0
```

```
Building configuration...
```

```
Current configuration : 66 bytes
```

```
!
```

```
interface Loopback0
```

```
 ip address 10.10.10.1 255.255.255.255
```

```
end
```

```
PE1#
```

```
show mpls ldp bindings 10.10.10.1 32
```

```
lib entry: 10.10.10.1/32, rev 24
```

```
local binding: label: imp-null
```

```
remote binding: lsr: 10.10.10.6:0, label: 23
```

```
remote binding: lsr: 10.10.10.5:0, label: 24001
```

```
remote binding: lsr: 10.10.10.2:0, label: 24000
```



Como PE1 é um roteador Cisco IOS XE, o uso do comando show bgp vpnv4 unicast all ou show bgp vpnv4 unicast rd <value> <destination IP> pode ajudar a identificar e confirmar que o prefixo de destino 172.16.1.0/24 está sendo aprendido corretamente através do MP-BGP. A saída desses comandos mostra o prefixo após a exportação:

<#root>

PE1#

show bgp vpnv4 unicast all

BGP table version is 61, local router ID is 10.10.10.1  
Status codes: s suppressed, d damped, h history, \* valid, > best, i - internal,  
r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,  
x best-external, a additional-path, c RIB-compressed,  
t secondary path, L long-lived-stale,  
Origin codes: i - IGP, e - EGP, ? - incomplete  
RPKI validation codes: V valid, I invalid, N Not found

Network	Next Hop	Metric	LocPrf	Weight	Path
Route Distinguisher: 65000:1 (default for vrf WEST)					
*>i 0.0.0.0	10.10.10.3	0	100	0	65001 i
*bi	10.10.10.4	0	100	0	65001 i
*> 172.16.1.0/24 10.10.0.1 0 0 65000 i					
<<<<<<					
*> 172.16.2.0/24	10.10.0.1	0		0	65000 i
*>i 192.168.1.0	10.10.10.4	0	100	0	65001 i
*>i 192.168.2.0	10.10.10.3	0	100	0	65001 i
*>i 192.168.3.0	10.10.10.4	0	100	0	65001 i
Route Distinguisher: 65001:1					
*>i 0.0.0.0	10.10.10.4	0	100	0	65001 i
*>i 192.168.1.0	10.10.10.4	0	100	0	65001 i
*>i 192.168.3.0	10.10.10.4	0	100	0	65001 i
Route Distinguisher: 65001:2					
Network	Next Hop	Metric	LocPrf	Weight	Path
*>i 0.0.0.0	10.10.10.3	0	100	0	65001 i
*>i 192.168.2.0	10.10.10.3	0	100	0	65001 i

PE1#

show bgp vpnv4 unicast rd 65000:1 172.16.1.10

BGP routing table entry for 65000:1:172.16.1.0/24, version 2  
Paths: (1 available, best #1, table WEST)  
Additional-path-install  
Advertised to update-groups:  
6  
Refresh Epoch 2  
65000

10.10.0.1 (via vrf WEST) from 10.10.0.1 (172.16.2.10)

<<<<<<  
Origin IGP, metric 0, localpref 100, valid, external, best

Extended Community: RT:65000:1 , recursive-via-connected

<<<<<<

```
mpls labels in/out 16/nolabel
rx pathid: 0, tx pathid: 0x0
Updated on Sep 15 2023 18:27:23 UTC
```

De forma semelhante, observando o prefixo VPNv4 BGP no VRF, que é o prefixo recebido pelo CE-WEST, com o uso do comando `show bgp vpnv4 unicast vrf <nome> <prefixo>`, a saída mostra o rótulo 16 MP-BGP que foi transportado por todo o caminho para o PE4 de entrada, bem como a exportação RT configurada 65000:1:

```
<#root>
```

```
PE1#
```

```
show bgp vpnv4 unicast vrf WEST 172.16.1.10
```

```
BGP routing table entry for 65000:1:172.16.1.0/24, version 2
Paths: (1 available, best #1, table WEST)
  Additional-path-install
  Advertised to update-groups:
    6
  Refresh Epoch 2
  65000
  10.10.0.1 (via vrf WEST) from 10.10.0.1 (172.16.2.10)
  Origin IGP, metric 0, localpref 100, valid, external, best
```

```
Extended Community: RT:65000:1 , recursive-via-connected
```

```
<<<<<
```

```
mpls labels in/out 16/nolabel
```

```
<<<<<
rx pathid: 0, tx pathid: 0x0
Updated on Sep 15 2023 18:27:23 UTC
```

```
PE1#
```

```
show run vrf WEST
```

```
Building configuration...
```

```
Current configuration : 478 bytes
vrf definition WEST
  rd 65000:1
```

```
route-target export 65000:1
```

```
<<<<<
route-target import 65000:1
route-target import 65001:1
route-target import 65001:2
!
address-family ipv4
exit-address-family
!
!
interface GigabitEthernet3
```

```
vrf forwarding WEST
ip address 10.10.0.2 255.255.255.252
negotiation auto
no mop enabled
no mop sysid
!
router bgp 65500
!
address-family ipv4 vrf WEST
 neighbor 10.10.0.1 remote-as 65000
 neighbor 10.10.0.1 activate
exit-address-family
!
end
```

As últimas informações a serem verificadas nesse PE são as entradas RIB e CEF no nível de VRF para o IP de destino, ao contrário da entrada vista no PE4, não há nenhum Rótulo no RIB para o prefixo 172.16.1.0/24, o motivo é que essa é a rota de entrada do CE e isso é aprendido por eBGP e inserido na tabela de roteamento VRF antes que esse prefixo seja exportado para o VPNv4. Isso pode ser verificado com o uso dos comandos `show ip route vrf <name> <prefix>` e `show ip cef vrf <name> <prefix>` mostrados a seguir:

<#root>

PE1#

```
show ip route vrf WEST 172.16.1.10
```

Routing Table: WEST

Routing entry for 172.16.1.0/24

Known via "bgp 65500", distance 20, metric 0

Tag 65000, type external

Last update from 10.10.0.1 1w0d ago

Routing Descriptor Blocks:

\* 10.10.0.1, from 10.10.0.1, 1w0d ago, recursive-via-conn

opaque\_ptr 0x7F8B4E3E1D50

Route metric is 0, traffic share count is 1

AS Hops 1

Route tag 65000

MPLS label: none

PE1#

```
show ip cef vrf WEST 172.16.1.10
```

172.16.1.0/24

nexthop 10.10.0.1 GigabitEthernet3

Neste ponto, foi confirmado que o prefixo de destino 172.16.1.0/24 foi aprendido corretamente pela origem do tráfego CE (CE-EAST), foi propagado corretamente através de MP-BGP e também os rótulos de PEs e loopbacks Ps foram aprendidos através do caminho do switch de rótulo. Mas mesmo assim, a acessibilidade entre origem/destino não é bem-sucedida e ainda há

um último roteador para verificar o CE-WEST. A primeira coisa a verificar neste roteador é a tabela de roteamento. Lembre-se de que o prefixo IP origem 192.168.1.0/24 deve aparecer ali:

```
<#root>
CE-WEST#
show ip route 192.168.1.10
% Network not in table
CE-WEST#
```

A "Rede não na tabela" é claramente o problema, a tabela BGP também pode ser verificada, mas depois de procurar o prefixo, ela também não está lá:

```
<#root>
CE-WEST#
show ip bgp
BGP table version is 41, local router ID is 172.16.2.10
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
               t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop          Metric LocPrf Weight Path
*>  172.16.1.0/24    0.0.0.0           0         32768 i
*>  172.16.2.0/24    0.0.0.0           0         32768 i
CE-WEST#
```

Indo um passo atrás, você pode verificar se este roteador de borda do provedor (PE1) está anunciando o prefixo para o vizinho eBGP CE-WEST, isso pode ser feito com o uso do comando `show bgp vpnv4 unicast vrf <nome> neighbors <IP vizinho> advertised-routes` mostrado a seguir:

```
<#root>
PE1#
show bgp vpnv4 unicast vrf WEST neighbors 10.10.0.1 advertised-routes

BGP table version is 61, local router ID is 10.10.10.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
               t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

```

Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 65000:1 (default for vrf WEST)
*>i 0.0.0.0        10.10.10.3        0    100    0 65001 i

*>i 192.168.1.0 10.10.10.4 0 100 0 65001 i

      <<<<<<
*>i 192.168.2.0    10.10.10.3        0    100    0 65001 i
*>i 192.168.3.0    10.10.10.4        0    100    0 65001 i

```

Total number of prefixes 4

Com base na etapa anterior, pode-se confirmar que o roteador PE1 está anunciando o prefixo corretamente ao CE-WEST, portanto, é hora de examinar os vizinhos BGP no lado CE:

<#root>

CE-WEST#

show ip bgp neighbors

```

BGP neighbor is 10.10.0.2, remote AS 65500, external link
  BGP version 4, remote router ID 10.10.10.1
  BGP state = Established, up for 1w4d
  Last read 00:00:40, last write 00:00:43, hold time is 180, keepalive interval is 60 seconds
  Neighbor sessions:
    1 active, is not multiseession capable (disabled)
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Four-octets ASN Capability: advertised and received
    Address family IPv4 Unicast: advertised and received
    Enhanced Refresh Capability: advertised and received
    Multiseession Capability:
      Stateful switchover support enabled: NO for session 1
  Message statistics:
    InQ depth is 0
    OutQ depth is 0

      Sent      Rcvd
  Opens:          1         1
  Notifications:  0         0
  Updates:        3        17
  Keepalives:    19021    18997
  Route Refresh:  2         0
  Total:         19029    19019
  Do log neighbor state changes (via global configuration)
  Default minimum time between advertisement runs is 30 seconds

```

For address family: IPv4 Unicast

```

Session: 10.10.0.2
BGP table version 41, neighbor version 41/0
Output queue size : 0
Index 3, Advertise bit 0
3 update-group member
Inbound path policy configured

```

Route map for incoming advertisements is FILTER

<<<<<

Slow-peer detection is disabled  
Slow-peer split-update-group dynamic is disabled

	Sent	Rcvd
Prefix activity:	----	----
Prefixes Current:	2	0
Prefixes Total:	4	23
Implicit Withdraw:	2	13
Explicit Withdraw:	0	10
Used as bestpath:	n/a	0
Used as multipath:	n/a	0
Used as secondary:	n/a	0

	Outbound	Inbound
Local Policy Denied Prefixes:	-----	-----
route-map:	0	4
Bestpath from this peer:	18	n/a
Total:	18	4

Number of NLRI in the update sent: max 2, min 0  
Last detected as dynamic slow peer: never  
Dynamic slow peer recovered: never  
Refresh Epoch: 3  
Last Sent Refresh Start-of-rib: 4d23h  
Last Sent Refresh End-of-rib: 4d23h  
Refresh-Out took 0 seconds  
Last Received Refresh Start-of-rib: 4d23h  
Last Received Refresh End-of-rib: 4d23h  
Refresh-In took 0 seconds

	Sent	Rcvd
Refresh activity:	----	----
Refresh Start-of-RIB	1	2
Refresh End-of-RIB	1	2

Address tracking is enabled, the RIB does have a route to 10.10.0.2  
Route to peer address reachability Up: 1; Down: 0  
Last notification 1w5d  
Connections established 3; dropped 2  
Last reset 1w4d, due to Peer closed the session of session 1  
External BGP neighbor configured for connected checks (single-hop no-disable-connected-check)  
Interface associated: GigabitEthernet0/3 (peering address in same link)  
Transport(tcp) path-mtu-discovery is enabled  
Graceful-Restart is disabled  
SSO is disabled

Connection state is ESTAB, I/O status: 1, unread input bytes: 0  
Connection is ECN Disabled, Minimum incoming TTL 0, Outgoing TTL 1  
Local host: 10.10.0.1, Local port: 179  
Foreign host: 10.10.0.2, Foreign port: 39410  
Connection tableid (VRF): 0  
Maximum output segment queue size: 50

Enqueued packets for retransmit: 0, input: 0 mis-ordered: 0 (0 bytes)

Event Timers (current time is 0x4D15FD56):

Timer	Starts	Wakeups	Next
Retrans	19027	1	0x0
TimeWait	0	0	0x0
AckHold	19012	18693	0x0
SendWnd	0	0	0x0
KeepAlive	0	0	0x0
GiveUp	0	0	0x0
PmtuAger	0	0	0x0

```
DeadWait      0      0      0x0
Linger        0      0      0x0
ProcessQ      0      0      0x0
```

```
iss: 1676751051  snduna: 1677112739  sndnxt: 1677112739
irs: 2109012892  rcvnxt: 2109374776
```

```
sndwnd: 16061  scale:      0  maxrcvwnd: 16384
rcvwnd: 15890  scale:      0  delrcvwnd:  494
```

```
SRTT: 1000 ms, RTT0: 1003 ms, RTV: 3 ms, KRTT: 0 ms
minRTT: 0 ms, maxRTT: 1000 ms, ACK hold: 200 ms
uptime: 1036662542 ms, Sent idletime: 40725 ms, Receive idletime: 40925 ms
Status Flags: passive open, gen tcbs
Option Flags: nagle, path mtu capable
IP Precedence value : 6
```

Datagrams (max data segment is 1460 bytes):

Rcvd: 37957 (out of order: 0), with data: 19014, total data bytes: 361883

Sent: 37971 (retransmit: 1, fastretransmit: 0, partialack: 0, Second Congestion: 0), with data: 19027,

Packets received in fast path: 0, fast processed: 0, slow path: 0

fast lock acquisition failures: 0, slow path: 0

TCP Semaphore 0x0F3194AC FREE

A saída anterior revela que há um mapa de rota aplicado para anúncios de entrada com o nome "FILTER", depois de olhar para a configuração do mapa de rota, ele mostra uma cláusula de correspondência apontando para uma lista de prefixos com uma instrução permit para 192.168.0.0/16, no entanto, isso está incorreto, pois a lista de prefixos só permite esse prefixo específico e não todos os que podem ser incluídos nesse intervalo:

```
<#root>
```

```
CE-WEST#
```

```
show route-map FILTER
```

```
route-map FILTER, permit, sequence 10
```

```
Match clauses:
```

```
ip address prefix-lists: FILTER
```

```
Set clauses:
```

```
Policy routing matches: 0 packets, 0 bytes
```

```
CE-WEST#
```

```
show ip prefix-list FILTER
```

```
ip prefix-list FILTER: 1 entries
```

```
seq 5 permit 192.168.0.0/16
```

```
<<<<<
```

```
CE-WEST#
```

```
show run | i ip prefix-list
```

```
ip prefix-list FILTER seq 5 permit 192.168.0.0/16
```

Com uma pequena alteração na configuração da lista de prefixos, a rota em direção a 192.168.1.10 agora está instalada no RIB:

```
<#root>
```

```
CE-WEST#
```

```
show run | i ip prefix-list
```

```
ip prefix-list FILTER seq 5 permit 192.168.0.0/16 le 32
```

```
<<<<<
```

```
CE-WEST#
```

```
show ip bgp
```

```
BGP table version is 44, local router ID is 172.16.2.10
```

```
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,  
r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,  
x best-external, a additional-path, c RIB-compressed,  
t secondary path,
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete
```

```
RPKI validation codes: V valid, I invalid, N Not found
```

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	172.16.1.0/24	0.0.0.0	0		32768	i
*>	172.16.2.0/24	0.0.0.0	0		32768	i

```
*> 192.168.1.0 10.10.0.2 0 65500 65001 i
```

```
<<<<<
```

*>	192.168.2.0	10.10.0.2			0 65500	65001 i
*>	192.168.3.0	10.10.0.2			0 65500	65001 i

```
CE-WEST#
```

```
show ip route 192.168.1.10
```

```
Routing entry for 192.168.1.0/24
```

```
<<<<<
```

```
Known via "bgp 65000", distance 20, metric 0
```

```
Tag 65500, type external
```

```
Last update from 10.10.0.2 00:00:37 ago
```

```
Routing Descriptor Blocks:
```

```
* 10.10.0.2, from 10.10.0.2, 00:00:37 ago
```

```
Route metric is 0, traffic share count is 1
```

```
AS Hops 2
```

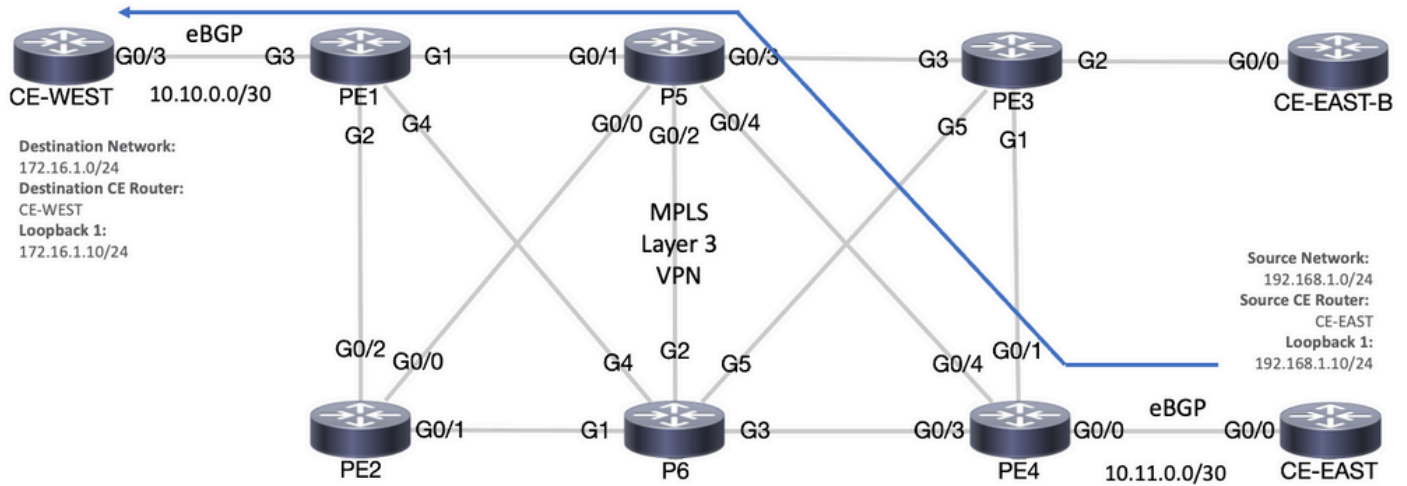
```
Route tag 65500
```

```
MPLS label: none
```



# Verificação

Agora, a acessibilidade entre a origem e o destino é bem-sucedida e pode ser confirmado que o traceroute passa pelo mesmo caminho de switch de rótulo que foi rastreado através da rede MPLS:



Caminho de Encaminhamento

```
<#root>
```

```
CE-EAST#
```

```
ping 172.16.1.10 source loopback 1
```

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

```
Sending 5, 100-byte ICMP Echos to 172.16.1.10, timeout is 2 seconds:
```

```
Packet sent with a source address of 192.168.1.10
```

```
!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 7/7/9 ms
```

```
<<<<<
```

```
CE-EAST#
```

```
traceroute 172.16.1.10 source loop1 probe 1 numeric
```

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

```
Tracing the route to 172.16.1.10
```

```
VRF info: (vrf in name/id, vrf out name/id)
```

```
 1 10.11.0.2 2 msec
```

```
 2 10.0.0.16 [MPLS: Labels 24001/16 Exp 0] 9 msec
```

```
 3 10.10.0.2 [MPLS: Label 16 Exp 0] 8 msec
```

```
 4 10.10.0.1 9 msec
```

```
RP/0/0/CPU0:P5#
```

```
show ipv4 interface brief
```

```
Wed Sep 20 18:23:47.158 UTC
```

Interface	IP-Address	Status	Protocol	Vrf-Name
Loopback0	10.10.10.5	Up	Up	default
MgmtEth0/0/CPU0/0	unassigned	Shutdown	Down	default
GigabitEthernet0/0/0/0	10.0.0.7	Up	Up	default
<b>GigabitEthernet0/0/0/1 10.0.0.1 Up Up default</b>				
<<<<<				
GigabitEthernet0/0/0/2	10.0.0.10	Up	Up	default
GigabitEthernet0/0/0/3	10.0.0.14	Up	Up	default
<b>GigabitEthernet0/0/0/4 10.0.0.16 Up Up default</b>				
<<<<<				
RP/0/0/CPU0:P5#				

## Comandos de verificação do Cisco IOS XE

<#root>

### MPLS/LDP

```
show mpls interfaces
show mpls forwarding-table
show mpls ldp bindings [destination prefix]
show mpls ldp neighbor [neighbor address]
clear mpls ldp neighbor [neighbor address|*]
```

### RIB and CEF

```
show ip vrf [detail]
show run vrf
show ip route [destination prefix]
show ip route vrf <name> [destination prefix]
show ip cef vrf <name> [destination prefix]
show ip cef exact-route <source IP> <destination IP>
show ip cef vrf <name> exact-route <source IP> <destination IP>
```

### BGP/VPNv4

```
show ip bgp [neighbors] <neighbor address>
show bgp vpnv4 unicast all [summary|destination prefix]
show bgp vpnv4 unicast all neighbor <neighbor address> advertised-routes
show bgp vpnv4 unicast vrf <name> neighbors <neighbor IP> advertised-routes
show bgp vpnv4 unicast vrf <name> <prefix>
show bgp vpnv4 unicast rd <value> <destination IP>
```

## Comandos de verificação do Cisco IOS XR

<#root>

#### **MPLS/LDP**

```
show mpls interfaces
show mpls forwarding
show mpls ldp bindings [destination prefix/mask]
show mpls ldp neighbor [neighbor address]
show mpls forwarding prefix [destination prefix/mask]
show mpls forwarding prefix [destination prefix/mask] detail hardware egress
clear mpls ldp neighbor [neighbor address]
```

#### **RIB and CEF**

```
show vrf [name|all]
show run vrf [name]
show route [destination prefix]
show route vrf <name> [destination prefix]
show cef vrf <name> [destination prefix]
show cef exact-route <source IP> <destination IP>
show cef vrf <name> exact-route <source IP> <destination IP>
```

#### **BGP/VPNv4**

```
show bgp vpnv4 unicast [summary|destination prefix/mask]
show bgp vpnv4 unicast neighbors <neighbor address> advertised-routes
show bgp vpnv4 unicast vrf <name> [prefix]
show bgp vrf <name> neighbors <neighbor IP> advertised-routes
show bgp vpnv4 unicast rd [value|all] [destination IP]
```

## Informações Relacionadas

- [Guia de configuração MPLS básica](#)
- [Configurar uma rede VPN MPLS básica](#)
- [Como solucionar problemas de MPLS VPN](#)
- [Verificar a Conectividade de Ponta a Ponta em um SP de Roteamento de Segmento](#)

## Sobre esta tradução

A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês ([link fornecido](#)) seja sempre consultado.