

# Vérifier le transfert VPN de couche MPLS 3

## Table des matières

---

[Introduction](#)

[Conditions préalables](#)

[Exigences](#)

[Composants utilisés](#)

[Informations générales](#)

[Conventions](#)

[Topologie](#)

[Dépannage](#)

[Informations initiales](#)

[Vérification](#)

[Commandes de vérification Cisco IOS XE](#)

[Commandes de vérification de Cisco IOS XR](#)

[Informations connexes](#)

---

## Introduction

Ce document décrit le processus de vérification de la connectivité de bout en bout sur un réseau principal VPN de couche 3 MPLS.

## Conditions préalables

### Exigences

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Connaissance du routage IP de base
- Connaissance de Cisco IOS® XE et de la ligne de commande de Cisco IOS® XR

### Composants utilisés

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- Routeur avec logiciel Cisco IOS XR
- Routeur avec logiciel Cisco IOS XE

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l'incidence possible des commandes.

## Informations générales

L'objectif de ce document est de démontrer les étapes de vérification et de dépannage de base pour vérifier la connectivité et le transfert entre deux routeurs CE (Customer Edge) interconnectés avec BGP (Border Gateway Protocol) par un réseau principal VPN de couche 3 MPLS avec un mélange de routeurs Cisco IOS XE et Cisco IOS XR agissant comme des routeurs PE (Provider Edge) et P (Provider).

## Conventions

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

## Topologie

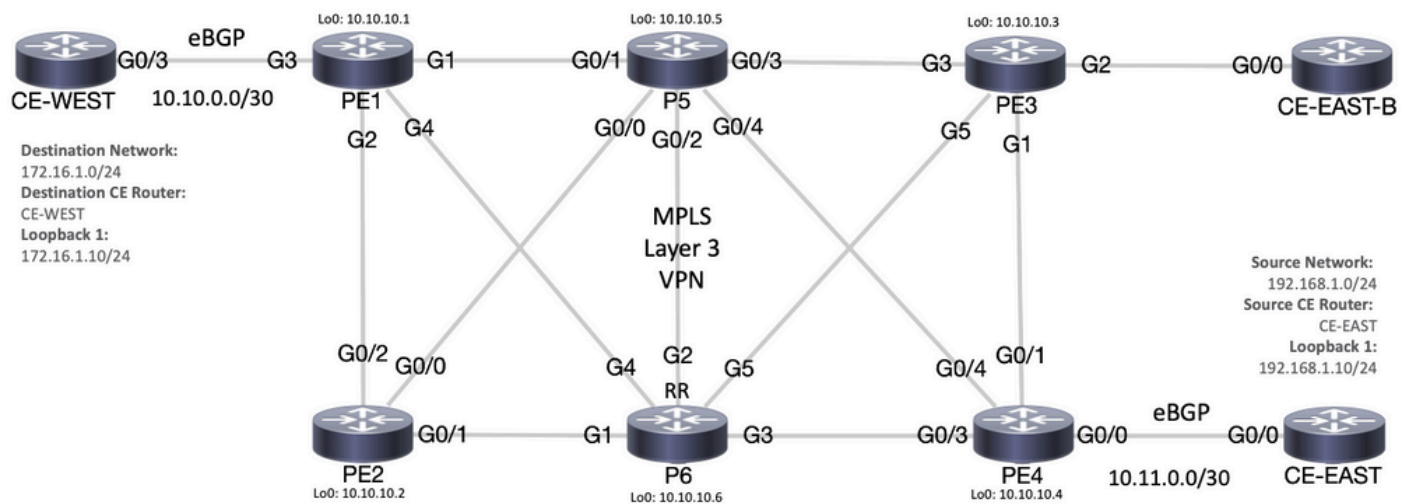


Schéma de topologie MPLS

## Dépannage

### Informations initiales

Réseau source : 192.168.1.0/24

Routeur CE source : CE-EAST

Réseau de destination : 172.16.1.0/24

Routeur CE de destination : CE-WEST

D'après les informations initiales et la topologie, l'accessibilité doit réussir entre l'adresse source 192.168.1.10 représentée par Loopback1 sur le routeur CE-EAST et l'adresse de destination 172.16.1.10 représentée par Loopback1 sur le routeur 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
```

L'accessibilité ICMP et une commande traceroute ont été utilisées pour commencer à vérifier la connectivité entre ces adresses source et de destination. Toutefois, d'après les résultats suivants, il est possible de constater que cette opération n'a pas abouti :

```
<#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 *
```

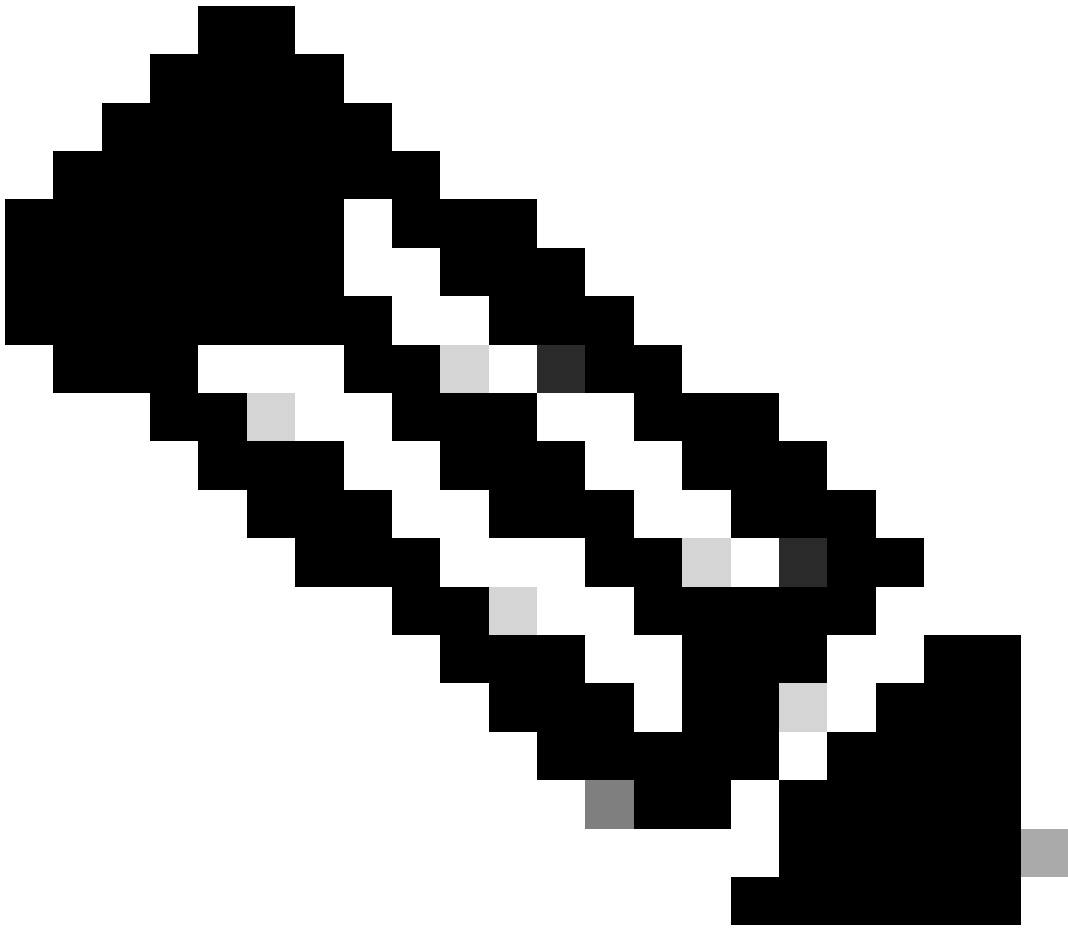
```
10 *
```

```
11 *
```

12 \*  
13 \*  
14 \*  
15 \*  
16 \*  
17 \*  
18 \*  
19 \*  
20 \*  
21 \*  
22 \*  
23 \*  
24 \*  
25 \*  
26 \*  
27 \*  
28 \*  
29 \*  
30 \*

CE-EAST#

---



---

---

Remarque : lors du dépannage, l'utilisation d'une commande traceroute lorsqu'elle est connectée à un réseau MPLS peut être moins efficace, car certains fournisseurs de services ont tendance à configurer la commande `no mpls ip propagate-ttl forward` dans Cisco IOS XE ou la commande `mpls ip-ttl-propagate disable forwarded` dans Cisco IOS XR pour masquer tous les routeurs LSR (Label Switch Router) dans le cœur (à l'exception toutefois des routeurs PE d'entrée et de sortie).

---

Lors de la vérification de l'état du routeur CE source, étant donné que ce routeur n'a pas de VRF (Virtual Route Forwarding) et qu'il ne reconnaît pas MPLS, vous devez vérifier le RIB (Routing Information Base), le CEF (Cisco Express Forwarding) et le BGP. Dans les sorties suivantes, on peut observer qu'il existe une entrée de routage connue via BGP vers le sous-réseau de destination 172.16.1.0/24 et accessible via l'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#

Comme le routeur source CE-EAST dispose d'une route vers la destination installée dans le RIB, il est temps d'examiner le routeur de périphérie fournisseur PE4 (PE d'entrée), comme indiqué dans la topologie. À ce stade, le VRF et les identificateurs de route sont configurés, ainsi que l'importation et l'exportation de la route cible, comme indiqué sur les sorties suivantes :

<#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#

D'après le résultat précédent, le nom VRF « EAST » a été défini avec l'importation route-target pour 65000:1. La table de routage VRF peut maintenant être vérifiée, ce qui permet de déterminer si PE4 dispose d'une route vers l'adresse IP de destination 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#
```

Comme ce PE est un périphérique Cisco IOS XR, le mot clé « detail » peut être utilisé à la fin de la commande show route vrf <name> pour examiner certaines informations supplémentaires telles que l'étiquette VPNv4 imposée par MP-BGP (Multiprotocol BGP) et la source RD (Route Distinguisher) à partir du préfixe :

<#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#
```

Maintenant, regardons le préfixe BGP VPNv4 qui a été importé dans le VRF, observons qu'il s'agit du même label 16 du résultat précédent et qu'il a également la communauté étendue 65000:1. Il est également important de noter que 10.10.10.1 est le tronçon suivant dont PE4 a besoin pour pouvoir effectuer une récursivité de route, l'adresse suivante « from 10.10.10.6 » est l'homologue BGP que PE4 a utilisé pour apprendre ce préfixe (dans ce scénario, il s'agit du réflecteur de route 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
```

```
<<<<<
```

En examinant CEF avec le mot clé exact-route au niveau VRF, vous pouvez avoir une idée de l'interface de sortie pour les paquets. Cette commande peut également donner quelques détails importants, car elle montre les deux étiquettes imposées au préfixe, 24001 et 16, la raison étant que l'étiquette 16 provient de BGP VPNv4 et l'étiquette 24001 provient de 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}

<<<<<

Comme étape suivante, la commande show bgp vpnv4 unicast est utilisée pour vérifier les routes VPNv4 qui sont apprises par ce PE. Ce résultat montre les informations avant l'importation du préfixe VPNv4 dans le VRF. N'oubliez pas que le RT (Route Target) configuré (pour cet exemple, les RT importés sont 65000:1, 65001:1, 65001:2) indique les routes et les VRF à importer :

<#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 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          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          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

Dans cet exemple, la table VPNv4 peut être petite, mais dans un environnement de production, au lieu d'examiner tous les préfixes VPNv4, vous pouvez limiter la vérification à la distance et au préfixe spécifiques avec la commande suivante :

```
<#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
```

À ce stade, le plan de contrôle MP-BGP a le préfixe de destination et les étiquettes LDP et VPNv4 {24001/16} respectivement, l'interface de sortie pour ce trafic semble être Gi0/0/0/4 et le saut suivant où le trafic doit être transféré est 10.10.10.1. Mais, Existe-t-il une autre option pour vérifier l'interface de sortie préférée ? Il est temps de jeter un oeil à la table de transfert MPLS ou LFIB (Label Forwarding Information Base). Avec l'utilisation de la commande show mpls forwarding deux entrées sont affichées vers la destination 10.10.10.1 (Loopback0 de PE1), un chemin avec une interface sortante de Gi0/0/0/4 et un tronçon suivant de 10.0.0.16 (routeur P5) où l'étiquette sortante imposée est 24001 et un autre chemin à travers Gi0/0/0/3 avec un tronçon suivant de 10.0.0.13 (routeur P6) et une étiquette sortante 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
```

D'après les résultats précédents, il est clair qu'il y a deux options de chemin où le trafic peut être équilibré en charge, mais il y a deux façons qui peuvent aider à déterminer lequel est le chemin préféré. Une façon consiste à utiliser la commande `show cef exact-route <source IP> <destination IP>` en ajoutant le Loopback0 à partir du PE source et le Loopback0 à partir du PE de destination. Comme indiqué dans le résultat suivant, le chemin préféré passe par 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}
```

Une autre option consiste à d'abord vérifier la LIB (Label Information Base) et obtenir la liaison LDP de la destination Loopback0 (10.10.10.1 qui appartient au PE de sortie) en utilisant la commande `show mpls ldp bindings <prefix/mask>`, et une fois que l'étiquette de liaison locale est trouvée à partir de cette sortie, utilisez cette valeur d'étiquette dans la commande `show mpls forwarding exact-route label <label> ipv4 <source IP> <destination IP> detail` pour trouver le chemin préféré :

<#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)
```

Ensuite, il est important de vérifier le routeur de tronçon suivant qui se trouve dans le plan de données, pour cet exemple particulier, le routeur à vérifier est P5 (qui a l'interface 10.0.0.16). Le premier point à prendre en compte est la table de transfert MPLS, où l'étiquette locale pour le préfixe 10.10.10.1 doit être 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#

D'après le résultat précédent, on peut voir que l'entrée LFIB pour le préfixe 10.10.10.1/32 montre "Pop" comme l'étiquette sortante, ce qui signifie que ce routeur est l'avant-dernier saut Popping (PHP). Il montre également que le trafic doit être envoyé via Gi0/0/0/1 en fonction des informations LFIB, et cela peut également être vérifié lors de l'examen de CEF. La sortie de route exacte CEF suivante montre l'étiquette Null implicite comme étiquette imposée, ceci est dû au fait que le tronçon suivant connecté à Gi0/0/0/1 est le dernier routeur dans le chemin de commutation d'étiquette et est également le PE face au site de destination (CE-WEST). C'est aussi la raison pour laquelle le routeur P5 supprime et n'impose pas une autre étiquette au paquet, grâce à ce processus le routeur de sortie PE1 va recevoir un paquet sans étiquette 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 [f flags 0x0]
  path-idx 0 NHID 0x0 [0xa166e280 0xa166e674]
  next hop 10.0.0.0/32

```

Local adjacency

```
local label 24001 labels imposed {ImplNull}
```

```
<<<<<
```

Le dernier point pour vérifier le chemin du commutateur d'étiquette est PE1. Lorsque vous consultez la table de transfert MPLS, vous remarquerez qu'il n'y a pas d'entrée pour le préfixe 10.10.10.1/32 dans la LFIB :

```
<#root>
```

```
PE1#
```

```
show mpls forwarding-table
```

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

Comme vous l'avez compris, la raison de ce comportement est que le préfixe (10.10.10.1/32) appartient à PE1, et le routeur a également attribué une étiquette Null implicite à ce préfixe connecté. Vous pouvez vérifier cela à l'aide de la commande `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

```

Comme PE1 est un routeur Cisco IOS XE, l'utilisation de la commande `show bgp vpnv4 unicast all` ou `show bgp vpnv4 unicast rd <value> <destination IP>` peut aider à identifier et confirmer que le préfixe de destination 172.16.1.0/24 est appris correctement via MP-BGP. Le résultat de ces commandes affiche le préfixe après l'exportation :

<#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 la même manière, en regardant le préfixe BGP VPNv4 au VRF qui est le préfixe reçu par CE-WEST, avec l'utilisation de la commande `show bgp vpnv4 unicast vrf <name> <prefix>`, le résultat montre l'étiquette MP-BGP 16 qui a été transportée jusqu'à l'entrée PE4 ainsi que l'exportation RT configurée 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
```

Les dernières informations à vérifier sur ce PE sont les entrées RIB et CEF au niveau VRF vers l'IP de destination, contrairement à l'entrée vue sur PE4, il n'y a pas d'étiquette sur le RIB pour le préfixe 172.16.1.0/24, la raison est que c'est la route qui arrive du CE et cela est appris via eBGP et inséré dans la table de routage VRF avant que ce préfixe soit exporté dans VPNv4. Vous pouvez vérifier cela à l'aide des commandes `show ip route vrf <name> <prefix>` et `show ip cef vrf <name> <prefix>` montrées ci-dessous :

<#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
```

À ce stade, il a été confirmé que le préfixe de destination 172.16.1.0/24 a été appris correctement par la source du trafic CE (CE-EAST), qu'il a été propagé correctement via MP-BGP et que les étiquettes des boucles PE et Ps ont également été apprises sur le chemin du commutateur d'étiquettes. Cependant, l'accessibilité entre la source et la destination ne fonctionne pas et il reste encore un dernier routeur pour vérifier CE-WEST. La première chose à vérifier dans ce routeur est la table de routage, rappelez-vous que le préfixe IP source 192.168.1.0/24 devrait y apparaître :

<#root>

CE-WEST#

```
show ip route 192.168.1.10
```

```
% Network not in table
```

CE-WEST#

Le « Réseau non dans la table » est clairement le problème, la table BGP peut également être vérifiée, mais après avoir recherché le préfixe, il n'est pas là non plus :

<#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#
```

En revenant en arrière, vous pouvez vérifier si ce routeur de périphérie de fournisseur (PE1) annonce le préfixe au voisin eBGP CE-WEST, ceci peut être fait avec l'utilisation de la commande `show bgp vpnv4 unicast vrf <name> neighbors <neighbor IP> advertised-routes` montrée ci-dessous :

```
<#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
```

Sur la base de l'étape précédente, il peut être confirmé que le routeur PE1 annonce correctement le préfixe à CE-WEST, il est donc temps d'examiner les voisins BGP du côté 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 multisession 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  
Multisession 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 NLRIs 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

```

Le résultat précédent révèle qu'il y a une carte de route appliquée pour les annonces entrantes avec le nom "FILTER", après avoir regardé la configuration de la carte de route, il montre une clause de correspondance pointant vers une liste de préfixes avec une instruction permit pour 192.168.0.0/16, cependant ceci est incorrect car la liste de préfixes autorise seulement ce préfixe spécifique et pas tous ceux qui peuvent être inclus dans cette plage :

<#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

Avec une légère modification de la configuration de la liste de préfixes, la route vers 192.168.1.10 est maintenant installée dans le 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#

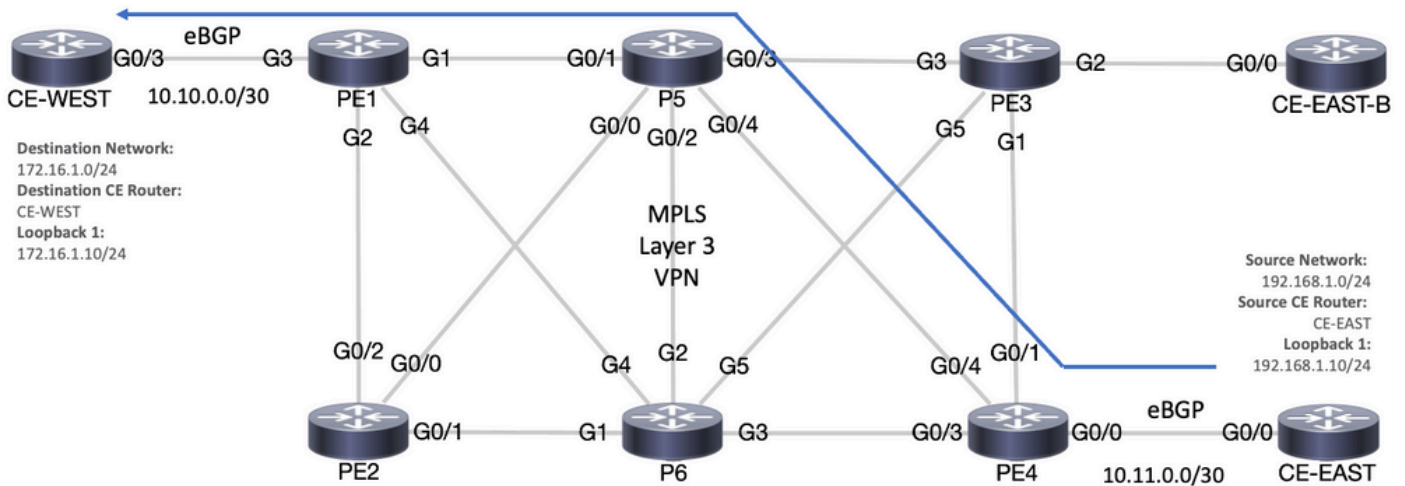
```
show ip route 192.168.1.10
```

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
```

## Vérification

Maintenant, l'accessibilité entre la source et la destination est réussie et il peut être confirmé que la commande traceroute passe par le même chemin de commutateur d'étiquette qui a été suivi sur le réseau MPLS :



Chemin de transfert

```
<#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

GigabitEthernet0/0/0/1 10.0.0.1 Up Up default

<<<<<

GigabitEthernet0/0/0/2	10.0.0.10	Up	Up	default
GigabitEthernet0/0/0/3	10.0.0.14	Up	Up	default

GigabitEthernet0/0/0/4 10.0.0.16 Up Up default

<<<<<

RP/0/0/CPU0:P5#

## Commandes de vérification 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>
```

## Commandes de vérification de 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]
```

## Informations connexes

- [Guide de configuration MPLS Basic MPLS](#)
- [Configurer un réseau VPN MPLS de base](#)
- [Comment dépanner le VPN MPLS](#)
- [Vérification de la connectivité de bout en bout sur un SP de routage de segment](#)

À propos de cette traduction

Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.