International Digital Organization for Scientific Research
IDOSR JOURNAL OF COMPUTER AND APPLIED SCIENCES 7(1):95-108,2022. ISSN: 2579-0803

Security Analysis for Virtual Private Network Based on Site to Site Circuit Switching (Vpns2scs) Case Study: Liquid Telecommunication Ggaba

Hamisi Sekiti and Adabara Ibrahim

Department of Electrical and Telecommunications Engineering, School of Engineering & Applied Sciences, Kampala International University, Uganda.

ABSTRACT

This study followed a descriptive pattern focusing on a point-to-point layer 2 design based on serial technology in relation to the efficiency in terms of cost, speeds, scalability and quality of service. The configuration of route exchange between PE and CE routers involved the implementation of a routing protocol (or static/default routes) on the CE routers. No specific configuration other than the regular routing protocol configuration was required on the CE routers. On the PE router, VRF routing contexts (or address family contexts) were required for route exchange between the PE and CE. Then, these routes were mutually redistributed with the MP-BGP process per VRF. The next tables show the steps that followed to configure the routers. We used the command sh ip route on the Customer Network to show the end-to-end connectivity between the CE routers within each VRF and their complete routing tables.
Keywords: Security analysis, virtual private networks, circuit switching

INTRODUCTION

Numbers of VPN customers like private companies, or public administrations have several scattered site locations, and need an excellent, reliable and secure connection between all of them, preferably using a single IP network, their own IP addressing plan and their proper traffic despite the fact that some other customers might be using the same infrastructure [1].
Multiprotocol Label Switching (MPLS) is a high performance telecommunications networks service that carries data and directs it from one network node to the next based on short path labels rather than long network addresses [2]. It speeds up while shaping network traffic flows and easily creates virtual links between data nodes. Also, it can encapsulate packets of various network protocols (multi-protocol) [3,4,5,6]. MPLS L3VPNs use a peer-to-peer model that uses Border Gateway Protocol (BGP) to distribute VPN-related information. It is highly scalable, protocol agnostic, allowing enterprise subscribers to outsource routing information to service providers, resulting in significant cost savings and a reduction in operational complexity for enterprises [7,8]. In an MPLS network, data packets are assigned labels. Packet forwarding decisions are made solely on the contents of this label, without the need to examine the packet itself. This allows creation of end-to-end circuits across any type of transport medium, using any protocol [5,9].

METHODOLOGY

Research design

This study followed a descriptive pattern focusing on a point-to-point layer 2 design based on serial technology in relation to the efficiency in terms of cost, speeds, scalability and quality of service.

Sources of information

These were mainly libraries of Cisco Networking Academy, and Internet searches. Relevant books and websites were visited. The obtained information from the internet were mainly from text books, journal presentations, technical reports, institutional records, and PDF files among others.

Design Configuration

All configurations were performed in the network shown in Figure 3.1. For simplicity issues, we redistributed only connected networks that were part of the VRF into the MP-BGP processes.

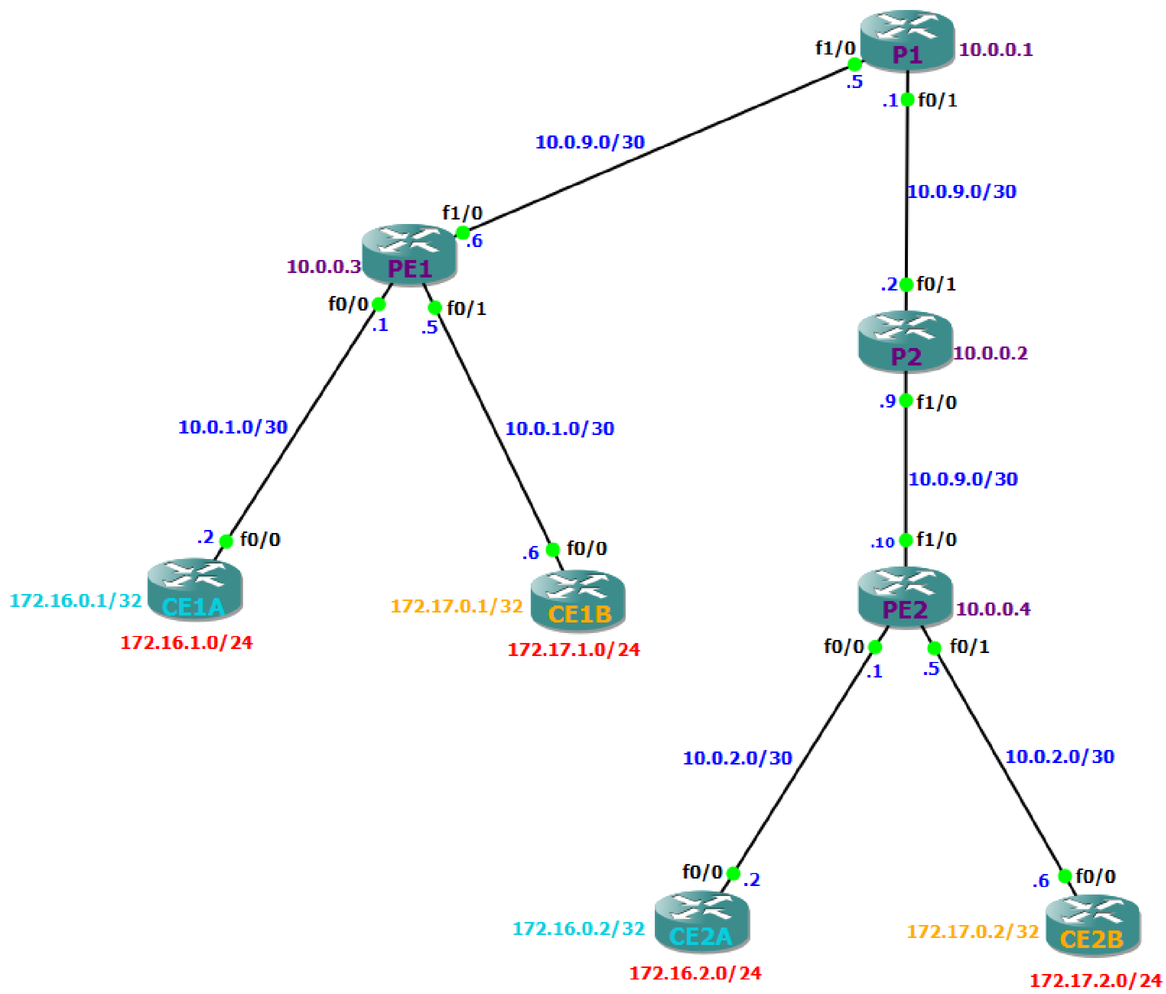


Figure 1 Network Topology

The topology in Figure 1 was attempted by implementing a simple intranet VPN between two sites belonging to two companies denoted as Customer_A and Customer_B. The customer network consisted of the CE routers CE1A, and CE2A for Customer_A; CE1B and CE2B for Customer_B. In addition, two loopbacks (loopback 0 and loopback 1) were configured as part of the VRF *Customer_A and Customer_B;* and finally redistributed into the MP-BGP routing contexts.

We also enable encrypted passwords on our provider routers (P1, P2, PE1 and PE2) to ensure security.

- **Design process steps**

The flow chart showing the process steps that was followed is as shown below in Figure 3.2:

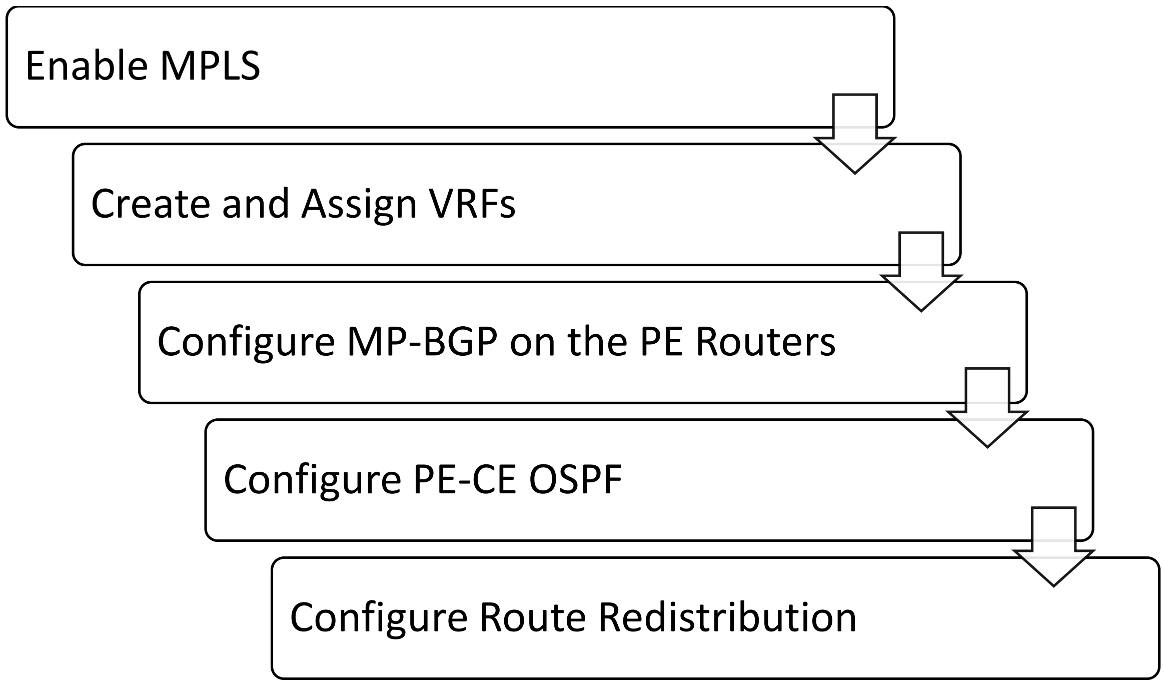


Figure 2 Design process steps

- **Hosts and Subnets**

Here are the types of subnets that we used to configure our routers.

Table 1 Classification of subnets used

CIDR	Host bits	Subnet mask	Addresses in subnet	Typical usage	Actual Usage
/24	8	255.255.255.0	$256 = 2^8$	Large LAN	Loopback 1
/30	2	255.255.255.252	$4 = 2^2$	"Glue network" (point to point links)	Interfaces
/32	0	255.255.255.255	$1 = 2^0$	Host route	Loopback 0

Configuration of Routers

The configuration of route exchange between PE and CE routers involved the implementation of a routing protocol (or static/default routes) on the CE routers. No specific configuration other than the regular routing protocol configuration was required on the CE routers. On the PE router, VRF routing contexts (or address family contexts) were required for route exchange between the PE and CE. Then, these routes were mutually redistributed with the MP-BGP process per VRF. The next tables show the steps that followed to configure the routers.

Enable Encrypted Password

We used these commands on each Provider Router and Provider Edge Router:

- **enable password** *bL@61ablA* to activate the password
- **service password-encryption** to encrypt the password

Table 2 First configuration steps

Step 1	Enable MPLS
	First we needed to enable MPLS on all the two Providers (P1 and P2); and the Providers to Provider Edges links with the **mpls ip** interface command. We verified the configuration of MPLS interfaces with **show mpls interfaces.** P1(config)# **int f0/1** P1(config-if)# **mpls ip** P1(config-if)# **int f1/0** P1(config-if)# **mpls ip** P1(config-if)# d**o show mpls interfaces** Interface IP Tunnel Operational FastEthernet0/1 Yes (ldp) No Yes FastEthernet1/0 Yes (ldp) No Yes P2(config)# **int f0/1** P2(config-if)# **mpls ip** P2(config-if)# **int f1/0** P2(config-if)# **mpls ip** PE1(config)# **int f1/0** PE1(config-if)# **mpls ip** PE2(config)# **int f1/0** PE2(config-if)# **mpls ip** LDP adjacencies could be verified with the command **sh mpls ldp neighbor** P1# **sh mpls ldp neighbor** Peer LDP Ident: 10.0.0.2:0; Local LDP Ident 10.0.0.1:0 TCP connection: 10.0.0.2.45114 - 10.0.0.1.646 State: Oper; Msgs sent/rcvd: 12/13; Downstream Up time: 00:02:43 LDP discovery sources: FastEthernet0/1, Src IP addr: 10.0.9.2 Addresses bound to peer LDP Ident: 10.0.9.2 10.0.9.9 10.0.0.2 Peer LDP Ident: 10.0.0.3:0; Local LDP Ident 10.0.0.1:0 TCP connection: 10.0.0.3.20327 - 10.0.0.1.646 State: Oper; Msgs sent/rcvd: 12/12; Downstream Up time: 00:02:25 LDP discovery sources: FastEthernet1/0, Src IP addr: 10.0.9.6 Addresses bound to peer LDP Ident: 10.0.9.6 10.0.0.3
Step 2	**Create and Assign VRFs**

After the previous step, we created customer VRFs on our PE routers and assigned to them customer-facing interfaces. We assigned to each VRF a route distinguisher (RD) to uniquely identify prefixes as belonging to that VRF and one or more route targets (RTs) to specify how routes should be imported to and exported from the VRF.

VRF configuration had to be performed on both PE routers.

PE1(config)# **ip vrf Customer_A**
PE1(config-vrf)# **rd 65000:1**
PE1(config-vrf)# **route-target both 65000:1**
PE1(config-vrf)# **ip vrf Customer_B**
PE1(config-vrf)# **rd 65000:2**
PE1(config-vrf)# **route-target both 65000:2**
PE2(config)# **ip vrf Customer_A**
PE2(config-vrf)# **rd 65000:1**
PE2(config-vrf)# **route-target both 65000:1**
PE2(config-vrf)# **ip vrf Customer_B**
PE2(config-vrf)# **rd 65000:2**
PE2(config-vrf)# **route-target both 65000:2**

The command **route-target both** was used as a shortcut for the two commands **route-target import** and **route-target export**, which appeared separately in the running configuration.

Then, we assigned the appropriate interfaces to each VRF and reapply their IP addresses.

The command **sh ip vrf int** was used to verify interface VRF assignment and addressing.

PE1(config)# **int f0/0**
PE1(config-if)# **ip vrf forward Customer_A**
PE1(config-if)# **ip add 10.0.1.1 255.255.255.252**
PE1(config-if)# **int f0/1**
PE1(config-if)# **ip vrf forward Customer_B**
PE1(config-if)# **ip add 10.0.1.5 255.255.255.252**
PE1(config-if)# **end**

PE1# **sh ip vrf interfaces**

Interface	IP-Address	VRF	Protocol
Fa0/0	10.0.1.1	Customer_A	up
Fa0/1	10.0.1.5	Customer_B	up

PE2(config)# **int f0/0**
PE2(config-if)# **ip vrf forwarding Customer_A**
PE2(config-if)# **ip add 10.0.2.1 255.255.255.252**
PE2(config-if)# **int f0/1**
PE2(config-if)# **ip vrf forward Customer_B**
PE2(config-if)# **ip add 10.0.2.5 255.255.255.252**
PE2(config-if)# **end**
PE2# **sh ip vrf interfaces**

Interface	IP-Address	VRF	Protocol
Fa0/0	10.0.2.1	Customer_A	up
Fa0/1	10.0.2.5	Customer_B	up

Table 3 Final configuration steps

Step 3	Configure MP-BGP on the PE Routers
	We configured multiprotocol BGP (MP-BGP) to advertise VRF routes from one PE router to the other, MP-BGP ran only on the PE routers: P routers relied entirely on the provider IGP and MPLS to forward traffic through the provider network, and CE routers had no knowledge of routes outside their own VRF. Both PE routers existed in BGP AS 65000. PE1(config)# **router bgp 65000** PE1(config-router)# **neighbor 10.0.0.4 remote-as 65000** PE1(config-router)# **neighbor 10.0.0.4 update-source loopback 0** PE1(config-router)# **address-family vpnv4** PE1(config-router-af)# **neighbor 10.0.0.4 activate** PE2(config)# **router bgp 65000** PE2(config-router)# **neighbor 10.0.0.3 remote-as 65000** PE2(config-router)# **neighbor 10.0.0.3 update-source loopback 0** PE2(config-router)# **address-family vpnv4** PE2(config-router-af)# **neighbor 10.0.0.3 activate** We noticed that a bit more configuration than we provided appeared when we looked at the running configuration of the BGP process on either PE router: PE1# **sh running-config | section router bgp** router bgp 65000 no synchronization bgp log-neighbor-changes neighbor 10.0.0.4 remote-as 65000 neighbor 10.0.0.4 update-source Loopback0 no auto-summary ! address-family vpnv4 neighbor 10.0.0.4 activate neighbor 10.0.0.4 send-community extended exit-address-family ! address-family ipv4 vrf Customer_B no synchronization exit-address-family ! address-family ipv4 vrf Customer_A no synchronization exit-address-family In addition to our VPNv4 address family, address families for the two customer VRFs were created automatically. Also, support for extended community strings were added to the VPNv4 neighbor configuration. To verify that the MP-BGP adjacency between PE1 and PE2 was formed successfully, we used the command **sh bgp vpnv4 unicast all summary:** PE1# **sh bgp vpnv4 unicast all summary** BGP router identifier 10.0.0.3, local AS number 65000 BGP table version is 1, main routing table version 1 Neighbor V AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State/PfxRcd 10.0.0.4 4 65000 12 12 1 0 0 00:06:05 0 At that time, there wouldn't be any routes in the BGP table, because we had not specified anything to be advertised or redistributed.

Step 4	**Configure PE-CE OSPF**

We configured an IGP between each PE router and its attached CE routers to exchange routes with the customer sites. We used OSPF for this lab, but we could just as easily use another IGP like EIGRP or RIP.

Single-area OSPF had already been configured on the CE routers; all CE interfaces were in area 0. We had to note that although we were using OSPF between each of the CE routers and its upstream PE router, these OSPF processes would be isolated from the provider OSPF topology.

As the provider OSPF process had already been configured on the PE routers as process 1, we configured an additional OSPF process for each CE router on each PE router. Each PE router therefore had three OSPF processes total: one for the provider network, and one for each CE router. Whereas the provider OSPF process existed in the global routing table, the two CE processes would each be assigned to their respective customer VRFs.

PE1(config)# **router ospf 2 vrf Customer_A**
PE1(config-router)# **router-id 10.0.1.1**
PE1(config-router)# **int f0/0**
PE1(config-if)# **ip ospf 2 area 0**
PE1(config-if)# **router ospf 3 vrf Customer_B**
PE1(config-router)# **router-id 10.0.1.5**
PE1(config-router)# **int f0/1**
PE1(config-if**)# ip ospf 3 area 0**
PE2(config)# **router ospf 2 vrf Customer_A**
PE2(config-router)# **router-id 10.0.2.1**
PE2(config-router)# **int f0/0**
PE2(config-if)# **ip ospf 2 area 0**
PE2(config-if)# **router ospf 3 vrf Customer_B**
PE2(config-router**)# router-id 10.0.2.5**
PE2(config-router)# **int f0/1**
PE2(config-if)# **ip ospf 3 area 0**

We should see each PE router form an OSPF adjacency with both of its attached CE routers, and the customer routes should appear in the VRF tables on the PE routers.

PE1# **sh ip route vrf Customer_A**

```
Routing Table: Customer_A
...
172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
O       172.16.1.0/24 [110/11] via 10.0.1.2, 00:04:21, FastEthernet0/0
O       172.16.0.1/32 [110/11] via 10.0.1.2, 00:04:21, FastEthernet0/0
10.0.0.0/30 is subnetted, 1 subnets
C       10.0.1.0 is directly connected, FastEthernet0/0
```

PE1# **sh ip route vrf Customer_B**

```
Routing Table: Customer_B
...
172.17.0.0/16 is variably subnetted, 2 subnets, 2 masks
O       172.17.1.0/24 [110/11] via 10.0.1.6, 00:03:03, FastEthernet0/1
O       172.17.0.1/32 [110/11] via 10.0.1.6, 00:03:04, FastEthernet0/1
10.0.0.0/30 is subnetted, 1 subnets
C       10.0.1.4 is directly connected, FastEthernet0/1
```

Step 5	**Configure Route Redistribution**

At this point, we had our MPLS and MP-BGP backbone up and running, and our CE routers would be able to send routes to our PE routers within their VRFs. The last step was to join everything together by turning on route redistribution from the customer-side OSPF processes into MP-BGP and vice versa on the PE routers.

First we configured redistribution of CE routes in each VRF into MP-BGP. This was done under the BGP IPv4 address family for each VRF.

PE1(config)# **router bgp 65000**
PE1(config-router)# **address-family ipv4 vrf Customer_A**
PE1(config-router-af)# **redistribute ospf 2**
PE1(config-router-af)# **address-family ipv4 vrf Customer_B**
PE1(config-router-af)# **redistribute ospf 3**
PE2(config)# router **bgp 65000**
PE2(config-router)# **address-family ipv4 vrf Customer_A**
PE2(config-router-af)# **redistribute ospf 2**
PE2(config-router-af)# **address-family ipv4 vrf Customer_B**
PE2(config-router-af)# **redistribute ospf 3**

This enabled redistribution of OSPF routes into BGP for transport across the provider network between the two sites. We were able to verify that the routes learned from the customer sites (the 172.16.0.0/16 and 172.17.0.0/16 networks) appeared in the BGP tables for their respective VRFs.

PE1# **sh ip bgp vpnv4 vrf Customer_A**

```
...
Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 65000:1 (default for vrf Customer_A)
*> 10.0.1.0/30      0.0.0.0                032768?
*>i10.0.2.0/30      10.0.0.4               01000?
*> 172.16.0.1/32    10.0.1.2               1132768?
*>i172.16.0.2/32    10.0.0.4               111000?
*> 172.16.1.0/24    10.0.1.2               1132768?
*>i172.16.2.0/24    10.0.0.4               111000?
```

PE1# **sh ip bgp vpnv4 vrf Customer_B**

```
...
Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 65000:2 (default for vrf Customer_B)
*> 10.0.1.4/30      0.0.0.0                032768?
*>i10.0.2.4/30      10.0.0.4               01000?
*> 172.17.0.1/32    10.0.1.6               1132768?
*>i172.17.0.2/32    10.0.0.4               111000?
*> 172.17.1.0/24    10.0.1.6               1132768?
*>i172.17.2.0/24    10.0.0.4               111000?
```

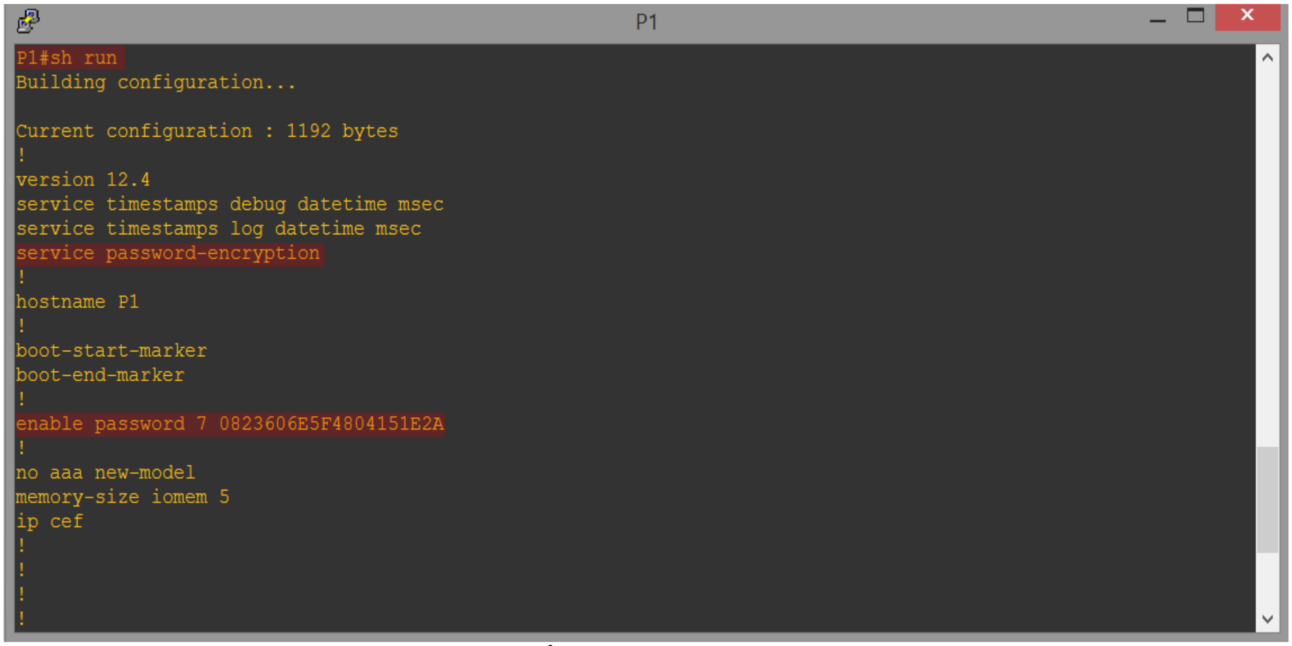
The final step was to complete the redistribution in the opposite direction: from BGP into the customer OSPF processes.

PE1(config)# **router ospf 2**
PE1(config-router)# **redistribute bgp 65000 subnets**
PE1(config-router)# **router ospf 3**
PE1(config-router)# **redistribute bgp 65000 subnets**
PE2(config)# **router ospf 2**
PE2(config-router)# **redistribute bgp 65000 subnets**
PE2(config-router)# **router ospf 3**
PE2(config-router)# **redistribute bgp 65000 subnets**

RESULTS

Password Encryption test[1]

The Figure 3.1 shows that we successfully enabled and encrypted a password for P1 router by running the command **sh run** to see the configurations on the Provider Router.



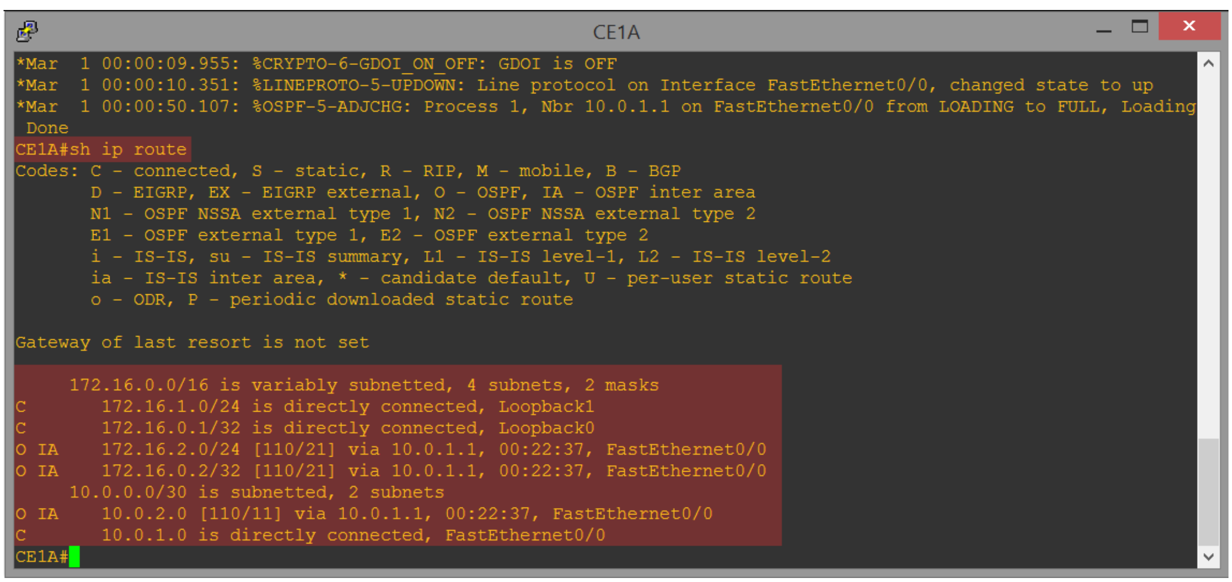
```
P1
P1#sh run
Building configuration...

Current configuration : 1192 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
!
hostname P1
!
boot-start-marker
boot-end-marker
!
enable password 7 0823606E5F4804151E2A
!
no aaa new-model
memory-size iomem 5
ip cef
!
!
!
!
```

Figure 3.1 P1 Password Encryption Results

IP Route test

We used the command **sh ip route** on the Customer Network to show the end-to-end connectivity between the CE routers within each VRF and their complete routing tables.



```
CE1A
*Mar  1 00:00:09.955: %CRYPTO-6-GDOI_ON_OFF: GDOI is OFF
*Mar  1 00:00:10.351: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
*Mar  1 00:00:50.107: %OSPF-5-ADJCHG: Process 1, Nbr 10.0.1.1 on FastEthernet0/0 from LOADING to FULL, Loading
 Done
CE1A#sh ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
C       172.16.1.0/24 is directly connected, Loopback1
C       172.16.0.1/32 is directly connected, Loopback0
O IA    172.16.2.0/24 [110/21] via 10.0.1.1, 00:22:37, FastEthernet0/0
O IA    172.16.0.2/32 [110/21] via 10.0.1.1, 00:22:37, FastEthernet0/0
     10.0.0.0/30 is subnetted, 2 subnets
O IA    10.0.2.0 [110/11] via 10.0.1.1, 00:22:37, FastEthernet0/0
C       10.0.1.0 is directly connected, FastEthernet0/0
CE1A#
```

Figure 3.2 CE1A Routing Table

[1] We managed to follow the same process on the whole Provider Network as shown in Appendix A.

```
CE1B
*Mar  1 00:00:10.059: %CRYPTO-6-GDOI_ON_OFF: GDOI is OFF
*Mar  1 00:00:10.507: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
*Mar  1 00:00:50.271: %OSPF-5-ADJCHG: Process 1, Nbr 10.0.1.5 on FastEthernet0/0 from LOADING to FULL, Loading
 Done
CE1B#sh ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     172.17.0.0/16 is variably subnetted, 4 subnets, 2 masks
C       172.17.1.0/24 is directly connected, Loopback1
C       172.17.0.1/32 is directly connected, Loopback0
O IA    172.17.2.0/24 [110/21] via 10.0.1.5, 00:25:57, FastEthernet0/0
O IA    172.17.0.2/32 [110/21] via 10.0.1.5, 00:25:57, FastEthernet0/0
     10.0.0.0/30 is subnetted, 2 subnets
O IA    10.0.2.4 [110/11] via 10.0.1.5, 00:25:57, FastEthernet0/0
C       10.0.1.4 is directly connected, FastEthernet0/0
CE1B#
```

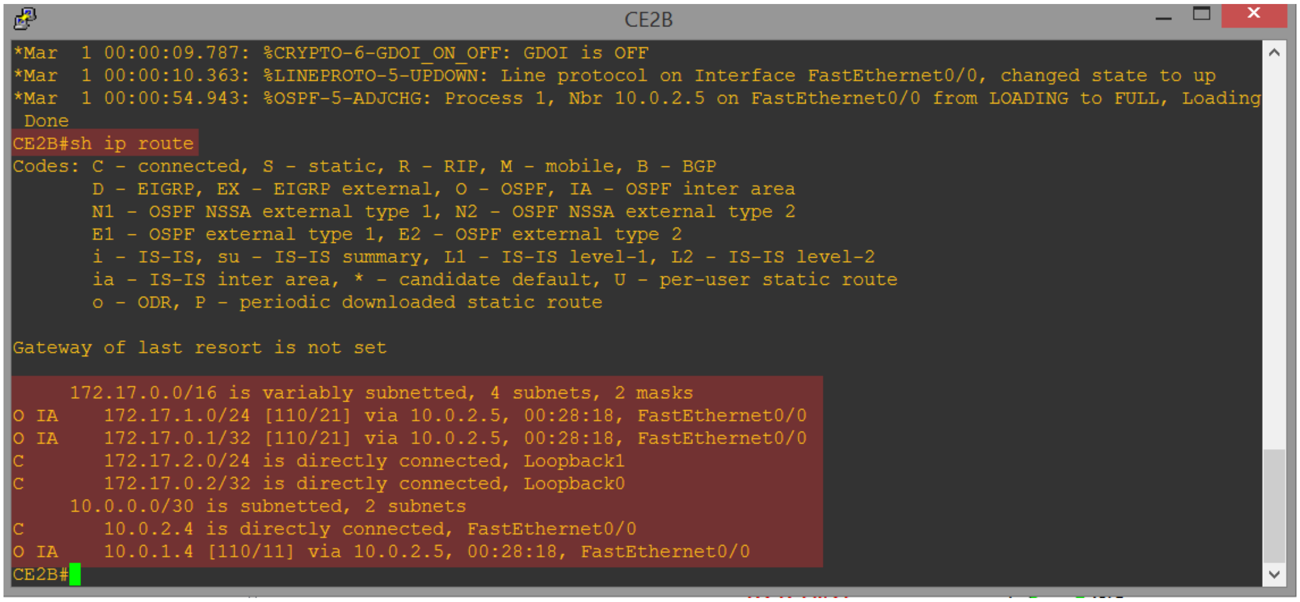
Figure 3.3 CE1B Routing Table

```
CE2A
*Mar  1 00:00:09.959: %CRYPTO-6-GDOI_ON_OFF: GDOI is OFF
*Mar  1 00:00:10.303: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
*Mar  1 00:00:50.171: %OSPF-5-ADJCHG: Process 1, Nbr 10.0.2.1 on FastEthernet0/0 from LOADING to FULL, Loading
 Done
CE2A#sh ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
O IA    172.16.1.0/24 [110/21] via 10.0.2.1, 00:26:26, FastEthernet0/0
O IA    172.16.0.1/32 [110/21] via 10.0.2.1, 00:26:26, FastEthernet0/0
C       172.16.2.0/24 is directly connected, Loopback1
C       172.16.0.2/32 is directly connected, Loopback0
     10.0.0.0/30 is subnetted, 2 subnets
C       10.0.2.0 is directly connected, FastEthernet0/0
O IA    10.0.1.0 [110/11] via 10.0.2.1, 00:26:26, FastEthernet0/0
CE2A#
```

Figure 3.4 CE2A Routing Table



```
*Mar  1 00:00:09.787: %CRYPTO-6-GDOI_ON_OFF: GDOI is OFF
*Mar  1 00:00:10.363: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
*Mar  1 00:00:54.943: %OSPF-5-ADJCHG: Process 1, Nbr 10.0.2.5 on FastEthernet0/0 from LOADING to FULL, Loading
 Done
CE2B#sh ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     172.17.0.0/16 is variably subnetted, 4 subnets, 2 masks
O IA    172.17.1.0/24 [110/21] via 10.0.2.5, 00:28:18, FastEthernet0/0
O IA    172.17.0.1/32 [110/21] via 10.0.2.5, 00:28:18, FastEthernet0/0
C       172.17.2.0/24 is directly connected, Loopback1
C       172.17.0.2/32 is directly connected, Loopback0
     10.0.0.0/30 is subnetted, 2 subnets
C       10.0.2.4 is directly connected, FastEthernet0/0
O IA    10.0.1.4 [110/11] via 10.0.2.5, 00:28:18, FastEthernet0/0
CE2B#
```

Figure 3.5 CE2B Routing Table

Ping Test
We were able to ping from one CE router to the other accordingly using both loopbacks IP addresses.

- CE1A to CE2A and CE2A to CE1A
- CE1B to CE2B and CE2B to CE1B

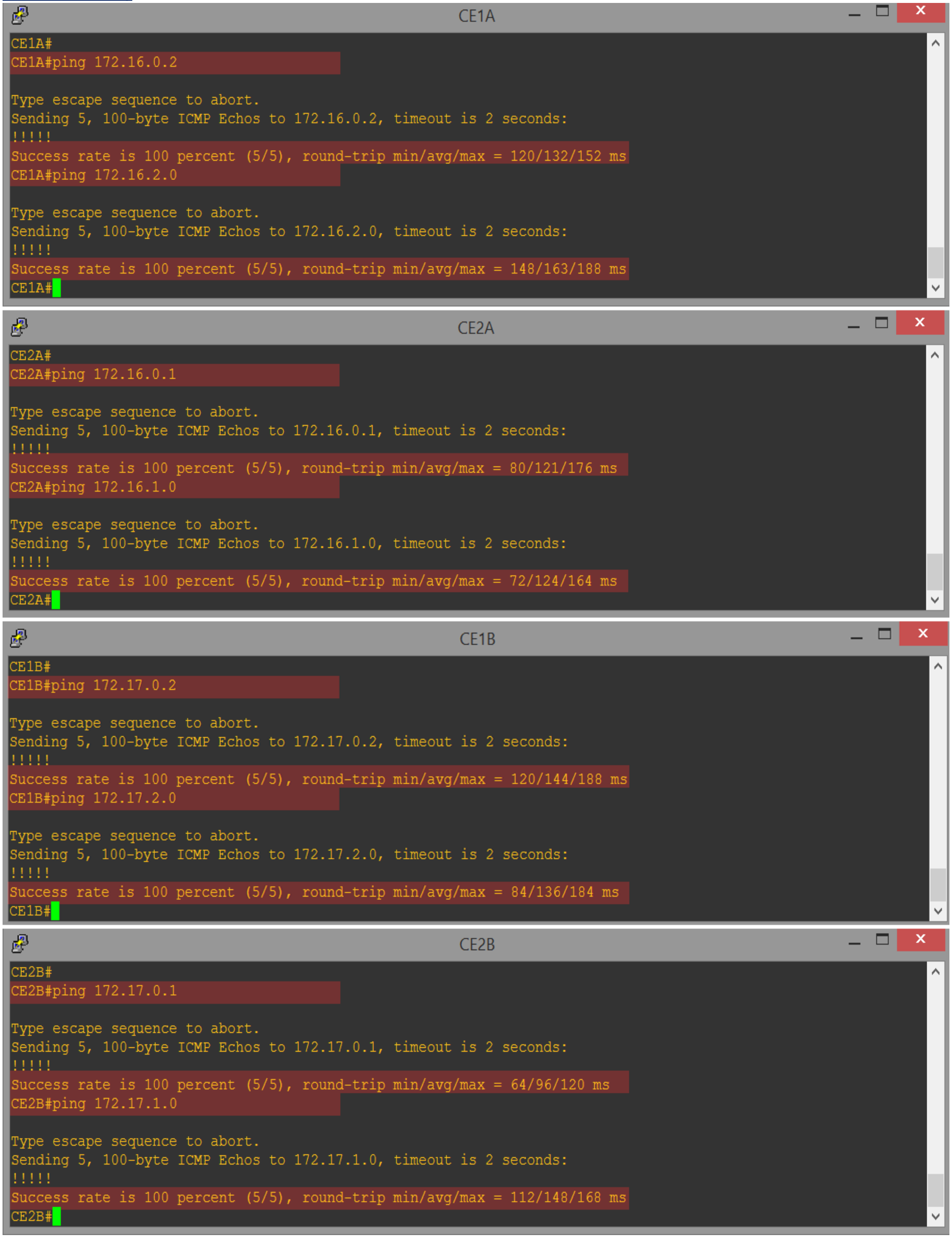


Figure 3. 6 Ping test results

Traceroute Test

We were able to perform a traceroute to verify the path taken, as well as the MPLS labels used to traverse the provider network accordingly using both loopbacks IP addresses.

- CE1A to CE2A and CE2A to CE1A: *Customer_A*
- CE1B to CE2B and CE2B to CE1B: *Customer_B*

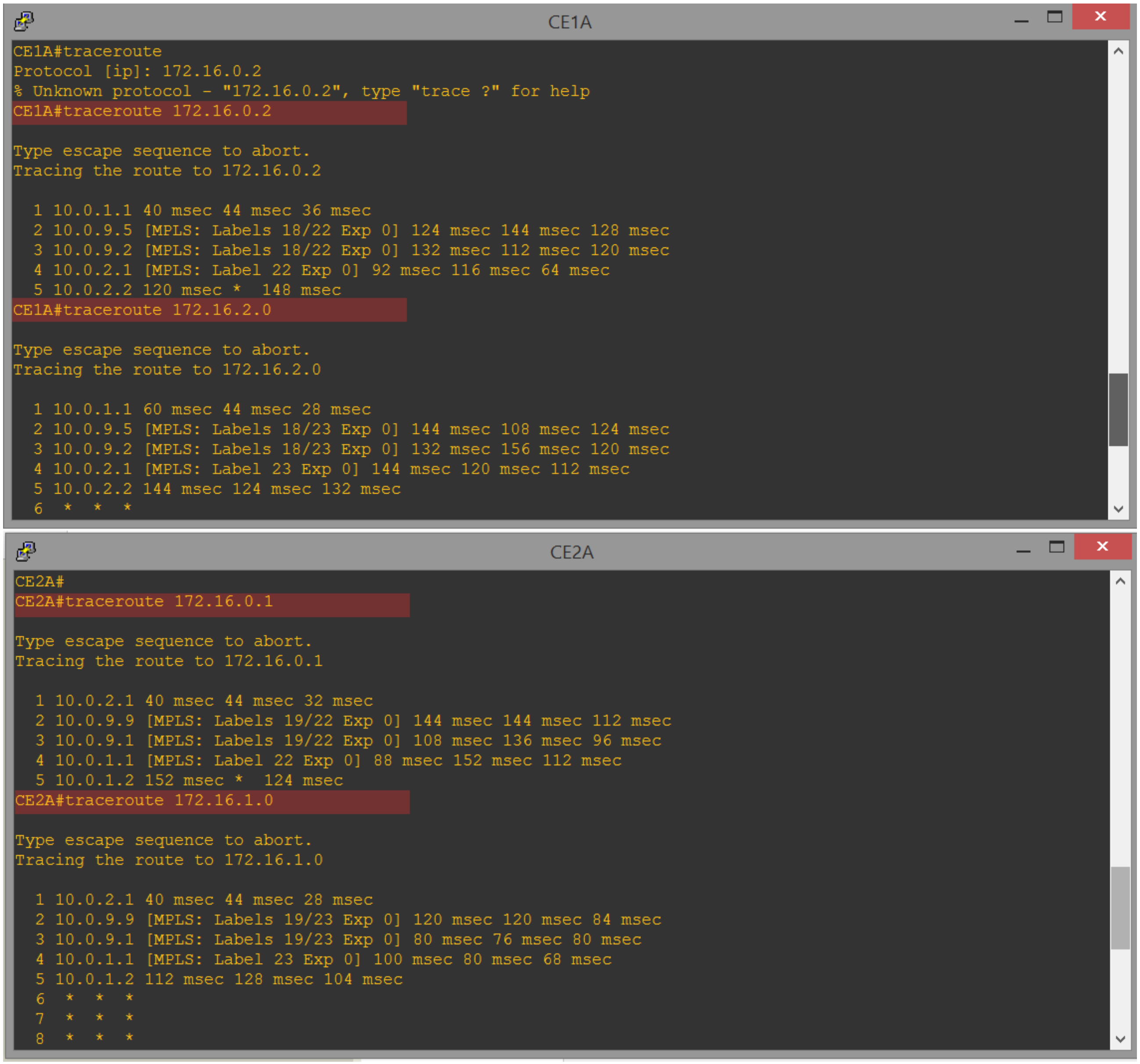


Figure 3.7 Customer_A traceroute

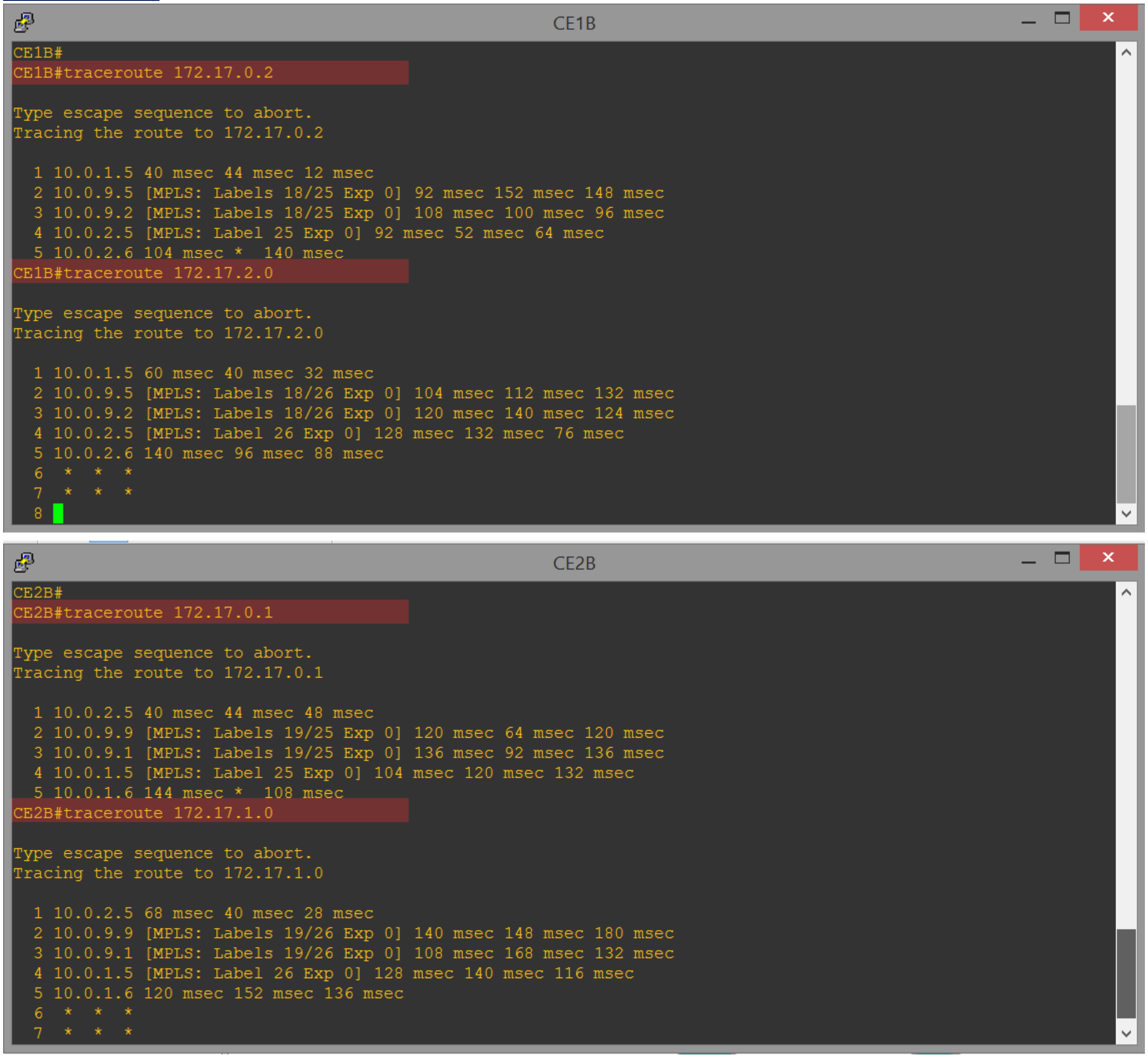


Figure 3.8 Customer_B traceroute

The results proved to be very conclusive on the benefits of MPLS in terms of speed and security.

CONCLUSION

MPLS is a rapidly expanding technology that provides a number of advantages to its users such as scalability, security and QoS. The limitations of the technology lie in the expense of constructing MPLS backbones with the intelligent devices to enable the Traffic Engineering and the bandwidth to support it. Also high-end skilled network engineers to design, build and run it are needed.

The aim of the project was met, and we managed to implement our MPLS VPN network system. The end-to-end connectivity was accurate, and the routing table all present. But, we faced different issues picking the right routers, and the perks of using GNS3 which used a large amount of CPU unless we didn't set every router's idle PC setting. The more the routers, the longer the task to set the Idle-Pc setting manually, and the longer it takes to input all commands. As the system failed sometimes, this work required a lot of patience and concentration, but despite of all the difficulties, the project in reference to the objectives stated at the beginning, the project was a success because its objectives have been achieved.

REFERENCES

1. Cisco Systems Inc. *Catalyst 4500 series Switch Cisco IOS Software*. USA: Cisco Press, 2019.s
2. Hucaby, D., McQuerry, S. and Whitaker A. (2010). *Cisco Router Configuration Handbook* (2nd Ed.). USA: Cisco Press.
3. Lewis, C. and Pickavance, S. (2006) *Selecting MPLS VPN Services* (1st Ed.). USA: Cisco Press.
4. Luc De Ghein. (2019). *MPLS Fundamentals* (1st Ed.). USA: Cisco Press.
5. Umesh Lakshman, Lancy Lobo. (2005). *MPLS Configuration on Cisco IOS Software (paperback*). USA: Cisco Press.
6. Masisani William Mufana and Adabara Ibrahim (2022). Monitoring with Communication Technologies of the Smart Grid. *IDOSR Journal of Applied Sciences* 7(1) 102-112.
7. Nabiryo Patience and Itodo Anthony Ejeh (2022). Design and Implementation of Base Station Temperature Monitoring System Using Raspberry Pi. *IDOSR Journal of Science and Technology* 7(1):53-66.\
8. Masisani William Mufana and Adabara Ibrahim (2022). Implementation of Smart Grid Decision Support Systems. *IDOSR Journal of Scientific Research* 7(1) 50-57, 2022.
9. Natumanya Akimu (2022). Design and Construction of an Automatic Load Monitoring System on a Transformer in Power Distribution Networks. *IDOSR Journal of Scientific Research* 7(1) 58-76, 2022.