



# Traffic Engineering

2nd Lab Project Report

## MPLS and VPN-IP over MPLS

Group 8

Francisco Bento - 93581

Pedro Tracana - 93610

May 19, 2024

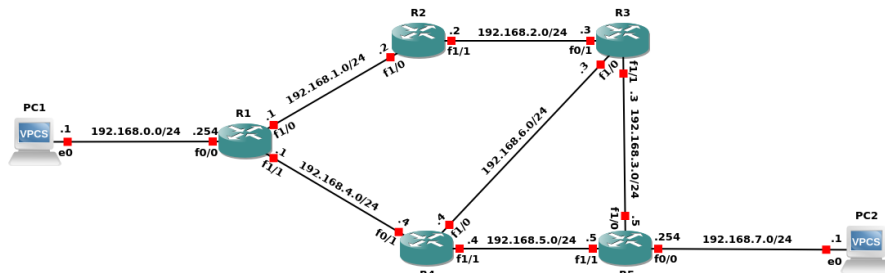
# 1 Introduction

The aim of this report is to implement MPLS and VPN-IP over MPLS using two distinct network topologies, both simulated in GNS3. This report provides detailed configuration examples applied to Cisco IOS devices (Cisco 7200), along with graphical representations of the network topologies. Furthermore, it includes a thorough validation of the commands through Wireshark captures, accompanied by comprehensive explanations that demonstrate the correct behavior and functionality of both MPLS and VPN-IP over MPLS networks.

## 2 Network Topology and Configuration

### 2.1 Part A and B - MPLS

To begin, Parts A and B of the second laboratory were treated as a single unit, and will be reported as one network topology used to simulate an MPLS network. The network topology is shown in Figure 1.



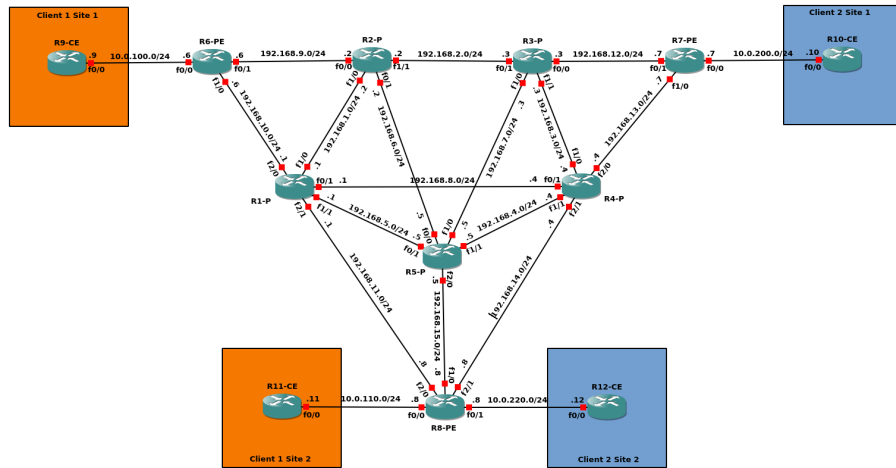
**Figure 1:** Network Topology for Part A and B.

Since the goal of this exercise is to simulate an MPLS network, it is essential to provide the configurations applied to the VPCSs and Cisco routers. These configurations are detailed in Annex A. The key steps to set up an MPLS network are as follows:

1. Have an established IGP protocol, such as OSPF.
2. Enable MPLS globally by using the command *mpls ip*.
3. Configure the protocol for MPLS label distribution and mapping, which is LDP, as it is widely used in MPLS networks. This is done by using the command *mpls label protocol ldp*.
4. Enable MPLS locally on the interfaces that are turned towards the core of the network by using the command *mpls ip* within the scope of the respective interfaces.

## 2.2 Part C - VPN-IP over MPLS

The purpose of this exercise is to simulate an MPLS network, as illustrated in Figure 2, and to examine how VPN-IP over MPLS functions to connect the sites.



**Figure 2:** Network Topology for Part C.

Since the goal of this exercise is to simulate MPLS IP VPN, it is essential to provide the configurations to the Cisco routers. This includes the Customer Edge (CE) routers at different sites, the Provider Edge (PE) routers, and the Provider (P) routers. The keys steps to set up the VPN-IP's over MPLS are as follows:

1. Have an established IGP protocol such as OSPF between all nodes P and PE nodes within the same process ID (1).
2. Enable MPLS globally by using the command *mpls ip* on all P and PE routers.
3. Configure the protocol for MPLS label distribution and mapping, which is LDP, as it is widely used in MPLS networks. This is done by using the command *mpls label protocol ldp*.
4. Enable MPLS locally on the interfaces that are turned towards the core of the network by using the command *mpls ip* within the scope of the respective interfaces.
5. Now, configure MP-BGP between all PE routers so that these LERs can exchange VPN-related routing information. This is done by configuring *router bgp 1* and specifying the neighbors for each PE router using commands *neighbor X.X.X.X remote-as 1*, *neighbor X.X.X.X update-source*

*Lo0*, and defining the address families to be shared with *address-family vpnv4*.

6. Establish two additional OSPF process IDs, each associated with a specific Virtual Routing and Forwarding (VRF) entity, one for each customer (using the same process ID for both sites of a customer). This setup enables communication between the CEs and the PEs, which is essential for the next step: configuring the VRF tables on the PEs.
7. Create the VRFs that enable multiple instances of routing tables to coexist without interference (hence the different process IDs), allowing different sites of the same customer to communicate remotely with each other. When configuring R8 to handle two VPNs simultaneously, as detailed in B, the following steps are necessary for setting up each VRF:
  - Create the VRFs using the command *ip vrf Customer\_X*, and within this scope, define the route distinguisher (RD) and the route target (RT). The RD, specified with *rd X:Y*, creates a unique VPNv4 address by combining the AS number with the customer number. The RT, configured with *route-target both W:Z*, controls the import and export of routes between different VPNs.
  - The next step is to activate the respective VRF forwarding for each PE interface that connects to CEs. In the case of R8, interface Fa0/0 should activate *ip vrf forwarding Customer\_1*, while interface Fa0/1 should activate *ip vrf forwarding Customer\_2*.
  - Final step is to ensure that customer sites have end-to-end connectivity. For R8, it's required to inject BGP routes into each OSPF routing table (*router ospf 2 vrf Customer\_1* and *router ospf 3 Customer\_2*) with the command *redistribute bgp 1 subnets*. On the other hand, it's also necessary to inject the OSPF routes into the BGP routing table. To achieve this, under both address-families *ipv4 vrf Customer\_1* and *ipv4 vrf Customer\_2*, each OSPF has to be redistributed with *redistribute ospf 2* and *redistribute ospf 3* respectively.

All related configurations are detailed in Annex B.

## 3 Experimental Results

### 3.1 Part A and B

To validate the MPLS implementation, it is necessary to ping between PC1 and PC2 and verify that they can indeed communicate with each other. However, it is also necessary to answer the question of what the Label Switched Path (LSP) looks like.

To address this question, the following commands should be used to understand the overall behavior of both OSPF (an Interior Gateway Protocol) and MPLS with LDP:

- Routing table (RT) - *show ip route*.
- Forwarding Information Base (FIB) - *show ip cef*.
- Label Information Base (LIB) - *show mpls ldp bindings*.
- Label Forwarding Information Base (LFIB) - *show mpls forwarding-table*.

When R1 receives a route from an OSPF neighbor, it stores it in the routing table (RT). This routing table is part of the control plane, so it does not make forwarding decisions. To forward traffic, it is necessary to parse the routing table and build what is known as the Forwarding Information Base (FIB).

The same process occurs with MPLS. The association of IP address prefixes to label bindings is built and stored in the Label Information Base (LIB), which is then used to create the Label Forwarding Information Base (LFIB) on the data/forwarding plane.

Now, when an incoming packet arrives, it is checked to determine if the packet is labeled or not. If it is labeled, it should be processed by the LFIB; otherwise, the FIB performs the lookup. In our scenario, it is expected that the lookup is performed against the FIB on the ingress LSR (R1). In the core, the lookup is performed against the LFIB, and on the egress router (R5), the lookup is performed against the FIB if there is only a single label in the stack that was previously popped by Penultimate Hop Popping (PHP). Otherwise, the LFIB is looked up.

```

R1#show ip route
Codes: L - local, 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, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

    192.168.0.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.0.0/24 is directly connected, FastEthernet0/0
L       192.168.0.254/32 is directly connected, FastEthernet0/0
    192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/24 is directly connected, FastEthernet1/0
L       192.168.1.1/32 is directly connected, FastEthernet1/0
O       192.168.2.0/24 [110/2] via 192.168.1.2, 00:54:04, FastEthernet1/0
O       192.168.3.0/24 [110/3] via 192.168.4.4, 00:54:14, FastEthernet1/1
        [110/3] via 192.168.1.2, 00:54:04, FastEthernet1/0
    192.168.4.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.4.0/24 is directly connected, FastEthernet1/1
L       192.168.4.1/32 is directly connected, FastEthernet1/1
O       192.168.5.0/24 [110/2] via 192.168.4.4, 00:54:04, FastEthernet1/1
O       192.168.6.0/24 [110/2] via 192.168.4.4, 00:54:14, FastEthernet1/1
O       192.168.7.0/24 [110/3] via 192.168.4.4, 00:54:04, FastEthernet1/1

```

**Figure 3:** Routing Table of R1.

To begin, let's verify the routing table (RT) of R1, as shown in Figure 3. It indicates that to reach the subnet 192.168.7.0/24 (where PC2 is located), the next hop must be 192.168.4.4 (the R4 interface on the subnet 192.168.4.0/24) via interface FastEthernet1/1.

Additionally, when examining the LIB table of R1 in Figure 4, it shows that R1 has assigned the local binding label value 19 for this specific path (value to be used by other LSR's on the upstream side). The next-hop label value through the next LSR with the interface 192.168.6.4 should also be 19, which is the label included in the MPLS packet.

```
R1#show mpls ldp bindings
lib entry: 192.168.0.0/24, rev 7
  local binding: label: imp-null
  remote binding: lsr: 192.168.6.4:0, label: 21
  remote binding: lsr: 192.168.2.2:0, label: 19
lib entry: 192.168.1.0/24, rev 8
  local binding: label: imp-null
  remote binding: lsr: 192.168.6.4:0, label: 20
  remote binding: lsr: 192.168.2.2:0, label: imp-null
lib entry: 192.168.2.0/24, rev 15
  local binding: label: 21
  remote binding: lsr: 192.168.6.4:0, label: 22
  remote binding: lsr: 192.168.2.2:0, label: imp-null
lib entry: 192.168.3.0/24, rev 17
  local binding: label: 22
  remote binding: lsr: 192.168.6.4:0, label: 23
  remote binding: lsr: 192.168.2.2:0, label: 21
lib entry: 192.168.4.0/24, rev 9
  local binding: label: imp-null
  remote binding: lsr: 192.168.6.4:0, label: imp-null
  remote binding: lsr: 192.168.2.2:0, label: 23
lib entry: 192.168.5.0/24, rev 13
  local binding: label: 20
  remote binding: lsr: 192.168.6.4:0, label: imp-null
  remote binding: lsr: 192.168.2.2:0, label: 20
lib entry: 192.168.6.0/24, rev 19
  local binding: label: 23
  remote binding: lsr: 192.168.6.4:0, label: imp-null
  remote binding: lsr: 192.168.2.2:0, label: 22
lib entry: 192.168.7.0/24, rev 11
  local binding: label: 19
  remote binding: lsr: 192.168.6.4:0, label: 19
  remote binding: lsr: 192.168.2.2:0, label: 18
```

**Figure 4:** Label Information Base of R1.

This information is useful because on the link between PC1 and R1, shown in Figure 5, there is no MPLS traffic (only IP traffic). R1 will act as the ingress router and will be responsible for labeling the packet.

No.	Time	Source	Destination	Protocol	Length	Info
16	52.439024	192.168.0.1	192.168.7.1	ICMP	98	Echo (ping) request id=0x1ab1, seq=1/256, ttl=64 (reply in 17)
17	52.529537	192.168.7.1	192.168.0.1	ICMP	98	Echo (ping) reply id=0x1ab1, seq=1/256, ttl=61 (request in 16)
18	53.531025	192.168.0.1	192.168.7.1	ICMP	98	Echo (ping) request id=0x1bb1, seq=2/512, ttl=64 (reply in 19)
19	53.563594	192.168.7.1	192.168.0.1	ICMP	98	Echo (ping) reply id=0x1bb1, seq=2/512, ttl=61 (request in 18)
20	54.564730	192.168.0.1	192.168.7.1	ICMP	98	Echo (ping) request id=0x1cb1, seq=3/768, ttl=64 (reply in 21)
21	54.601115	192.168.7.1	192.168.0.1	ICMP	98	Echo (ping) reply id=0x1cb1, seq=3/768, ttl=61 (request in 20)
22	55.601744	192.168.0.1	192.168.7.1	ICMP	98	Echo (ping) request id=0x1db1, seq=4/1024, ttl=64 (reply in 23)
23	55.650186	192.168.7.1	192.168.0.1	ICMP	98	Echo (ping) reply id=0x1db1, seq=4/1024, ttl=61 (request in 22)
24	56.651126	192.168.0.1	192.168.7.1	ICMP	98	Echo (ping) request id=0x1eb1, seq=5/1280, ttl=64 (reply in 25)
25	56.689007	192.168.7.1	192.168.0.1	ICMP	98	Echo (ping) reply id=0x1eb1, seq=5/1280, ttl=61 (request in 24)

**Figure 5:** Wireshark capture between PC1 and R1.

The incoming packet will be processed by the FIB, as shown in Figure 6. Upon lookup, the FIB retrieves the next-hop label from the LIB and labels the packet with label 19 to reach R4.

```
R1#show ip cef
```

Prefix	Next Hop	Interface
0.0.0.0/0	no route	
0.0.0.0/8	drop	
0.0.0.0/32	receive	
127.0.0.0/8	drop	
192.168.0.0/24	attached	FastEthernet0/0
192.168.0.0/32	receive	FastEthernet0/0
192.168.0.254/32	receive	FastEthernet0/0
192.168.0.255/32	receive	FastEthernet0/0
192.168.1.0/24	attached	FastEthernet1/0
192.168.1.0/32	receive	FastEthernet1/0
192.168.1.1/32	receive	FastEthernet1/0
192.168.1.2/32	attached	FastEthernet1/0
192.168.1.255/32	receive	FastEthernet1/0
192.168.2.0/24	192.168.1.2	FastEthernet1/0
192.168.3.0/24	192.168.1.2	FastEthernet1/0
	192.168.4.4	FastEthernet1/1
192.168.4.0/24	attached	FastEthernet1/1
192.168.4.0/32	receive	FastEthernet1/1
192.168.4.1/32	receive	FastEthernet1/1
192.168.4.4/32	attached	FastEthernet1/1
192.168.4.255/32	receive	FastEthernet1/1
192.168.5.0/24	192.168.4.4	FastEthernet1/1
192.168.6.0/24	192.168.4.4	FastEthernet1/1
192.168.7.0/24	192.168.4.4	FastEthernet1/1
224.0.0.0/4	drop	
224.0.0.0/24	receive	
240.0.0.0/4	drop	
255.255.255.255/32	receive	

**Figure 6:** Forwarding Information Base of R1.

This labeling can be verified in Figure 7, which provides a sample of the ICMP traffic on the R1-R4 link. In this figure, we can dissect the MPLS shim header with the following values:

1. MPLS Label: 19.
2. MPLS Experimental Bits: 0 (there is no traffic classification).



3. MPLS Bottom of Label Stack: 1 (enabled as there is only one layer of labels in this exercise).
4. MPLS TTL: 63 (number of next-hops that can be done within the MPLS network).

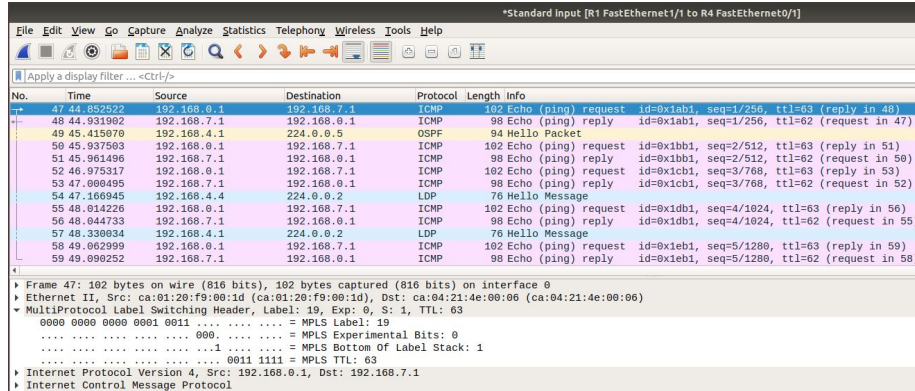


Figure 7: Wireshark capture between R1 and R4.

Upon arrival at router R4, it will function as a Label Switching Router (LSR) and merely switch the label according to its own LFIB table, depicted in Figure 8. According to the table, an incoming packet with label 19 (local label) targeting the prefix 192.168.7.0/24 should be sent out with an outgoing label set to "Pop Label" via the outgoing interface FastEthernet1/1 to reach the next hop 192.168.5.5.

```
R4#show mpls forwarding-table
```

Local Label	Outgoing Label	Prefix or Tunnel Id	Bytes Label Switched	Outgoing interface	Next Hop
19	Pop Label	192.168.7.0/24	490	Fa1/1	192.168.5.5
20	Pop Label	192.168.1.0/24	0	Fa0/1	192.168.4.1
21	Pop Label	192.168.0.0/24	490	Fa0/1	192.168.4.1
22	Pop Label	192.168.2.0/24	656	Fa1/0	192.168.6.3
23	Pop Label	192.168.3.0/24	0	Fa1/1	192.168.5.5
	Pop Label	192.168.3.0/24	0	Fa1/0	192.168.6.3

Figure 8: Label Forwarding Information Base of R4.

Setting the outgoing label to "Pop Label" indicates that the router will not switch labels; instead, it will remove the MPLS label and header from the packet. This operation is commonly used on an egress router (although not applicable to R4) to forward the packet based on its destination IP address.

It is also utilized when implementing Penultimate Hop Popping (PHP) to prevent a double lookup scenario, where the next hop would otherwise need to perform lookups in both the FIB and LFIB.



The result of performing PHP can be seen in Figure 9, where the packet is forwarded on plain IP addresses, without MPLS. Therefore, when it reaches router R5 the lookup is only done on the R5's FIB, shown in Figure 10.

\*Standard Input [R5 FastEthernet1/1 to R4 FastEthernet1/1]

No.	Time	Source	Destination	Protocol	Length	Info
37	32.522178	192.168.0.1	192.168.7.1	ICMP	98	Echo (ping) request id=0x1ab1, seq=1/256, ttl=62 (reply in 38)
38	32.593114	192.168.7.1	192.168.0.1	ICMP	102	Echo (ping) reply id=0x1ab1, seq=1/256, ttl=63 (request in 37)
39	33.603030	192.168.0.1	192.168.7.1	ICMP	98	Echo (ping) request id=0x1bb1, seq=2/512, ttl=62 (reply in 40)
40	33.617108	192.168.7.1	192.168.0.1	ICMP	102	Echo (ping) reply id=0x1bb1, seq=2/512, ttl=63 (request in 39)
41	34.118673	192.168.5.5	224.0.0.2	LDP	76	Hello Message
42	34.641919	192.168.0.1	192.168.7.1	ICMP	98	Echo (ping) request id=0x1cb1, seq=3/768, ttl=62 (reply in 43)
43	34.659582	192.168.7.1	192.168.0.1	ICMP	102	Echo (ping) reply id=0x1cb1, seq=3/768, ttl=63 (request in 42)
44	34.672479	192.168.5.4	224.0.0.2	LDP	76	Hello Message
45	35.686378	192.168.0.1	192.168.7.1	ICMP	98	Echo (ping) request id=0x1db1, seq=4/1024, ttl=62 (reply in 46)
46	35.701547	192.168.7.1	192.168.0.1	ICMP	102	Echo (ping) reply id=0x1db1, seq=4/1024, ttl=63 (request in 45)
47	36.731815	192.168.0.1	192.168.7.1	ICMP	98	Echo (ping) request id=0x1eb1, seq=5/1280, ttl=62 (reply in 48)
48	36.747858	192.168.7.1	192.168.0.1	ICMP	102	Echo (ping) reply id=0x1eb1, seq=5/1280, ttl=63 (request in 47)

▶ Frame 37: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0  
 ▶ Ethernet II, Src: ca:04:21:4e:00:1d (ca:04:21:4e:00:1d), Dst: ca:05:21:6b:00:1d (ca:05:21:6b:00:1d)  
 ▶ Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.7.1  
 ▶ Internet Control Message Protocol

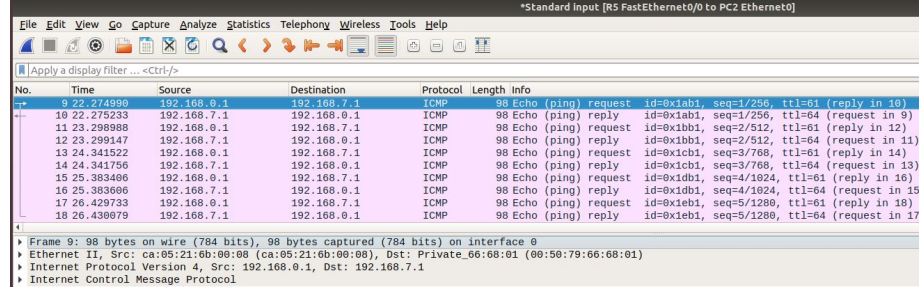
Figure 9: Wireshark capture between R4 and R5.

```
R5#show ip cef
```

Prefix	Next Hop	Interface
0.0.0.0/0	no route	
0.0.0.0/8	drop	
0.0.0.0/32	receive	
127.0.0.0/8	drop	
192.168.0.0/24	192.168.5.4	FastEthernet1/1
192.168.1.0/24	192.168.3.3	FastEthernet1/0
	192.168.5.4	FastEthernet1/1
192.168.2.0/24	192.168.3.3	FastEthernet1/0
192.168.3.0/24	attached	FastEthernet1/0
192.168.3.0/32	receive	FastEthernet1/0
192.168.3.3/32	attached	FastEthernet1/0
192.168.3.5/32	receive	FastEthernet1/0
192.168.3.255/32	receive	FastEthernet1/0
192.168.4.0/24	192.168.5.4	FastEthernet1/1
192.168.5.0/24	attached	FastEthernet1/1
192.168.5.0/32	receive	FastEthernet1/1
192.168.5.4/32	attached	FastEthernet1/1
192.168.5.5/32	receive	FastEthernet1/1
192.168.5.255/32	receive	FastEthernet1/1
192.168.6.0/24	192.168.3.3	FastEthernet1/0
	192.168.5.4	FastEthernet1/1
192.168.7.0/24	attached	FastEthernet0/0
192.168.7.0/32	receive	FastEthernet0/0
192.168.7.1/32	attached	FastEthernet0/0
192.168.7.254/32	receive	FastEthernet0/0
192.168.7.255/32	receive	FastEthernet0/0
224.0.0.0/4	drop	
224.0.0.0/24	receive	
240.0.0.0/4	drop	
255.255.255.255/32	receive	

Figure 10: Forwarding Information Base of R5.

More precisely, when the next-hop is designated as "attached," it means that a route is directly connected, enabling a router to forward packets for that network without executing a complete routing table lookup. This optimization is utilized to expedite packet forwarding for locally connected networks, as shown in Figure 11.



No.	Time	Source	Destination	Protocol	Length	Info
9	22.274990	192.168.0.1	192.168.7.1	ICMP	98	Echo (ping) request id=0x1ab1, seq=1/256, ttl=61 (reply in 10)
10	22.275233	192.168.7.1	192.168.0.1	ICMP	98	Echo (ping) reply id=0x1ab1, seq=1/256, ttl=64 (request in 9)
11	23.290988	192.168.0.1	192.168.7.1	ICMP	98	Echo (ping) request id=0x1bb1, seq=2/512, ttl=61 (reply in 12)
12	23.299147	192.168.7.1	192.168.0.1	ICMP	98	Echo (ping) reply id=0x1bb1, seq=2/512, ttl=64 (request in 11)
13	24.341522	192.168.0.1	192.168.7.1	ICMP	98	Echo (ping) request id=0x1cb1, seq=3/768, ttl=61 (reply in 14)
14	24.341756	192.168.7.1	192.168.0.1	ICMP	98	Echo (ping) reply id=0x1cb1, seq=3/768, ttl=64 (request in 13)
15	25.383486	192.168.0.1	192.168.7.1	ICMP	98	Echo (ping) request id=0x1db1, seq=4/1024, ttl=61 (reply in 16)
16	25.383686	192.168.7.1	192.168.0.1	ICMP	98	Echo (ping) reply id=0x1db1, seq=4/1024, ttl=64 (request in 15)
17	26.429733	192.168.0.1	192.168.7.1	ICMP	98	Echo (ping) request id=0x1eb1, seq=5/1280, ttl=61 (reply in 18)
18	26.430079	192.168.7.1	192.168.0.1	ICMP	98	Echo (ping) reply id=0x1eb1, seq=5/1280, ttl=64 (request in 17)

Figure 11: Wireshark capture between R5 and PC2.

So, when identifying the default Label Switch Path (LSP), it can be deduced that from PC1 to PC2, the LSRs used for this unidirectional path are R1, followed by R4, and finally R5.

Interestingly, the ICMP Echo replies to the ICMP Echo requests follow the same path in terms of LSRs (due to the topology); however, it shouldn't be assumed that it is the same LSP, as the label bindings are different in the opposite direction.

Additionally, the behavior of having only one link displaying MPLS remains consistent, although it applies to the R5 to R4 link when considering the opposite direction, due to PHP from R4 to R1. Apart from these two details, the analysis is considerably identical.

Next, we examined the network's response by simulating a failure in one of the links within the default LSP from PC1 to PC2. Specifically, we targeted the link between R1 and R4, as illustrated in Figure 12.

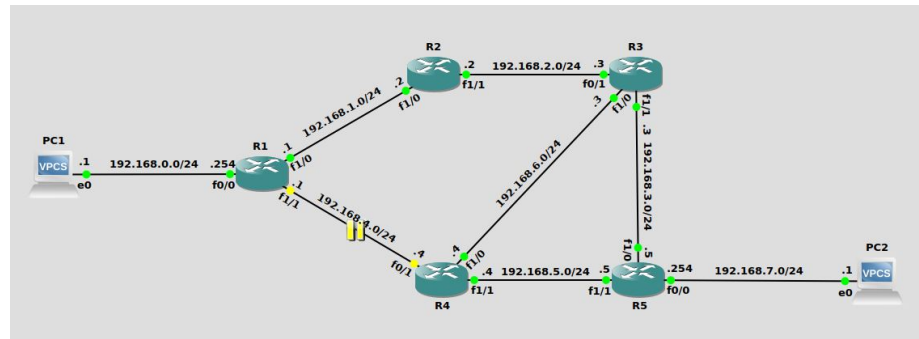


Figure 12: Suspending link between R1 and R4.

Following the concepts mentioned above, let's evaluate what has changed in the behaviour of OSPF, routing tables and MPLS.

Starting with R1, it is worth to take a look into R1's new routing table and LIB table. The routing table now presents a different next-hop IP address, as shown in Figure 13.

```
R1#show ip route
Codes: L - local, 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, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

    192.168.0.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.0.0/24 is directly connected, FastEthernet0/0
L       192.168.0.254/32 is directly connected, FastEthernet0/0
    192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/24 is directly connected, FastEthernet1/0
L       192.168.1.1/32 is directly connected, FastEthernet1/0
O       192.168.2.0/24 [110/2] via 192.168.1.2, 05:30:13, FastEthernet1/0
O       192.168.3.0/24 [110/3] via 192.168.1.2, 05:30:13, FastEthernet1/0
    192.168.4.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.4.0/24 is directly connected, FastEthernet1/1
L       192.168.4.1/32 is directly connected, FastEthernet1/1
O       192.168.5.0/24 [110/4] via 192.168.1.2, 00:10:22, FastEthernet1/0
O       192.168.6.0/24 [110/3] via 192.168.1.2, 00:10:22, FastEthernet1/0
O       192.168.7.0/24 [110/4] via 192.168.1.2, 00:10:22, FastEthernet1/0
```

**Figure 13:** New routing table of R1.

Furthermore, the LDP label mappings have been updated. Each LIB entry associated with a specific prefix has now generated another LSP path, resulting in the creation of additional labels with different values to accommodate the new path.

In this scenario, when a packet arrives at R1 destined for subnet 192.168.7.0/24, it is assigned an outgoing label with the value 18, corresponding to the local label of the interface 192.168.2.2 of R2, as illustrated in Figure 14.

This labeling can be confirmed in Figure 15, which presents a sample of the ICMP traffic on the R1-R2 link. In this figure, we can analyze the MPLS shim header with the following values:

- MPLS Label: 18.
- MPLS Experimental Bits: 0.
- MPLS Bottom of Label Stack: 1.
- MPLS TTL: 63.

```

R1#show mpls ldp bindings
lib entry: 192.168.0.0/24, rev 7
  local binding: label: imp-null
  remote binding: lsr: 192.168.2.2:0, label: 19
lib entry: 192.168.1.0/24, rev 8
  local binding: label: imp-null
  remote binding: lsr: 192.168.2.2:0, label: imp-null
lib entry: 192.168.2.0/24, rev 15
  local binding: label: 21
  remote binding: lsr: 192.168.2.2:0, label: imp-null
lib entry: 192.168.3.0/24, rev 17
  local binding: label: 22
  remote binding: lsr: 192.168.2.2:0, label: 21
lib entry: 192.168.4.0/24, rev 9
  local binding: label: imp-null
  remote binding: lsr: 192.168.2.2:0, label: 23
lib entry: 192.168.5.0/24, rev 13
  local binding: label: 20
  remote binding: lsr: 192.168.2.2:0, label: 20
lib entry: 192.168.6.0/24, rev 19
  local binding: label: 23
  remote binding: lsr: 192.168.2.2:0, label: 22
lib entry: 192.168.7.0/24, rev 11
  local binding: label: 19
  remote binding: lsr: 192.168.2.2:0, label: 18

```

Figure 14: New LIB table of R1.

No.	Time	Source	Destination	Protocol	Length	Info
48	47.373455	192.168.0.1	192.168.7.1	ICMP	102	Echo (ping) request id=0x90e0, seq=1/256, ttl=63 (reply in 48)
49	47.437960	192.168.7.1	192.168.0.1	ICMP	98	Echo (ping) reply id=0x90e0, seq=1/256, ttl=63 (request in 48)
50	48.454148	192.168.0.1	192.168.7.1	ICMP	102	Echo (ping) request id=0x90e0, seq=2/512, ttl=63 (reply in 51)
51	48.497723	192.168.7.1	192.168.0.1	ICMP	98	Echo (ping) reply id=0x90e0, seq=2/512, ttl=63 (request in 50)
52	48.525021	192.168.1.1	224.0.0.5	OSPF	94	Hello Packet
53	48.681384	192.168.1.2	224.0.0.2	LDP	76	Hello Message
54	49.508066	192.168.0.1	192.168.7.1	ICMP	102	Echo (ping) request id=0x90e0, seq=3/768, ttl=63 (reply in 55)
55	49.550889	192.168.7.1	192.168.0.1	ICMP	98	Echo (ping) reply id=0x90e0, seq=3/768, ttl=63 (request in 54)
56	49.750083	192.168.1.1	224.0.0.2	LDP	76	Hello Message
57	50.414059	ca:01:20:f9:00:1c	ca:01:20:f9:00:1c	LOOP	60	Reply

Frame 48: 192 bytes on wire (816 bits), 192 bytes captured (816 bits) on interface 0  
 Ethernet II, Src: ca:01:20:f9:00:1c (ca:01:20:f9:00:1c), Dst: ca:02:21:16:00:1c (ca:02:21:16:00:1c)  
 MultiProtocol Label Switching Header, Label: 18, Exp: 0, S: 1, TTL: 63  
 0000 0000 0000 0010 ..... = MPLS Label: 18  
 ..... 0000 ..... = MPLS Experimental Bits: 0  
 ..... 0001 ..... = MPLS Bottom Of Label Stack: 1  
 ..... 0011 1111 = MPLS TTL: 63  
 Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.7.1  
 Internet Control Message Protocol

Figure 15: Wireshark capture between R1 and R2 with suspended link.

Upon reaching router R2, it performs a simple label switch based on its LFIB table, as shown in Figure 16. According to the table, an incoming packet with label 18 should have its label swapped to 19, enabling routing towards subnet 192.168.7.0/24 via R2's FastEthernet1/1 interface with the next hop being 192.168.2.3.



```
R2#show mpls forwarding-table
```

Local Label	Outgoing Label	Prefix or Tunnel Id	Bytes Switched	Label	Outgoing interface	Next Hop
18	19	192.168.7.0/24	0		Fa1/1	192.168.2.3
19	Pop Label	192.168.0.0/24	0		Fa1/0	192.168.1.1
20		192.168.5.0/24	0		Fa1/1	192.168.2.3
21	Pop Label	192.168.3.0/24	0		Fa1/1	192.168.2.3
22	Pop Label	192.168.6.0/24	1404		Fa1/1	192.168.2.3
23	Pop Label	192.168.4.0/24	0		Fa1/0	192.168.1.1

```
R2#
```

**Figure 16:** LFIB table of R2.

This label swap can be confirmed in Figure 17, which presents a sample of the ICMP traffic on the R2-R3 link. In this figure, we can observe alterations in two fields of the MPLS shim header:

- MPLS Label: 19.
- TTL: 62 (This value decreases by one for each LSR that it hops through using MPLS).

No.	Time	Source	Destination	Protocol	Length	Info
39	38.942888	192.168.0.1	192.168.7.1	ICMP	102	Echo (ping) request id=0x95e0, seq=1/256, ttl=63 (reply in 40)
40	38.997125	192.168.7.1	192.168.0.1	ICMP	102	Echo (ping) reply id=0x95e0, seq=1/256, ttl=63 (request in 39)
41	39.365912	192.168.2.2	192.168.6.3	LDP	72	Keep Alive Message
42	39.506368	192.168.6.3	192.168.2.2	TCP	60	11300 → 646 [ACK] Seq=19 Ack=19 Win=3804 Len=0
43	40.023712	192.168.0.1	192.168.7.1	ICMP	102	Echo (ping) request id=0x96e0, seq=2/512, ttl=63 (reply in 44)
44	40.054834	192.168.7.1	192.168.0.1	ICMP	102	Echo (ping) reply id=0x96e0, seq=2/512, ttl=63 (request in 43)
45	41.086099	192.168.0.1	192.168.7.1	ICMP	102	Echo (ping) request id=0x97e0, seq=3/768, ttl=63 (reply in 46)
46	41.186964	192.168.7.1	192.168.0.1	ICMP	102	Echo (ping) reply id=0x97e0, seq=3/768, ttl=63 (request in 45)
47	41.729458	ca:03:21:32:00:06	ca:03:21:32:00:06	LOOP	60	Reply
48	41.760152	192.168.2.3	224.0.0.2	LDP	76	Hello Message
49	42.134539	192.168.0.1	192.168.7.1	ICMP	102	Echo (ping) request id=0x98e0, seq=4/1024, ttl=63 (reply in 50)
50	42.159593	192.168.7.1	192.168.0.1	ICMP	102	Echo (ping) reply id=0x98e0, seq=4/1024, ttl=63 (request in 49)

▶ Frame 39: 102 bytes on wire (816 bits), 102 bytes captured (816 bits) on interface 0  
 ▶ Ethernet II, Src: ca:02:21:16:00:1d (ca:02:21:16:00:1d), Dst: ca:03:21:32:00:06 (ca:03:21:32:00:06)  
 ▶ MultiProtocol Label Switching Header, Label: 19, Exp: 0, S: 1, TTL: 62  
   0000 0000 0000 0011 .... = MPLS Label: 19  
   .... = MPLS Experimental Bits: 0  
   .... 1 .... = MPLS Bottom Of Label Stack: 1  
   .... 0011 1110 = MPLS TTL: 62  
 ▶ Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.7.1  
 ▶ Internet Control Message Protocol

**Figure 17:** Wireshark capture between R2 and R3 with suspended link.

Similar to the functionality described for R4 previously, R3 now has the same responsibility with R5. According to its LFIB table depicted in Figure 18, an incoming packet with label 19 should prompt R3 to perform PHP, enabling R5 to receive the forwarded packet without MPLS, as demonstrated in Figure 19.

```
R3#show mpls forwarding-table
```

Local Label	Outgoing Label	Prefix or Tunnel Id	Bytes Switched	Label	Outgoing interface	Next Hop
19	Pop Label	192.168.7.0/24	0	0	Fa1/1	192.168.3.5
20	Pop Label	192.168.5.0/24	0	0	Fa1/1	192.168.3.5
	Pop Label	192.168.5.0/24	0	0	Fa1/0	192.168.6.4
21	19	192.168.0.0/24	0	0	Fa0/1	192.168.2.2
22	Pop Label	192.168.4.0/24	0	0	Fa1/0	192.168.6.4
23	Pop Label	192.168.1.0/24	0	0	Fa0/1	192.168.2.2

```
R3#
```

Figure 18: LFIB table of R3.

No.	Time	Source	Destination	Protocol	Length	Info
33	29.395438	192.168.0.1	192.168.7.1	ICMP	60	Echo (ping) request id=0x9e0, seq=2/512, ttl=61 (reply in 34)
34	29.415826	192.168.7.1	192.168.0.1	ICMP	102	Echo (ping) reply id=0x9e0, seq=2/512, ttl=63 (request in 33)
35	30.367609	192.168.3.3	224.0.0.2	LDP	76	Hello Message
36	30.459747	192.168.0.1	192.168.7.1	ICMP	98	Echo (ping) request id=0x97e0, seq=3/768, ttl=61 (reply in 37)
37	30.479758	192.168.7.1	192.168.0.1	ICMP	102	Echo (ping) reply id=0x97e0, seq=3/768, ttl=63 (request in 36)
38	31.108437	ca:05:21:6b:00:1d	ca:03:21:32:00:1d	LOOP	60	Reply
39	31.510320	192.168.0.1	192.168.7.1	ICMP	98	Echo (ping) request id=0x9e0, seq=4/1024, ttl=61 (reply in 40)
40	31.528482	192.168.7.1	192.168.0.1	ICMP	102	Echo (ping) reply id=0x9e0, seq=4/1024, ttl=63 (request in 39)
41	31.538611	ca:05:21:6b:00:1c	ca:05:21:6b:00:1c	LOOP	60	Reply
42	32.405431	192.168.3.5	224.0.0.5	OSPF	94	Hello Packet
43	32.561814	192.168.0.1	192.168.7.1	ICMP	98	Echo (ping) request id=0x99e0, seq=5/1280, ttl=61 (reply in 44)
44	32.580298	192.168.7.1	192.168.0.1	ICMP	102	Echo (ping) reply id=0x99e0, seq=5/1280, ttl=63 (request in 43)

Frame 33: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0  
 Ethernet II, Src: ca:03:21:32:00:1d (ca:03:21:32:00:1d), Dst: ca:05:21:6b:00:1c (ca:05:21:6b:00:1c)  
 Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.7.1  
 Internet Control Message Protocol

Figure 19: Wireshark capture between R3 and R5 with suspended link.

Upon reaching R5, only a lookup with R5's FIB table is necessary. It indicates that for traffic destined for 192.168.7.0/24, packets should be expedited without a complete routing table lookup, as the subnet is directly connected to R5, as depicted in Figure 20.

No.	Time	Source	Destination	Protocol	Length	Info
18	18.381099	ca:05:21:6b:00:08	Private_66:68:01	ARP	60	192.168.7.254 is at ca:05:21:6b:00:08
19	18.382148	192.168.7.1	192.168.0.1	ICMP	98	Echo (ping) reply id=0x95e0, seq=1/256, ttl=64 (request in 7)
20	19.439342	192.168.0.1	192.168.7.1	ICMP	98	Echo (ping) request id=0x9e0, seq=2/512, ttl=60 (reply in 12)
21	19.439488	192.168.7.1	192.168.0.1	ICMP	98	Echo (ping) reply id=0x9e0, seq=2/512, ttl=64 (request in 11)
22	20.593230	192.168.0.1	192.168.7.1	ICMP	98	Echo (ping) request id=0x97e0, seq=3/768, ttl=60 (reply in 14)
23	20.593403	192.168.7.1	192.168.0.1	ICMP	98	Echo (ping) reply id=0x97e0, seq=3/768, ttl=64 (request in 13)
24	21.551912	192.168.0.1	192.168.7.1	ICMP	98	Echo (ping) request id=0x98e0, seq=4/1024, ttl=60 (reply in 16)
25	21.552285	192.168.7.1	192.168.0.1	ICMP	98	Echo (ping) reply id=0x98e0, seq=4/1024, ttl=64 (request in 15)
26	21.572284	ca:05:21:6b:00:08	ca:05:21:6b:00:08	LOOP	60	Reply
27	22.603885	192.168.0.1	192.168.7.1	ICMP	98	Echo (ping) request id=0x99e0, seq=5/1280, ttl=60 (reply in 19)
28	22.604034	192.168.7.1	192.168.0.1	ICMP	98	Echo (ping) reply id=0x99e0, seq=5/1280, ttl=64 (request in 18)
29	22.271133	192.168.7.254	224.0.0.5	OSPF	90	Hello Packet

Frame 18: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0  
 Ethernet II, Src: Private\_66:68:01 (00:50:79:66:68:01), Dst: ca:05:21:6b:00:08 (ca:05:21:6b:00:08)  
 Internet Protocol Version 4, Src: 192.168.7.1, Dst: 192.168.0.1  
 Internet Control Message Protocol

Figure 20: Wireshark capture between R5 and PC2 with suspended link.

This concludes the scenario in which a failure occurs in the default LSP path and the system successfully recovers and converges to a new LSP path using alternative routes. During this process, the network's routing tables and LIB/FIB/LFIB entries are updated accordingly.

### 3.2 Part C

To validate the VPN-IP over MPLS implementation, it is necessary to test connectivity by pinging between sites of the same customer, such as R9 to R11 and R10 to R12. Pinging between sites of different customers should fail, as they are on separate VPNs, reflecting a real-world scenario where different customers require private connections within the network.

In addition to the commands provided in Part A and B, it is crucial to verify that protocols such as MP-BGP are functioning correctly, each PE router maintains additional routing tables beyond the global routing table, and the CEs have routing information that enables access to other customer sites.

The following commands should be used to assess the overall behavior of VPN-IP over MPLS:

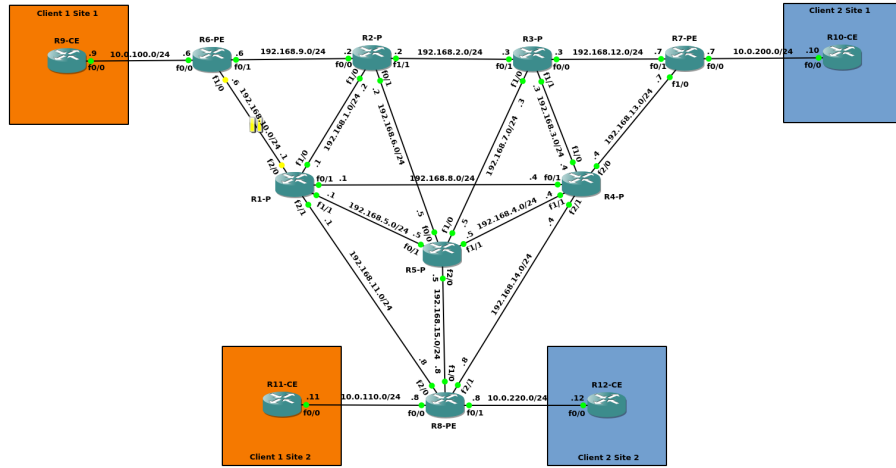
- Verifies BGP neighbors - *show bgp vpnv4 unicast all summary* on any PE router.
- Displays all routes from various routing protocols within the VRF - *show ip route vrf Customer\_X*, being X dependant on which PE router the command is being executed and which CE routers are connected to it.
- Displays BGP-specific routes in the VPNv4 address family for the specified VRF - *show ip bgp vpnv4 vrf Customer\_X*, being X dependant on which PE router the command is being executed and which CE routers are connected to it.
- Displays CE routes - *show ip route* in any of the customers edge routers to guarantee that it detects both endpoint sites after the redistribution step.
- Display VPN labels associated with VPNv4 addresses - *show ip bgp vpnv4 all label* on any PE router.

In this scenario, we will introduce a link failure to create a more complex Label Switch Path (LSP), as show in Figure 21. This ensures that the default LSP involves multiple hops rather than just a single hop through an LSR (P router) performing Penultimate Hop Popping (PHP).

By doing this, we will observe the use of a label stack, consisting of both the IGP label and the VPN label, rather than just seeing a single label. This setup will provide a more realistic and comprehensive demonstration of MPLS behavior in a network with redundancy and multiple routing paths.

Specifically, we have chosen to temporarily suspend the link between R1 and R6, allowing the network to adapt and converge to an alternative LSP path. The subsequent analysis will be based on this adjusted network configuration.





**Figure 21:** Suspending link between R1 and R6.

Starting with Customer Edge R9, we can start by visualizing what routes does it have after the system is up and running. Figure 22 displays the subnets that customer 1 on site 1 can access after the VPN-IP over MPLS is applied.

```
R9#show ip route
Codes: L - local, 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, H - NHRP, l - LISP
+ - replicated route, % - next hop override

Gateway of last resort is not set

  9.0.0.0/32 is subnetted, 1 subnets
C      9.9.9.9 is directly connected, Loopback0
 10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C      10.0.100.0/24 is directly connected, FastEthernet0/0
L      10.0.100.9/32 is directly connected, FastEthernet0/0
O IA   10.0.110.0/24 [110/2] via 10.0.100.6, 02:02:51, FastEthernet0/0
 11.0.0.0/32 is subnetted, 1 subnets
O IA   11.11.11.11 [110/3] via 10.0.100.6, 02:02:51, FastEthernet0/0
R9#
```

**Figure 22:** Routing table of R9.

As observed, R9 has access to both the subnet connecting R8 and R11 and R11's loopback interface. This means it can reach either of them, as illustrated in Figure 23. However, pinging other customer's site leads no to avail, as depicted in Figure 24.

```

R9#ping 10.0.110.11
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.110.11, timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 48/60/80 ms

```

**Figure 23:** Ping from R9 to R11.

```

R9#ping 10.0.220.12
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.220.12, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

```

**Figure 24:** Ping from R9 to R12.

The next question is, what LSP path is chosen when communicating from R9 to R11 ? This can be determined by using the command *trace 10.0.110.11* (Figure 25), which reveals the following path:

- R9 ->R6
- R6 ->R2
- R2 ->R1
- R1 ->R8
- R8 ->R11

```

R9#trace 10.0.110.11
Type escape sequence to abort.
Tracing the route to 10.0.110.11
VRF info: (vrf in name/id, vrf out name/id)
 0  10.0.100.6 48 msec 40 msec 4 msec
 1  192.168.9.2 [MPLS: Labels 23/33 Exp 0] 68 msec 60 msec 72 msec
 2  192.168.1.1 [MPLS: Labels 21/33 Exp 0] 60 msec 68 msec 48 msec
 3  10.0.110.8 64 msec 72 msec 92 msec
 4  10.0.110.11 80 msec 100 msec 88 msec

```

**Figure 25:** Trace from R9 to R11.

Before examining the Wireshark captures, let's review some of the tables that can be extracted from R6 using the commands mentioned earlier in this section.

R6's global routing table, shown in Figure 26, includes all the subnets used within the MPLS network. Additionally, it contains the loopback interfaces of all the Provider Edge routers that were used to set up MP-BGP.

```

R6#show ip route
Codes: L - local, 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, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

  6.0.0.0/32 is subnetted, 1 subnets
C    6.6.6.6 is directly connected, Loopback0
  7.0.0.0/32 is subnetted, 1 subnets
O    7.7.7.7 [110/4] via 192.168.9.2, 02:08:22, FastEthernet0/1
  8.0.0.0/32 is subnetted, 1 subnets
O    8.8.8.8 [110/4] via 192.168.9.2, 00:29:27, FastEthernet0/1
O    192.168.1.0/24 [110/2] via 192.168.9.2, 02:08:33, FastEthernet0/1
O    192.168.2.0/24 [110/2] via 192.168.9.2, 02:08:33, FastEthernet0/1
O    192.168.3.0/24 [110/3] via 192.168.9.2, 02:08:33, FastEthernet0/1
O    192.168.4.0/24 [110/3] via 192.168.9.2, 02:08:22, FastEthernet0/1
O    192.168.5.0/24 [110/3] via 192.168.9.2, 00:29:27, FastEthernet0/1
O    192.168.6.0/24 [110/2] via 192.168.9.2, 02:08:33, FastEthernet0/1
O    192.168.7.0/24 [110/3] via 192.168.9.2, 02:08:33, FastEthernet0/1
O    192.168.8.0/24 [110/3] via 192.168.9.2, 00:29:27, FastEthernet0/1
O    192.168.9.0/24 is variably subnetted, 2 subnets, 2 masks
C    192.168.9.0/24 is directly connected, FastEthernet0/1
L    192.168.9.6/32 is directly connected, FastEthernet0/1
O    192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
C    192.168.10.0/24 is directly connected, FastEthernet1/0
L    192.168.10.6/32 is directly connected, FastEthernet1/0
O    192.168.11.0/24 [110/3] via 192.168.9.2, 00:29:27, FastEthernet0/1
O    192.168.12.0/24 [110/3] via 192.168.9.2, 02:08:33, FastEthernet0/1
O    192.168.13.0/24 [110/4] via 192.168.9.2, 00:29:27, FastEthernet0/1
O    192.168.14.0/24 [110/4] via 192.168.9.2, 00:29:27, FastEthernet0/1
O    192.168.15.0/24 [110/3] via 192.168.9.2, 02:08:22, FastEthernet0/1

```

Figure 26: R6's global routing table.

This is further confirmed by Figure 27, which lists both BGP neighbors of R6.

```

R6#show bgp vpnv4 unicast all summary
BGP router identifier 6.6.6.6, local AS number 1
BGP table version is 7, main routing table version 7
4 network entries using 640 bytes of memory
4 path entries using 224 bytes of memory
4/4 BGP path/bestpath attribute entries using 544 bytes of memory
2 BGP extended community entries using 80 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 1488 total bytes of memory
BGP activity 4/0 prefixes, 4/0 paths, scan interval 60 secs

Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ OutQ Up/Down  State/PfxRcd
7.7.7.7        4        1    144    143      7     0   0 02:05:06      0
8.8.8.8        4        1    144    143      7     0   0 02:05:07      2
R6#

```

Figure 27: R6's BGP neighbors.

Additionally, it's possible to view R6's Virtual Routing and Forwarding (VRF) table for Customer\_1 in Figure 28. This table shows that to reach subnet 10.0.110.0/24, the traffic must go through 8.8.8.8, which is R8's loopback interface, using BGP.

```

R6#show ip route vrf Customer_1

Routing Table: Customer_1
Codes: L - local, 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, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

    9.0.0.0/32 is subnetted, 1 subnets
O       9.9.9.9 [110/2] via 10.0.100.9, 02:06:20, FastEthernet0/0
    10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C       10.0.100.0/24 is directly connected, FastEthernet0/0
L       10.0.100.6/32 is directly connected, FastEthernet0/0
B       10.0.110.0/24 [200/0] via 8.8.8.8, 02:05:03
    11.0.0.0/32 is subnetted, 1 subnets
B       11.11.11.11 [200/2] via 8.8.8.8, 02:05:03

```

**Figure 28:** R6's VRF Customer\_1 table.

This information is useful as we now need to determine which labels were used for the ICMP packets. Starting with R6, this router, acting as a Label Edge Router (LER), is responsible for labeling the IP packets with two labels: the outer label (IGP label) and the inner label (VPN label).

The outer label is used to switch the packets within the MPLS network to traverse from one point to another, while the inner label is used by the other PE router to forward traffic to the respective CE router.

Figures 29 and 30 display these labels. It can be concluded that the outer label to be inserted is 23, as previously mentioned, with the next hop being 8.8.8.8.

For the inner label, R6 will use label 33, as indicated in the third entry of Figure 30, which shows that for the route distinguisher matching Customer\_1, subnet 10.0.110.0/24 can be reached via the next hop 8.8.8.8 with an "Out label" value of 33.

```

R6#show mpls ldp bindings
lib entry: 6.6.6.6/32, rev 6
  local binding: label: imp-null
  remote binding: lsr: 192.168.9.2:0, label: 25
lib entry: 7.7.7.7/32, rev 34
  local binding: label: 28
  remote binding: lsr: 192.168.9.2:0, label: 24
lib entry: 8.8.8.8/32, rev 33
  local binding: label: 27
  remote binding: lsr: 192.168.9.2:0, label: 23

```

**Figure 29:** R6's LIB table.

```

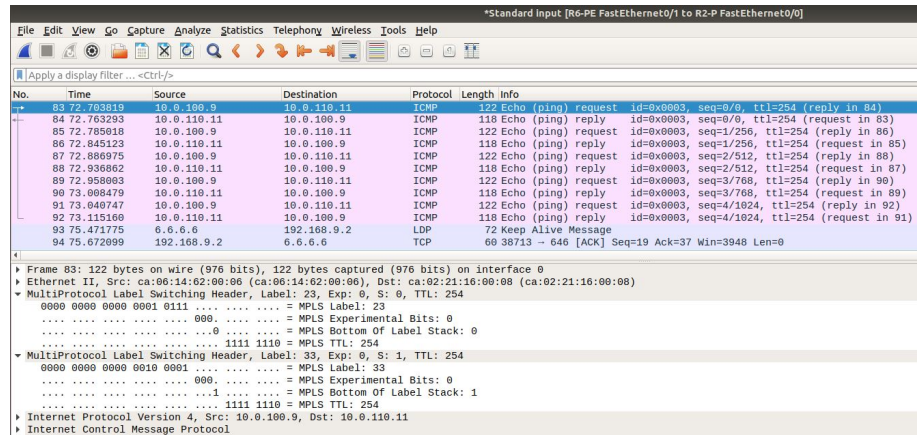
R6#show ip bgp vpnv4 all label
Network          Next Hop        In label/Out label
Route Distinguisher: 1:1 (Customer_1)
9.9.9.9/32       10.0.100.9     33/nolabel
10.0.100.0/24    0.0.0.0        34/nolabel(Customer_1)
10.0.110.0/24    8.8.8.8        nolabel/33
11.11.11.11/32   8.8.8.8        nolabel/34

```

**Figure 30:** R6's BGP VPNv4 address labels.

To guarantee these proposed values were applied to the packets, let's inspect Figure 31 which contains a sample of the Wireshark capture between R6 and R2. As expected, the ICMP Echo Request packet now contains two MPLS shim headers:

- MPLS label 23 with the bottom of label stack bit set to 0.
- MPLS label 33 has the bottom of the label stack bit set to 1. This indicates that label 33 is the last or inner label, confirming that the VPN label is only needed for the final portion of the LSP path.



**Figure 31:** Wireshark capture between R2 and R6.

Now that the packet has arrived at R2, this LSR must perform label switching on the outer label. Since the packet includes a label, R2 will refer to its LFIB to switch the label and forward the packet. As shown in Figure 32, which contains R2's LFIB table, label 23 will be swapped with label 21, with the next hop being 192.168.1.1 (R1's interface).



```
R2#show mpls for
```

```
R2#show mpls forwarding-table
```

Local Label	Outgoing Label	Prefix or Tunnel Id	Bytes Switched	Outgoing interface	Next Hop
20	Pop Label	192.168.12.0/24	0	Fa1/1	192.168.2.3
21	Pop Label	192.168.7.0/24	0	Fa1/1	192.168.2.3
	Pop Label	192.168.7.0/24	0	Fa0/1	192.168.6.5
22	Pop Label	192.168.3.0/24	0	Fa1/1	192.168.2.3
23	21	8.8.8.8/32	30554	Fa1/0	192.168.1.1
	16	8.8.8.8/32	0	Fa0/1	192.168.6.5
24	24	7.7.7.7/32	37582	Fa1/1	192.168.2.3
25	Pop Label	6.6.6.6/32	43831	Fa0/0	192.168.9.6
26	24	192.168.13.0/24	0	Fa1/0	192.168.1.1
	30	192.168.13.0/24	0	Fa1/1	192.168.2.3
	27	192.168.13.0/24	0	Fa0/1	192.168.6.5
27	25	192.168.14.0/24	0	Fa1/0	192.168.1.1
	33	192.168.14.0/24	0	Fa1/1	192.168.2.3
	23	192.168.14.0/24	60	Fa0/1	192.168.6.5
28	Pop Label	192.168.10.0/24	0	Fa1/0	192.168.1.1

Figure 32: R2's LFIB table.

When examining Figure 33, the Wireshark capture confirms that the outer label was indeed swapped. The two MPLS shim headers now contain the following labels:

- MPLS label 21 with the bottom of the label stack bit set to 0 and a TTL of 253, reflecting the hop decrement by R2.
- MPLS label 33 with the bottom of the label stack bit set to 1 and a TTL of 254, as this label was not used for hopping.

\*Standard Input [R1-P FastEthernet1/0 to R2-P FastEthernet1/0]

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-F>

No.	Time	Source	Destination	Protocol	Length	Info
59	56.335264	10.0.100.9	10.0.110.11	ICMP	122	Echo (ping) request id=0x0003, seq=0/0, ttl=254 (reply in 60)
60	56.338456	10.0.110.11	10.0.100.9	ICMP	122	Echo (ping) reply id=0x0003, seq=0/0, ttl=254 (request in 59)
61	56.417685	10.0.100.9	10.0.110.11	ICMP	122	Echo (ping) request id=0x0003, seq=1/256, ttl=254 (reply in 62)
62	56.454854	10.0.110.11	10.0.100.9	ICMP	122	Echo (ping) reply id=0x0003, seq=1/256, ttl=254 (request in 61)
63	56.519586	10.0.100.9	10.0.110.11	ICMP	122	Echo (ping) request id=0x0003, seq=2/512, ttl=254 (reply in 64)
64	56.556484	10.0.110.11	10.0.100.9	ICMP	122	Echo (ping) reply id=0x0003, seq=2/512, ttl=254 (request in 63)
65	56.591254	10.0.100.9	10.0.110.11	ICMP	122	Echo (ping) request id=0x0003, seq=3/768, ttl=254 (reply in 66)
66	56.628849	10.0.110.11	10.0.100.9	ICMP	122	Echo (ping) reply id=0x0003, seq=3/768, ttl=254 (request in 65)
67	56.673208	10.0.100.9	10.0.110.11	ICMP	122	Echo (ping) request id=0x0003, seq=4/1024, ttl=254 (reply in 68)
68	56.718600	10.0.110.11	10.0.100.9	ICMP	122	Echo (ping) reply id=0x0003, seq=4/1024, ttl=254 (request in 67)
69	56.801910	192.168.1.2	224.0.0.2	LDP	76	Hello Message
70	57.538351	192.168.1.1	224.0.0.2	LDP	76	Hello Message

Frame 59: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface 0

Ethernet II, Src: ca:02:21:16:00:1c (ca:02:21:16:00:1c), Dst: ca:01:20:f9:00:1c (ca:01:20:f9:00:1c)

MultiProtocol Label Switching Header, Label: 21, Exp: 0, S: 0, TTL: 253

0000 0000 0000 0101 ..... = MPLS Label: 21

..... 000 ..... = MPLS Experimental Bits: 0

..... 1111 1101 = MPLS Bottom Of Label Stack: 0

..... 0 ..... = MPLS TTL: 253

MultiProtocol Label Switching Header, Label: 33, Exp: 0, S: 1, TTL: 254

0000 0000 0000 0010 0001 ..... = MPLS Label: 33

..... 000 ..... = MPLS Experimental Bits: 0

..... 1 ..... = MPLS Bottom Of Label Stack: 1

..... 1111 1110 = MPLS TTL: 254

Internet Protocol Version 4, Src: 10.0.100.9, Dst: 10.0.110.11

Internet Control Message Protocol

Figure 33: Wireshark capture between R2 and R1.

Upon arrival at R1, the packet undergoes the same process as at R2, consulting its LFIB table to determine the next label to be used. As shown in Figure 34, the table indicates that the next action is to perform label popping (PHP) and forward the packet towards the next hop, 192.168.11.8.

```
R1#show mpls forwarding-table
```

Local Label	Outgoing Label	Prefix or Tunnel Id	Bytes Switched	Label	Outgoing interface	Next Hop
21	Pop Label	8.8.8.8/32	37202		Fa2/1	192.168.11.8
22	25	6.6.6.6/32	2270		Fa1/0	192.168.1.2
23	20	192.168.12.0/24	0		Fa1/0	192.168.1.2
	18	192.168.12.0/24	0		Fa1/1	192.168.5.5
	24	192.168.12.0/24	0		Fa0/1	192.168.8.4
24	Pop Label	192.168.13.0/24	0		Fa0/1	192.168.8.4
25	Pop Label	192.168.14.0/24	0		Fa0/1	192.168.8.4
	Pop Label	192.168.14.0/24	0		Fa2/1	192.168.11.8
26	Pop Label	192.168.7.0/24	0		Fa1/1	192.168.5.5
27	Pop Label	192.168.3.0/24	0		Fa0/1	192.168.8.4
28	Pop Label	192.168.4.0/24	0		Fa1/1	192.168.5.5
	Pop Label	192.168.4.0/24	0		Fa0/1	192.168.8.4
29	Pop Label	192.168.15.0/24	0		Fa1/1	192.168.5.5
	Pop Label	192.168.15.0/24	0		Fa2/1	192.168.11.8
30	Pop Label	192.168.6.0/24	0		Fa1/0	192.168.1.2
	Pop Label	192.168.6.0/24	0		Fa1/1	192.168.5.5
31	Pop Label	192.168.2.0/24	0		Fa1/0	192.168.1.2
32	Pop Label	192.168.9.0/24	0		Fa1/0	192.168.1.2
33	22	7.7.7.7/32	0		Fa0/1	192.168.8.4

Figure 34: R1's LFIB table.

Moving on to examining the Wireshark capture between R1 and R8, we would anticipate that the number of MPLS shim headers is reduced to one and only the inner label should be present. We also expect that the bottom of the label stack bit is set to 1, and the MPLS TTL should continue the downward counting from the previous outer label. Figure 35 confirms these expectations.

\*Standard Input [R1-P FastEthernet2/1 to R8-PE FastEthernet2/0]

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-F>

No.	Time	Source	Destination	Protocol	Length	Info
55	48.102095	10.0.100.0	10.0.110.11	ICMP	118	Echo (ping) request id=0x0003, seq=0/0, ttl=254 (request in 56)
56	48.223000	10.0.110.11	10.0.100.9	ICMP	118	Echo (ping) reply id=0x0003, seq=0/0, ttl=254 (reply in 56)
57	48.274686	10.0.100.9	10.0.110.11	ICMP	118	Echo (ping) request id=0x0003, seq=1/256, ttl=254 (request in 58)
58	48.295068	10.0.110.11	10.0.100.9	ICMP	118	Echo (ping) reply id=0x0003, seq=1/256, ttl=254 (reply in 57)
59	48.376272	10.0.100.9	10.0.110.11	ICMP	118	Echo (ping) request id=0x0003, seq=2/512, ttl=254 (request in 60)
60	48.396993	10.0.110.11	10.0.100.9	ICMP	118	Echo (ping) reply id=0x0003, seq=2/512, ttl=254 (reply in 59)
61	48.448593	10.0.100.9	10.0.110.11	ICMP	118	Echo (ping) request id=0x0003, seq=3/768, ttl=254 (request in 62)
62	48.469420	10.0.110.11	10.0.100.9	ICMP	118	Echo (ping) reply id=0x0003, seq=3/768, ttl=254 (reply in 61)
63	48.530340	10.0.100.9	10.0.110.11	ICMP	118	Echo (ping) request id=0x0003, seq=4/1024, ttl=254 (request in 64)
64	48.551131	10.0.110.11	10.0.100.9	ICMP	118	Echo (ping) reply id=0x0003, seq=4/1024, ttl=254 (reply in 63)
65	50.475548	192.168.11.1	224.0.0.2	LDP	76	Hello Message

Frame 55: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface 0

Ethernet II, Src: ca:01:20:f9:00:39 (ca:01:20:f9:00:39), Dst: ca:08:14:9b:00:38 (ca:08:14:9b:00:38)

MultiProtocol Label Switching Header, Label: 33, Exp: 0, S: 1, TTL: 252

0000 0000 0000 0010 0001 ..... = MPLS Label: 33

..... 0000 ..... = MPLS Experimental Bits: 0

..... 1 ..... = MPLS Bottom Of Label Stack: 1

..... 1111 1100 = MPLS TTL: 252

Internet Protocol Version 4, Src: 10.0.100.9, Dst: 10.0.110.11

Internet Control Message Protocol

Figure 35: Wireshark capture between R1 and R8.

As the packet finally reaches R8, the destination PE router, R8's responsibility is to remove the VPN label and forward the traffic based on its own mapping of BGP VPNv4 address labels. Specifically, it specifies that for the route distinguisher matching Customer\_1, when attempting to reach subnet 10.0.110.0/24, the next hop should be 0.0.0.0 (directly connected), and the outgoing label should be empty, indicating that it is popping the VPN label. This is illustrated in Figure 36.



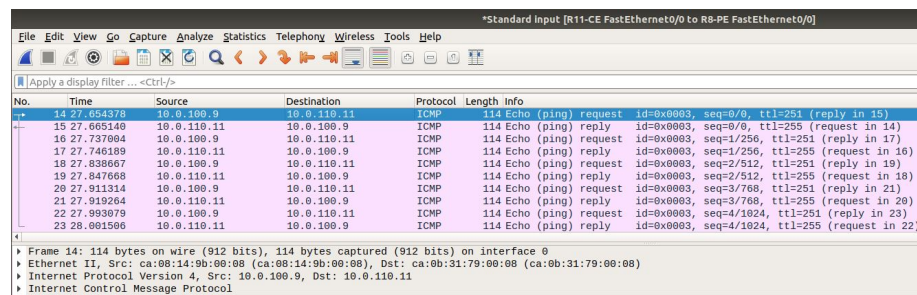
```

R8#show ip bgp vpnv4 all labels
Network          Next Hop      In label/Out label
Route Distinguisher: 1:1 (Customer_1)
 9.9.9.9/32      6.6.6.6      noLabel/33
10.0.100.0/24    6.6.6.6      noLabel/34
10.0.110.0/24    0.0.0.0      33/noLabel(Customer_1)
11.11.11.11/32   10.0.110.11   34/noLabel
Route Distinguisher: 1:2 (Customer_2)
10.0.200.0/24    7.7.7.7      noLabel/33
10.0.220.0/24    0.0.0.0      35/noLabel(Customer_2)
10.10.10.10/32   7.7.7.7      noLabel/34
12.12.12.12/32   10.0.220.12   36/noLabel

```

**Figure 36:** R8's BGP VPNv4 address labels.

Figure 37 illustrates the Wireshark capture between R8 and R11, indicating that the packet is now stripped of MPLS shim headers and has reverted to IP routing.



No.	Time	Source	Destination	Protocol	Length	Info
14	27.654378	10.0.100.9	10.0.110.11	ICMP	114	Echo (ping) request id=0x0003, seq=0/0, ttl=251 (reply in 15)
15	27.665140	10.0.110.11	10.0.100.9	ICMP	114	Echo (ping) reply id=0x0003, seq=0/0, ttl=255 (request in 14)
16	27.737084	10.0.100.9	10.0.110.11	ICMP	114	Echo (ping) request id=0x0003, seq=1/256, ttl=251 (reply in 17)
17	27.746189	10.0.110.11	10.0.100.9	ICMP	114	Echo (ping) reply id=0x0003, seq=1/256, ttl=255 (request in 16)
18	27.838667	10.0.100.9	10.0.110.11	ICMP	114	Echo (ping) request id=0x0003, seq=2/512, ttl=251 (reply in 19)
19	27.847668	10.0.110.11	10.0.100.9	ICMP	114	Echo (ping) reply id=0x0003, seq=2/512, ttl=255 (request in 18)
20	27.911314	10.0.100.9	10.0.110.11	ICMP	114	Echo (ping) request id=0x0003, seq=3/768, ttl=251 (reply in 21)
21	27.919264	10.0.110.11	10.0.100.9	ICMP	114	Echo (ping) reply id=0x0003, seq=3/768, ttl=255 (request in 20)
22	27.993079	10.0.100.9	10.0.110.11	ICMP	114	Echo (ping) request id=0x0003, seq=4/1024, ttl=251 (reply in 23)
23	28.001596	10.0.110.11	10.0.100.9	ICMP	114	Echo (ping) reply id=0x0003, seq=4/1024, ttl=255 (request in 22)

▶ Frame 14: 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interface 0  
 ▶ Ethernet II, Src: ca:08:14:9b:00:08 (ca:08:14:9b:00:08), Dst: ca:0b:31:79:00:08 (ca:0b:31:79:00:08)  
 ▶ Internet Protocol Version 4, Src: 10.0.100.9, Dst: 10.0.110.11  
 ▶ Internet Control Message Protocol

**Figure 37:** Wireshark capture between R8 and R11.

This concludes the demonstration of how one customer can communicate with his multiple sites using VPN-IP over a MPLS network.

# Appendices

## A Part A and B - MPLS

```
#PC1
ip 192.168.0.1 255.255.255.0 192.168.0.254
set pcname PC1

#PC2
ip 192.168.7.1 255.255.255.0 192.168.7.254
set pcname PC2

#R1
hostname R1
!
ip cef
mpls ip
!
router ospf 1
network 192.168.0.0 0.0.0.255 area 0
network 192.168.1.0 0.0.0.255 area 0
network 192.168.4.0 0.0.0.255 area 0
exit
!
mpls label protocol ldp
!
interface FastEthernet0/0
ip address 192.168.0.254 255.255.255.0
no shutdown
!
interface FastEthernet1/0
ip address 192.168.1.1 255.255.255.0
mpls ip
no shutdown
!
interface FastEthernet1/1
ip address 192.168.4.1 255.255.255.0
mpls ip
no shutdown

#R2
hostname R2
!
ip cef
mpls ip
!
router ospf 1
network 192.168.1.0 0.0.0.255 area 0
```

```

network 192.168.2.0 0.0.0.255 area 0
exit
!
mpls label protocol ldp
!
interface FastEthernet1/0
ip address 192.168.1.2 255.255.255.0
no shutdown
mpls ip
!
interface FastEthernet1/1
ip address 192.168.2.2 255.255.255.0
mpls ip
no shutdown

#R3
hostname R3
!
ip cef
mpls ip
!
router ospf 1
network 192.168.2.0 0.0.0.255 area 0
network 192.168.3.0 0.0.0.255 area 0
network 192.168.6.0 0.0.0.255 area 0
exit
!
mpls label protocol ldp
!
interface FastEthernet0/1
ip address 192.168.2.3 255.255.255.0
mpls ip
no shutdown
!
interface FastEthernet1/0
ip address 192.168.6.3 255.255.255.0
mpls ip
no shutdown
!
interface FastEthernet1/1
ip address 192.168.3.3 255.255.255.0
mpls ip
no shutdown

#R4
hostname R4
!
ip cef
mpls ip
!

```

```

router ospf 1
network 192.168.4.0 0.0.0.255 area 0
network 192.168.5.0 0.0.0.255 area 0
network 192.168.6.0 0.0.0.255 area 0
exit
!
mpls label protocol ldp
!
interface FastEthernet0/1
ip address 192.168.4.4 255.255.255.0
mpls ip
no shutdown
!
interface FastEthernet1/0
ip address 192.168.6.4 255.255.255.0
mpls ip
no shutdown
!
interface FastEthernet1/1
ip address 192.168.5.4 255.255.255.0
mpls ip
no shutdown

#R5
hostname R5
!
ip cef
mpls ip
!
router ospf 1
network 192.168.3.0 0.0.0.255 area 0
network 192.168.5.0 0.0.0.255 area 0
network 192.168.7.0 0.0.0.255 area 0
exit
!
mpls label protocol ldp
!
interface FastEthernet0/0
ip address 192.168.7.254 255.255.255.0
no shutdown
!
interface FastEthernet1/0
ip address 192.168.3.5 255.255.255.0
mpls ip
no shutdown
!
interface FastEthernet1/1
ip address 192.168.5.5 255.255.255.0
mpls ip
no shutdown

```

---

## B Part C - VPN-IP over MPLS

```
#R1
hostname R1
!
ip cef
mpls ip
!
mpls label protocol ldp
!
router ospf 1
network 192.168.1.0 0.0.0.255 area 0
network 192.168.5.0 0.0.0.255 area 0
network 192.168.8.0 0.0.0.255 area 0
network 192.168.10.0 0.0.0.255 area 0
network 192.168.11.0 0.0.0.255 area 0
!
interface FastEthernet0/1
ip address 192.168.8.1 255.255.255.0
mpls ip
no shutdown
!
interface FastEthernet1/0
ip address 192.168.1.1 255.255.255.0
mpls ip
no shutdown
!
interface FastEthernet1/1
ip address 192.168.5.1 255.255.255.0
mpls ip
no shutdown
!
interface FastEthernet2/0
ip address 192.168.10.1 255.255.255.0
mpls ip
no shutdown
!
interface FastEthernet2/1
ip address 192.168.11.1 255.255.255.0
mpls ip
no shutdown
!

#R2
hostname R2
!
```

```

ip cef
mpls ip
!
router ospf 1
network 192.168.1.0 0.0.0.255 area 0
network 192.168.2.0 0.0.0.255 area 0
network 192.168.6.0 0.0.0.255 area 0
network 192.168.9.0 0.0.0.255 area 0
!
mpls label protocol ldp
!
interface FastEthernet0/0
ip address 192.168.9.2 255.255.255.0
mpls ip
no shutdown
!
interface FastEthernet0/1
ip address 192.168.6.2 255.255.255.0
mpls ip
no shutdown
!
interface FastEthernet1/0
ip address 192.168.1.2 255.255.255.0
no shutdown
mpls ip
!
interface FastEthernet1/1
ip address 192.168.2.2 255.255.255.0
mpls ip
no shutdown

#R3
hostname R3
!
ip cef
mpls ip
!
router ospf 1
network 192.168.2.0 0.0.0.255 area 0
network 192.168.3.0 0.0.0.255 area 0
network 192.168.7.0 0.0.0.255 area 0
network 192.168.12.0 0.0.0.255 area 0
!
mpls label protocol ldp
!
interface FastEthernet0/0
ip address 192.168.12.3 255.255.255.0
mpls ip
no shutdown
!

```

```

interface FastEthernet0/1
ip address 192.168.2.3 255.255.255.0
mpls ip
no shutdown
!
interface FastEthernet1/0
ip address 192.168.7.3 255.255.255.0
mpls ip
no shutdown
!
interface FastEthernet1/1
ip address 192.168.3.3 255.255.255.0
mpls ip
no shutdown

#R4
hostname R4
!
ip cef
mpls ip
!
mpls label protocol ldp
!
router ospf 1
network 192.168.3.0 0.0.0.255 area 0
network 192.168.4.0 0.0.0.255 area 0
network 192.168.8.0 0.0.0.255 area 0
network 192.168.13.0 0.0.0.255 area 0
network 192.168.14.0 0.0.0.255 area 0
!
interface FastEthernet0/1
ip address 192.168.8.4 255.255.255.0
mpls ip
no shutdown
!
interface FastEthernet1/0
ip address 192.168.3.4 255.255.255.0
mpls ip
no shutdown
!
interface FastEthernet1/1
ip address 192.168.4.4 255.255.255.0
mpls ip
no shutdown
!
interface FastEthernet2/0
ip address 192.168.13.4 255.255.255.0
mpls ip
no shutdown
!

```



```

interface FastEthernet2/1
ip address 192.168.14.4 255.255.255.0
mpls ip
no shutdown
!

#R5
hostname R5
!
ip cef
mpls ip
!
mpls label protocol ldp
!
router ospf 1
network 192.168.4.0 0.0.0.255 area 0
network 192.168.5.0 0.0.0.255 area 0
network 192.168.6.0 0.0.0.255 area 0
network 192.168.7.0 0.0.0.255 area 0
network 192.168.15.0 0.0.0.255 area 0
!
interface FastEthernet0/0
ip address 192.168.6.5 255.255.255.0
mpls ip
no shutdown
!
interface FastEthernet0/1
ip address 192.168.5.5 255.255.255.0
mpls ip
no shutdown
!
interface FastEthernet1/0
ip address 192.168.7.5 255.255.255.0
mpls ip
no shutdown
!
interface FastEthernet1/1
ip address 192.168.4.5 255.255.255.0
mpls ip
no shutdown
!
interface FastEthernet2/0
ip address 192.168.15.5 255.255.255.0
mpls ip
no shutdown

#R6
hostname R6
!
ip cef

```

```

!
ip vrf Customer_1
  rd 1:1
  route-target export 1:1
  route-target import 1:1
!
mpls label protocol ldp
!
interface Loopback0
  ip address 6.6.6.6 255.255.255.255
  no shutdown
!
interface FastEthernet0/0
  ip vrf forwarding Customer_1
  ip address 10.0.100.6 255.255.255.0
  no shutdown
!
interface FastEthernet0/1
  ip address 192.168.9.6 255.255.255.0
  mpls ip
  no shutdown
!
interface FastEthernet1/0
  ip address 192.168.10.6 255.255.255.0
  mpls ip
  no shutdown
!
interface FastEthernet1/1
  no ip address
  shutdown
!
router ospf 2 vrf Customer_1
  redistribute bgp 1 subnets
  network 9.9.9.9 0.0.0.0 area 2
  network 10.0.100.0 0.0.0.255 area 2
!
router ospf 1
  network 6.6.6.6 0.0.0.0 area 0
  network 192.168.9.0 0.0.0.255 area 0
  network 192.168.10.0 0.0.0.255 area 0
!
router bgp 1
  bgp log-neighbor-changes
  neighbor 7.7.7.7 remote-as 1
  neighbor 7.7.7.7 update-source Loopback0
  neighbor 8.8.8.8 remote-as 1
  neighbor 8.8.8.8 update-source Loopback0
  !
  address-family vpnv4
    neighbor 7.7.7.7 activate

```

```

neighbor 7.7.7.7 send-community extended
neighbor 8.8.8.8 activate
neighbor 8.8.8.8 send-community extended
exit-address-family
!
address-family ipv4 vrf Customer_1
  redistribute ospf 2
exit-address-family

#R7
hostname R7
!
ip cef
!
ip vrf Customer_2
  rd 1:2
  route-target export 2:2
  route-target import 2:2
!
mpls label protocol ldp
!
interface Loopback0
  ip address 7.7.7.7 255.255.255.255
  no shutdown
!
interface FastEthernet0/0
  ip vrf forwarding Customer_2
  ip address 10.0.200.7 255.255.255.0
  ip ospf 3 area 3
  no shutdown
!
interface FastEthernet0/1
  ip address 192.168.12.7 255.255.255.0
  mpls ip
  no shutdown
!
interface FastEthernet1/0
  ip address 192.168.13.7 255.255.255.0
  mpls ip
  no shutdown
!
interface FastEthernet1/1
  no ip address
  shutdown
!
router ospf 3 vrf Customer_2
  redistribute bgp 1 subnets
  network 10.0.200.0 0.0.0.255 area 3
  network 10.10.10.10 0.0.0.0 area 3
!

```

```

router ospf 1
 network 7.7.7.7 0.0.0.0 area 0
 network 192.168.12.0 0.0.0.255 area 0
 network 192.168.13.0 0.0.0.255 area 0
!
router bgp 1
 bgp log-neighbor-changes
 neighbor 6.6.6.6 remote-as 1
 neighbor 6.6.6.6 update-source Loopback0
 neighbor 8.8.8.8 remote-as 1
 neighbor 8.8.8.8 update-source Loopback0
!
 address-family vpnv4
  neighbor 6.6.6.6 activate
  neighbor 6.6.6.6 send-community extended
  neighbor 8.8.8.8 activate
  neighbor 8.8.8.8 send-community extended
 exit-address-family
!
 address-family ipv4 vrf Customer_2
  redistribute ospf 3
 exit-address-family

#R8
!
hostname R8
!
ip cef
!
ip vrf Customer_1
 rd 1:1
 route-target export 1:1
 route-target import 1:1
!
ip vrf Customer_2
 rd 1:2
 route-target export 2:2
 route-target import 2:2
!
mpls label protocol ldp
!
interface Loopback0
 ip address 8.8.8.8 255.255.255.255
 no shutdown
!
interface FastEthernet0/0
 ip vrf forwarding Customer_1
 ip address 10.0.110.8 255.255.255.0
 no shutdown
!

```

```

interface FastEthernet0/1
 ip vrf forwarding Customer_2
 ip address 10.0.220.8 255.255.255.0
 ip ospf 3 area 3
 no shutdown
!
interface FastEthernet1/0
 ip address 192.168.15.8 255.255.255.0
 mpls ip
 no shutdown
!
interface FastEthernet1/1
 no ip address
 shutdown
!
interface FastEthernet2/0
 ip address 192.168.11.8 255.255.255.0
 mpls ip
 no shutdown
!
interface FastEthernet2/1
 ip address 192.168.14.8 255.255.255.0
 ip ospf 1 area 0
 mpls ip
 no shutdown
!
router ospf 2 vrf Customer_1
 redistribute bgp 1 subnets
 network 10.0.110.0 0.0.0.255 area 2
 network 11.11.11.11 0.0.0.0 area 2
!
router ospf 3 vrf Customer_2
 redistribute bgp 1 subnets
 network 10.0.220.0 0.0.0.255 area 3
 network 12.12.12.12 0.0.0.0 area 3
!
router ospf 1
 network 8.8.8.8 0.0.0.0 area 0
 network 192.168.11.0 0.0.0.255 area 0
 network 192.168.14.0 0.0.0.255 area 0
 network 192.168.15.0 0.0.0.255 area 0
!
router bgp 1
 bgp log-neighbor-changes
 neighbor 6.6.6.6 remote-as 1
 neighbor 6.6.6.6 update-source Loopback0
 neighbor 7.7.7.7 remote-as 1
 neighbor 7.7.7.7 update-source Loopback0
!
 address-family vpnv4

```

```

neighbor 6.6.6.6 activate
neighbor 6.6.6.6 send-community extended
neighbor 7.7.7.7 activate
neighbor 7.7.7.7 send-community extended
exit-address-family
!
address-family ipv4 vrf Customer_1
  redistribute ospf 2
exit-address-family
!
address-family ipv4 vrf Customer_2
  redistribute ospf 3
exit-address-family
!

#R9
!
hostname R9
!
ip cef
!
interface Loopback0
  ip address 9.9.9.9 255.255.255.255
  ip ospf 2 area 2
  no shutdown
!
interface FastEthernet0/0
  ip address 10.0.100.9 255.255.255.0
  no shutdown
!
interface FastEthernet0/1
  no ip address
  shutdown
!
interface FastEthernet1/0
  no ip address
  shutdown
!
interface FastEthernet1/1
  no ip address
  shutdown
!
router ospf 2
  network 10.0.100.0 0.0.0.255 area 2
!

#R10
!
hostname R10
!

```

```

ip cef
!
interface Loopback0
 ip address 10.10.10.10 255.255.255.255
 ip ospf 3 area 3
 no shutdown
!
interface FastEthernet0/0
 ip address 10.0.200.10 255.255.255.0
 no shutdown
!
interface FastEthernet0/1
 no ip address
 shutdown
!
interface FastEthernet1/0
 no ip address
 shutdown
!
interface FastEthernet1/1
 no ip address
 shutdown
!
router ospf 3
 network 10.0.200.0 0.0.0.255 area 3

#R11
hostname R11
!
ip cef
!
interface Loopback0
 ip address 11.11.11.11 255.255.255.255
 ip ospf 2 area 2
 no shutdown
!
interface FastEthernet0/0
 ip address 10.0.110.11 255.255.255.0
 no shutdown
!
interface FastEthernet0/1
 no ip address
 no shutdown
!
interface FastEthernet1/0
 no ip address
 no shutdown
!
interface FastEthernet1/1
 no ip address

```



```

no shutdown
!
router ospf 2
network 10.0.110.0 0.0.0.255 area 2
!

#R12
!
hostname R12
!
ip cef
!
interface Loopback0
ip address 12.12.12.12 255.255.255.255
ip ospf 3 area 3
no shutdown
!
interface FastEthernet0/0
ip address 10.0.220.12 255.255.255.0
ip ospf 3 area 3
no shutdown
!
interface FastEthernet0/1
no ip address
no shutdown
!
interface FastEthernet1/0
no ip address
no shutdown
!
interface FastEthernet1/1
no ip address
no shutdown
!
router ospf 3
network 10.0.220.0 0.0.0.255 area 3

```