



UNIVERSITY OF
SOUTH CAROLINA

SOFTWARE DEFINED NETWORKING

Lab 3: Early efforts of SDN: MPLS example of a control plane that establishes semi-static forwarding paths

Document Version: **06-01-2020**



Award 1829698

“CyberTraining CIP: Cyberinfrastructure Expertise on High-throughput Networks for Big Science Data Transfers”

Contents

Overview	2
Objectives.....	2
Lab settings	2
Lab roadmap	2
1 Introduction	2
1.1 Introduction to MPLS	3
1.2 MPLS Header architecture	4
2 Lab topology.....	5
2.1 Lab settings.....	5
2.2 Opening the topology	6
2.3 Loading the configuration file	7
2.4 Running the emulation.....	9
2.5 Enabling MPLS forwarding	9
2.6 Verifying the configuration	12
3 Configuring static MPLS from router r1 to router r2	17
3.1 Pushing labels.....	17
3.2 Swapping labels.....	21
3.3 Popping labels	27
4 Configuring static MPLS from router r2 to router r1	31
5 Verifying configuration	33
References	37

Overview

The lab discusses the concept of Multiprotocol Label Switching (MPLS). Furthermore, static MPLS will be configured and verified between two hosts that are required to exchange routes.

Objectives

By the end of this lab, the user should be able to:

1. Explain the concept of MPLS.
2. Understand the concept of Software Defined Networking (SDN).
3. Configure and verify static MPLS.

Lab settings

The information in Table 1 provides the credentials to access the Client's virtual machine.

Table 1. Credentials to access Client's virtual machine.

Device	Account	Password
Client	admin	password

Lab roadmap

This lab is organized as follows:

1. Section 1: Introduction.
2. Section 2: Lab topology.
3. Section 3: Configuring static MPLS from router r1 to router r2.
4. Section 4: Configuring static MPLS from router r2 to router r1.
5. Section 5: Verifying configuration.

1 Introduction

MPLS is a deviation from the autonomous and distributed forwarding decisions characteristic of traditional Internet routers, in that sense it was a small step toward a more SDN-like Internet switching paradigm¹.

In a traditional IP network, each network router performs an IP lookup on the routed data or packet, the router determines a next-hop based on its routing table and forwards the

packet to the next-hop. Every router does the same on the same data or packet each making its own independent routing decision until the final destination is reached.

MPLS was originally envisioned as a technology that could improve the scaling of Layer 3 IP routing by avoiding the large number of "fully connected" routers that were required for the Internet core. In an MPLS enabled network, MPLS does "label switching"; which means the first router or network device does a routing lookup, but instead of finding a next-hop, it finds the final destination router. An MPLS configured router applies a "label" on the data, other routers use the label to route the traffic without needing to perform any additional IP lookups which makes packet forwarding faster.

Similar to MPLS, SDN is meant to address the fact that the static architecture of traditional networks is decentralized and complex while current networks require more flexibility and easy troubleshooting. SDN attempts to centralize network intelligence in one network component by disassociating the forwarding process of network packets (data plane) from the routing process (control plane). The control plane makes routing decisions. The data plane forwards data (packets) through the router. With SDN routing, decisions are made remotely instead of on each individual router.

1.1 Introduction to MPLS

MPLS is data forwarding technology that increases the speed and controls the flow of network traffic. The packet enters the edge of the MPLS backbone, is examined, classified, given an appropriate label, and forwarded to the next hop in the pre-set Label Switched Path (LSP). The predetermined paths that make MPLS work are called label-switched paths (LSPs). As the packet travels that path, each router on the path uses the label – not other information, such as the IP header – to make the forwarding decision that keeps the packet moving along the LSP. An MPLS router that performs routing based only on the label is called a label switch router (LSR). Each LSR has the ability to do three main things: push, pop, or swap labels from a packet. To push a label simply means to add a label to a packet, to pop is to remove a label from a packet and to swap is to remove and add an alternative label to the packet. It is possible for a packet to have multiple labels attached which are arranged in a stack and are considered in the order from the most recent label to the oldest label.

Within each router, the incoming label is examined, and its next hop is matched with a new label. The old label is replaced with the new label for the packet's next destination, and then the freshly labeled packet is sent to the next router. Each router repeats the process until the packet reaches the exit router. The label information is removed at either the last hop or the exit router, so that the packet goes back to being identified by an IP header instead of an MPLS label.

There are three different types of LSR: ingress, egress and intermediate. The ingress LSR is at the edge of an MPLS network and is the first to insert an MPLS header and label on a packet. The egress LSR is at the edge of the network and is the last point before leaving

the network and thus removes all the MPLS labels and header. Both the ingress and egress LSR's are considered Provider Edge (PE) routers.

The LSRs which exist within the network are called intermediate LSR's and are responsible for pushing, popping, and swapping labels based on the routing with the MPLS network; the intermediate LSR's are considered Provider (P) routers.

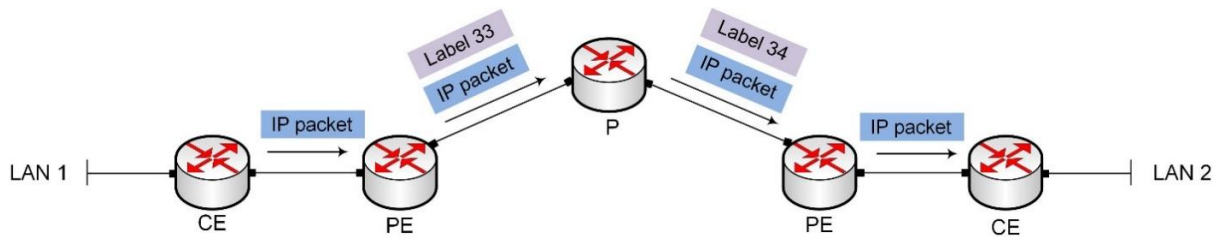


Figure 1. MPLS label forwarding.

Consider Figure 1. The Customer Edge (CE) of LAN 1 will forward the IP packet to the provider Edge (PE). The PE will push label 33 along with the IP packet to the provider (P). P will replace the label with label 34 which will be forwarded to the Provider Edge (PE). The PE will pop the label and the IP packet will be delivered to the Customer Edge (CE) of LAN 2 which is the destination router.

1.2 MPLS Header architecture

MPLS operates at a layer that is generally considered to lie between traditional definitions of OSI Layer 2 (data link layer) and Layer 3 (network layer), and thus is often referred to as a layer 2.5 protocol.

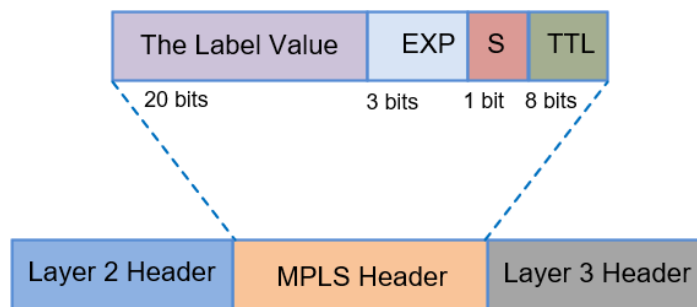


Figure 2. MPLS Header architecture.

Consider Figure 2. MPLS works by prefixing packets with an MPLS header, containing one or more labels. This is called a label stack. Each entry in the label stack contains four fields.

1. The Label Value: The first 20 bits are the label value. This value can be between 0 and 1,048,575. However, the first 16 values are exempted from normal use; that is, they have a special meaning.

2. EXP: Three bits are the experimental (EXP) bits. These bits are used solely for quality of service (QoS) which represents the set of techniques necessary to manage network bandwidth, delay, jitter, and packet loss.
3. S: 1 bit is the Bottom of Stack (BoS) bit. It is 0, unless this is the bottom label in the stack. If so, the BoS bit is set to 1. The stack is the collection of labels that are found on top of the packet. The stack can consist of just one label, or it might have more.
4. TTL: The last 8 bits are used for Time To Live (TTL). This TTL has the same function as the TTL found in the IP header. It is simply decreased by 1 at each hop, and its main function is to avoid a packet being stuck in a routing loop. If a routing loop occurs and no TTL is present, the packet loops forever. If the TTL of the label reaches 0, the packet is discarded².

2 Lab topology

Consider Figure 3. The topology consists of three networks, Customer 1, ISP and Customer 2. Customer 1 and Customer 2 exchange routes through static MPLS.

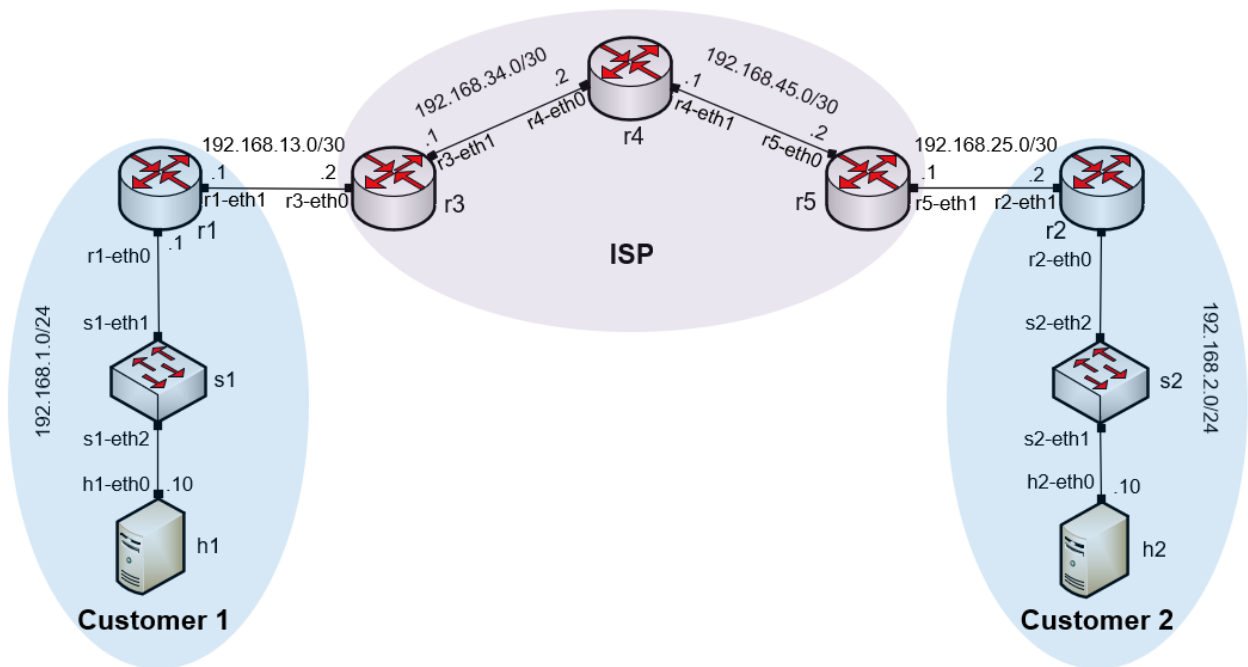


Figure 3. Lab topology.

2.1 Lab settings

Routers and hosts are already configured according to the IP addresses shown in Table 2.

Table 2. Topology information.

Device	Interface	IPV4 Address	Subnet	Default gateway
r1	r1-eth0	192.168.1.1	/24	N/A
	r1-eth1	192.168.13.1	/30	N/A
r2	r2-eth0	192.168.2.1	/24	N/A
	r2-eth1	192.168.25.1	/30	N/A
r3	r3-eth0	192.168.13.2	/30	N/A
	r3-eth1	192.168.34.1	/30	N/A
r4	r4-eth0	192.168.34.2	/30	N/A
	r4-eth1	192.168.45.1	/30	N/A
r5	r5-eth0	192.168.45.2	/30	N/A
	r5-eth1	192.168.25.2	/30	N/A
h1	h1-eth0	192.168.1.10	/24	192.168.1.1
h2	h2-eth0	192.168.2.10	/24	192.168.2.1

2.2 Opening the topology

In this section, you will open MiniEdit⁷ and load the lab topology. MiniEdit provides a Graphical User Interface (GUI) that facilitates the creation and emulation of network topologies in Mininet. This tool has additional capabilities such as: configuring network elements (i.e IP addresses, default gateway), saving topologies, and exporting layer 2 models.

Step 1. A shortcut to Miniedit is located on the machine's Desktop. Start Miniedit by clicking on Miniedit's shortcut. When prompted for a password, type `password`.

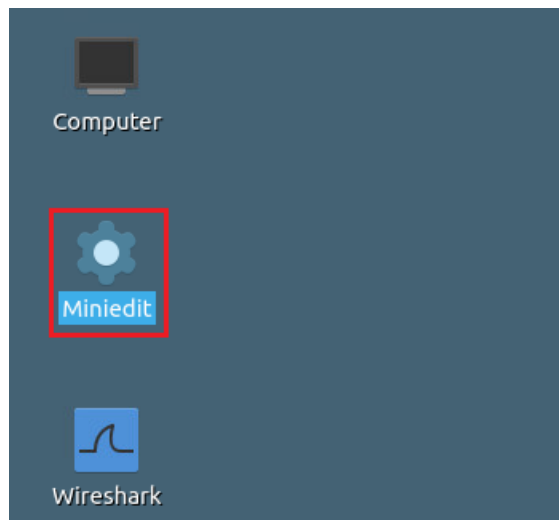


Figure 4. MiniEdit shortcut.

Step 2. On Miniedit's menu bar, click on *File* then *open* to load the lab topology. Open the *Lab3.mn* topology file stored in the default directory, */home/sdn/SDN_Labs /lab3* and click on *Open*.

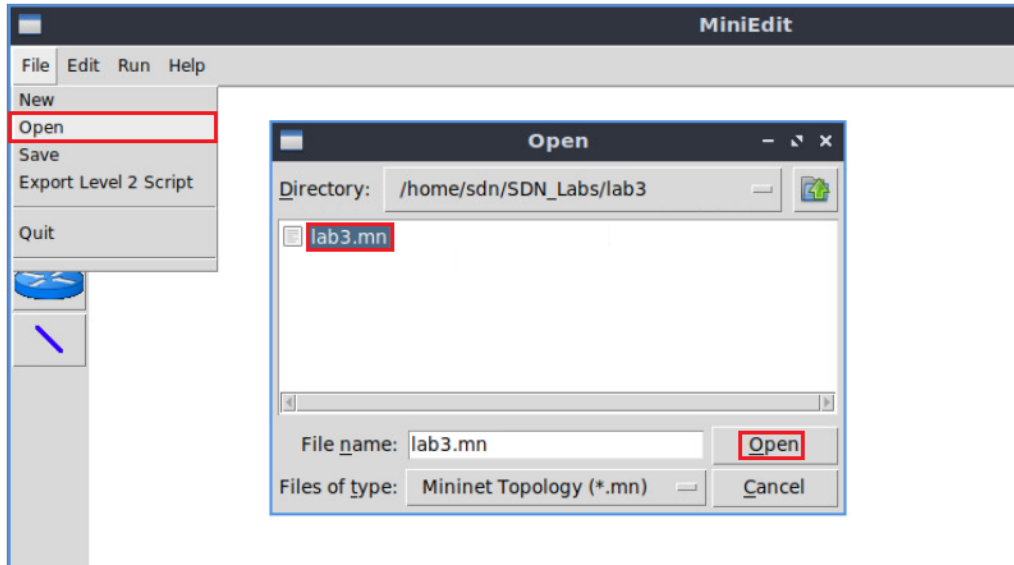


Figure 5. MiniEdit's open dialog.

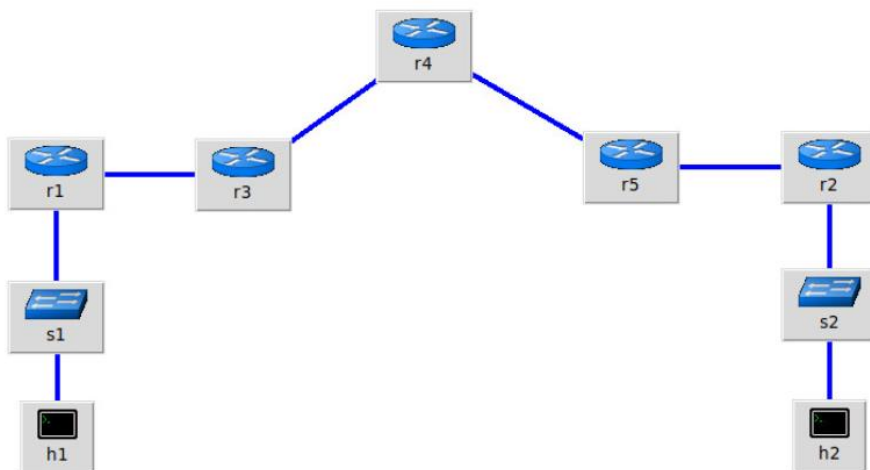


Figure 6. Mininet topology.

2.3 Loading the configuration file

At this point the topology is loaded however, the interfaces are not configured. In order to assign IP addresses to the devices' interfaces, you will execute a script that loads the configuration to the routers and end devices.

Step 1. Click on the icon below to open Linux terminal.



Figure 7. Opening the Linux terminal.

Step 2. Click on the Linux terminal and navigate into the *SDN_Labs/lab3* directory by issuing the following command. This folder contains a configuration file and the script responsible for loading the configuration. The configuration file will assign the IP addresses to the routers' interfaces. The `cd` command is short for change directory followed by an argument that specifies the destination directory.

```
cd SDN_Labs/lab3
```

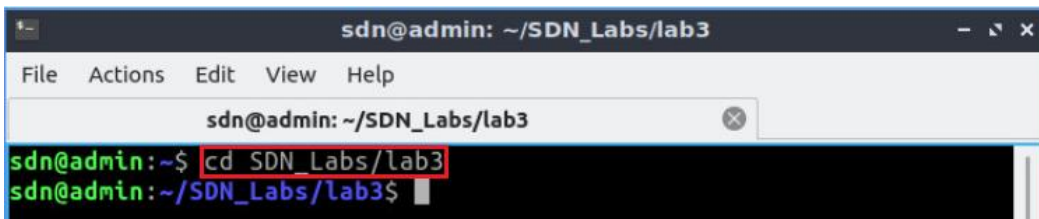


Figure 8. Entering the *SDN_Labs/lab3* directory.

Step 3. To execute the shell script, type the following command. The argument of the program corresponds to the configuration zip file that will be loaded in all the routers in the topology.

```
./config_loader.sh lab3_config.zip
```

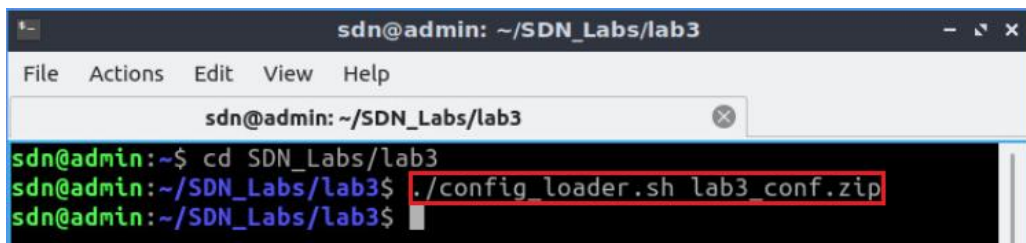


Figure 9. Executing the shell script to load the configuration.

Step 4. Type the following command to exit the Linux terminal.

```
exit
```

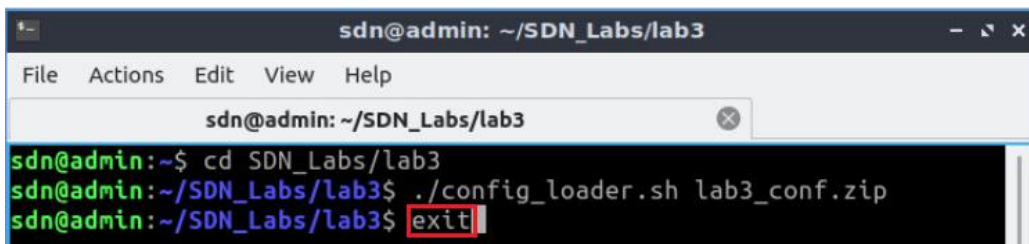


Figure 10. Exiting the terminal.

2.4 Running the emulation

In this section, you will run the emulation and check the links and interfaces that connect the devices in the given topology.

Step 1. At this point the interfaces of hosts h1 and h2 are configured. To proceed with the emulation, click on the *Run* button located in lower left-hand side.



Figure 11. Starting the emulation.

Step 2. Issue the following command to display the interface names and connections.

```
links
```

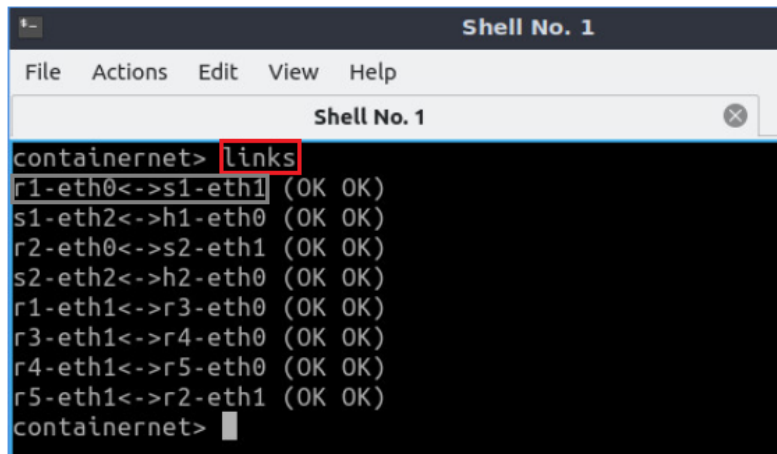


Figure 12. Displaying network interfaces.

In Figure 12, the link displayed within the gray box indicates that interface *eth1* of router r1 connects to interface *eth1* of switch s1 (i.e., *r1-eth0<->s1-eth1*).

2.5 Enabling MPLS forwarding

In this section, you will enable MPLS forwarding in the kernel. All the router interfaces assign labels which are used to forward packets. You will assign value 1 to all the router interfaces so that they participate in the labeling process. *Platform_labels* is the table which recognizes all the assigned labels and participates in the label forwarding. You will

assign value 100000 (maximum value for label forwarding) to *Platform_labels* in order to enable label forwarding.

Step 1. In order to open router r1's terminal, hold right-click on router r1 and select Terminal.

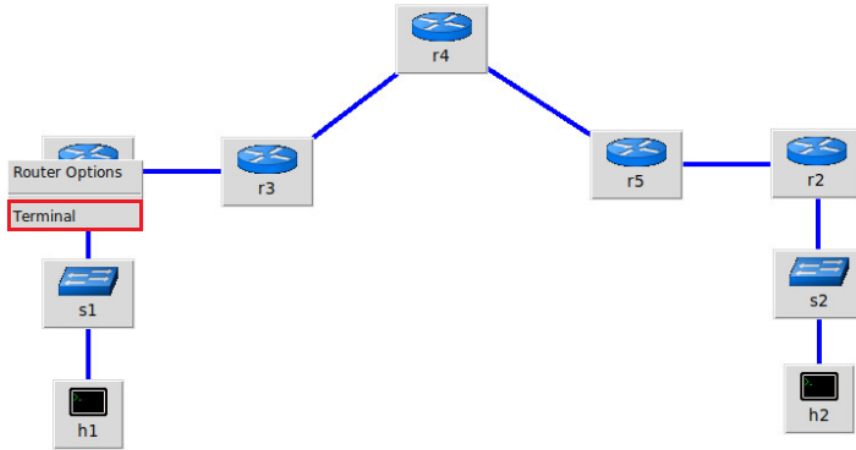


Figure 13. Opening a terminal on router r1.

Step 2. Type the following command to assign value 1 to interface *r1-eth0* in order to enable the interface to participate in the label processing.

```
sysctl -w net.mpls.conf.r1-eth0.input=1
```

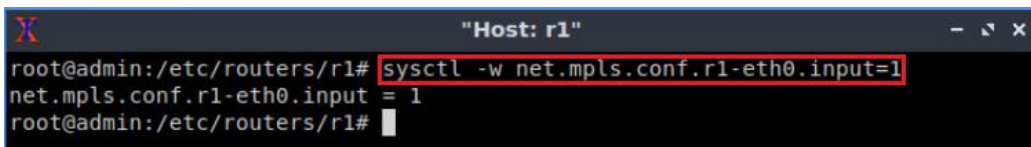


Figure 14. Enabling interface *r1-eth0* to participate in the label processing.

Step 3. Type the following command to assign value 1 to the interface *r1-eth1* in order to enable the interface to participate in the label processing.

```
sysctl -w net.mpls.conf.r1-eth1.input=1
```

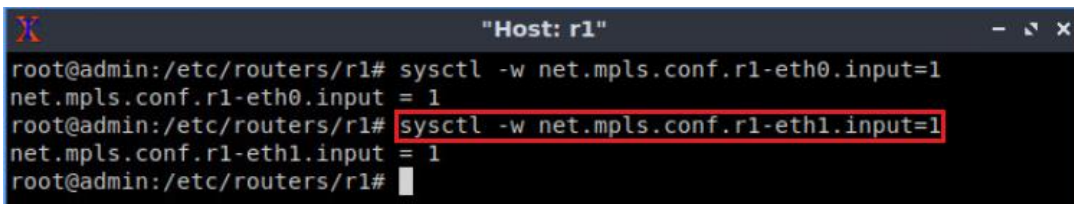
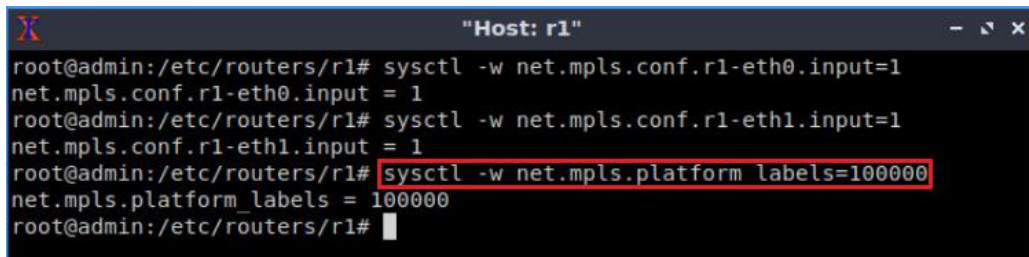


Figure 15. Enabling interface *r1-eth1* to participate in the label processing.

Step 4. In order to enable label forwarding, type the following command to assign the value 100000 to the *Platform_labels*.

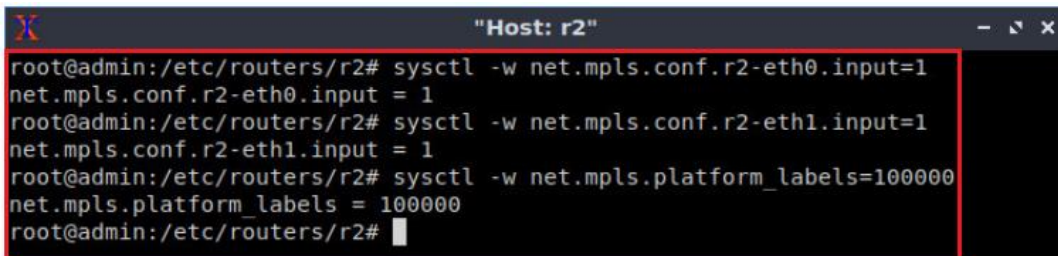
```
sysctl -w net.mpls.platform_labels=100000
```



```
root@admin:/etc/routers/r1# sysctl -w net.mpls.conf.r1-eth0.input=1
net.mpls.conf.r1-eth0.input = 1
root@admin:/etc/routers/r1# sysctl -w net.mpls.conf.r1-eth1.input=1
net.mpls.conf.r1-eth1.input = 1
root@admin:/etc/routers/r1# sysctl -w net.mpls.platform_labels=100000
net.mpls.platform_labels = 100000
root@admin:/etc/routers/r1#
```

Figure 16. Enabling label forwarding in router r1.

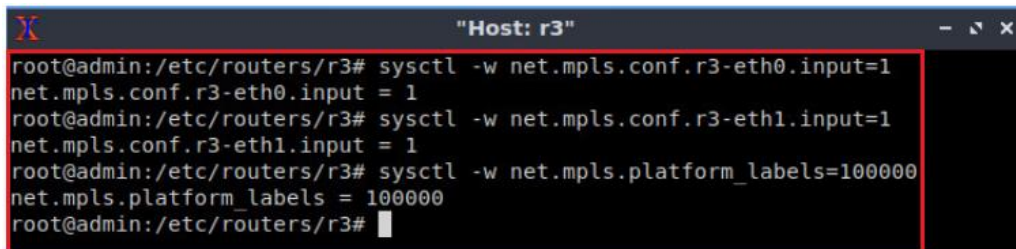
Step 5. Follow steps 2 to 4 in order to enable MPLS forwarding in router r2. All the steps are summarized in the following figure.



```
root@admin:/etc/routers/r2# sysctl -w net.mpls.conf.r2-eth0.input=1
net.mpls.conf.r2-eth0.input = 1
root@admin:/etc/routers/r2# sysctl -w net.mpls.conf.r2-eth1.input=1
net.mpls.conf.r2-eth1.input = 1
root@admin:/etc/routers/r2# sysctl -w net.mpls.platform_labels=100000
net.mpls.platform_labels = 100000
root@admin:/etc/routers/r2#
```

Figure 17. Enabling MPLS forwarding in router r2.

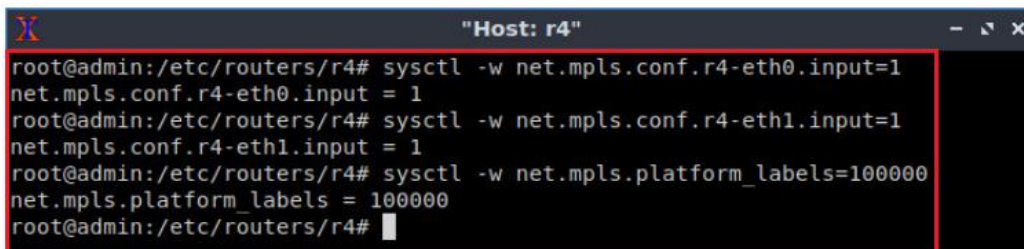
Step 6. Follow steps 2 to 4 in order to enable MPLS forwarding in router r3. All the steps are summarized in the following figure.



```
root@admin:/etc/routers/r3# sysctl -w net.mpls.conf.r3-eth0.input=1
net.mpls.conf.r3-eth0.input = 1
root@admin:/etc/routers/r3# sysctl -w net.mpls.conf.r3-eth1.input=1
net.mpls.conf.r3-eth1.input = 1
root@admin:/etc/routers/r3# sysctl -w net.mpls.platform_labels=100000
net.mpls.platform_labels = 100000
root@admin:/etc/routers/r3#
```

Figure 18. Enable MPLS forwarding in router r3.

Step 7. Follow steps 2 to 4 in order to enable MPLS forwarding in router r4. All the steps are summarized in the following figure.



```
root@admin:/etc/routers/r4# sysctl -w net.mpls.conf.r4-eth0.input=1
net.mpls.conf.r4-eth0.input = 1
root@admin:/etc/routers/r4# sysctl -w net.mpls.conf.r4-eth1.input=1
net.mpls.conf.r4-eth1.input = 1
root@admin:/etc/routers/r4# sysctl -w net.mpls.platform_labels=100000
net.mpls.platform_labels = 100000
root@admin:/etc/routers/r4#
```

Figure 19. Enable MPLS forwarding in router r4.

Step 8. Follow steps 2 to 4 in order to enable MPLS forwarding in router r5. All the steps are summarized in the following figure.

```
root@admin:/etc/routers/r5# sysctl -w net.mpls.conf.r5-eth0.input=1
net.mpls.conf.r5-eth0.input = 1
root@admin:/etc/routers/r5# sysctl -w net.mpls.conf.r5-eth1.input=1
net.mpls.conf.r5-eth1.input = 1
root@admin:/etc/routers/r5# sysctl -w net.mpls.platform_labels=100000
net.mpls.platform_labels = 100000
root@admin:/etc/routers/r5#
```

Figure 20. Enable MPLS forwarding in router r5.

2.6 Verifying the configuration

You will verify the IP addresses listed in Table 2 and inspect the routing table of routers r1, r2, r3, r4 and r5.

Step 1. Hold right-click on host h1 and select *Terminal*. This opens the terminal of host h1 and allows the execution of commands in that host.

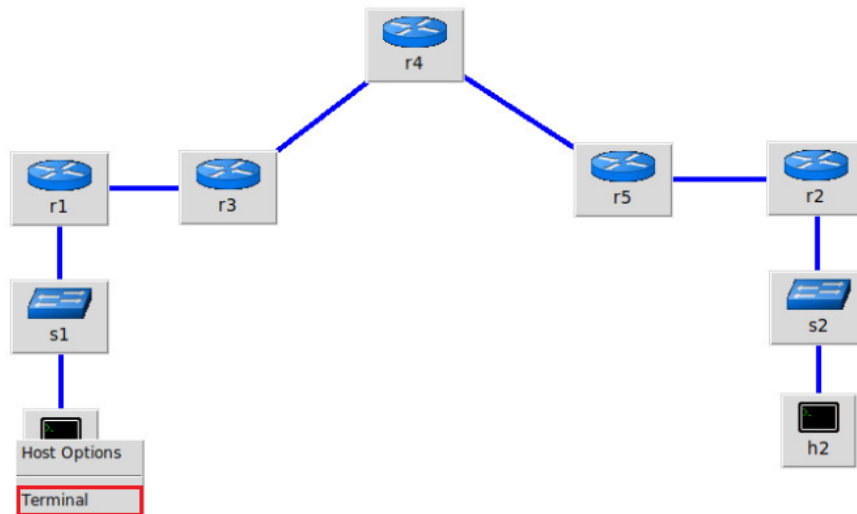


Figure 21. Opening a terminal in host h1.

Step 2. On host h1 terminal, type the command shown below to verify that the IP address was assigned successfully. You will corroborate that host h1 has interface, *h1-eth0* configured with the IP address 192.168.1.10 and the subnet mask 255.255.255.0.

```
ifconfig
```

```

Host: h1
root@admin:~# ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.10 netmask 255.255.255.0 broadcast 0.0.0.0
    ether la:a2:7f:17:52:dc txqueuelen 1000 (Ethernet)
    RX packets 23 bytes 3089 (3.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 3 bytes 270 (270.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@admin:~#
    
```

Figure 22. Output of the `ifconfig` command.

Step 3. In host h1’s terminal, type the command shown below to verify that the default gateway IP address is 192.168.1.1.

```
route
```

```

Host: h1
root@admin:~# route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
default 192.168.1.1 0.0.0.0 UG 0 0 0 h1-eth0
192.168.1.0 0.0.0.0 255.255.255.0 U 0 0 0 h1-eth0
root@admin:~#
    
```

Figure 23. Output of the `route` command.

Step 4. In order to verify host h2’s default route, proceed similarly by repeating from step 1 to step 3 in host h2’s terminal. Similar results should be observed.

Step 5. In order to verify router r1, hold right-click on router r1 and select *Terminal*.

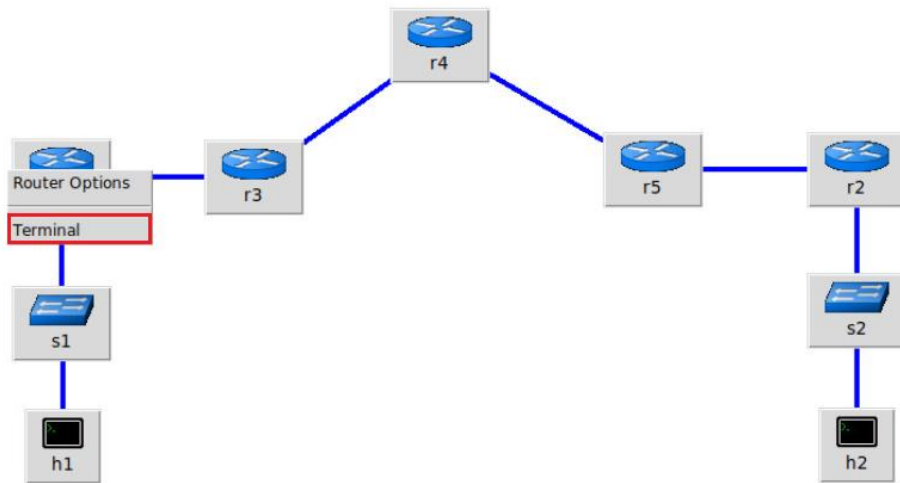


Figure 24. Opening a terminal in router r1.

Step 6. In this step, you will start zebra daemon, which is a multi-server routing software that provides TCP/IP based routing protocols. The configuration will not be working if you do not enable the zebra daemon initially. In order to start zebra, type the following command:

```
zebra
```

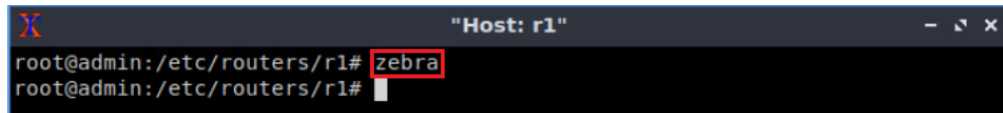


Figure 25. Starting the zebra daemon.

Step 7. After initializing zebra, vtysh should be started in order to provide all the CLI commands defined by the daemons. To proceed, issue the following command:

```
vtysh
```

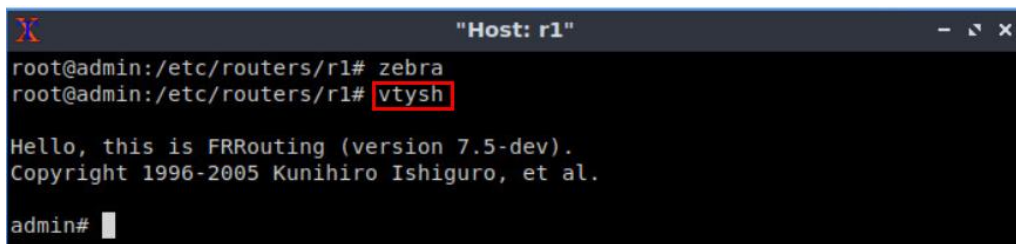


Figure 26. Starting vtysh on router r1.

Step 8. Type the following command in router r1 terminal to verify the routing table of router r1. It will list all the directly connected networks. The routing table of router r1 does not contain any route to the network of router r2 (192.168.2.0/24) as there is no routing protocol configured yet.

```
show ip route
```

```

Host: r1
root@admin:/etc/routers/r1# zebra
root@admin:/etc/routers/r1# vtysh

Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

admin# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued route, r - rejected route

C>* 192.168.1.0/24 is directly connected, r1-eth0, 00:01:15
C>* 192.168.13.0/30 is directly connected, r1-eth1, 00:01:15
admin#
    
```

Figure 27. Displaying routing table of router r1.

The output in the figure above shows that the network 192.168.1.0/24 is directly connected through the interface *r1-eth0*. The network 192.168.13.0/30 is connected via the interface *r1-eth1*.

Step 9. Router r2 is configured similarly to router r1, but with different IP addresses (see Table 2). Those steps are summarized in the following figure. To proceed, in router r2's terminal issue the commands depicted below. At the end, you will verify all the directly connected networks of router r2.

```

Host: r2
root@admin:/etc/routers/r2# zebra
root@admin:/etc/routers/r2# vtysh

Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

admin# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued route, r - rejected route

C>* 192.168.2.0/24 is directly connected, r2-eth0, 00:00:05
C>* 192.168.25.0/30 is directly connected, r2-eth1, 00:00:05
admin#
    
```

Figure 28. Displaying the routing table of router r2.

Step 10. Router r3 is configured similarly to router r1, but with different IP addresses (see Table 2). Those steps are summarized in the following figure. To proceed, in router r3's terminal issue the commands depicted below. At the end, you will verify all the directly connected networks of router r3.


```
Host: r3
root@admin:/etc/routers/r3# zebra
root@admin:/etc/routers/r3# vtysh

Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

admin# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued route, r - rejected route

C>* 192.168.13.0/30 is directly connected, r3-eth0, 00:00:05
C>* 192.168.34.0/30 is directly connected, r3-eth1, 00:00:05
admin#
```

Figure 29. Displaying the routing table of router r3.

Step 11. Router r4 is configured similarly to router r1, but with different IP addresses (see Table 2). Those steps are summarized in the following figure. To proceed, in router r4 terminal issue the commands depicted below. Towards the end, you will verify all the directly connected networks of router r4.

```
Host: r4
root@admin:/etc/routers/r4# zebra
root@admin:/etc/routers/r4# vtysh

Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

admin# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued route, r - rejected route

C>* 192.168.34.0/30 is directly connected, r4-eth0, 00:00:06
C>* 192.168.45.0/30 is directly connected, r4-eth1, 00:00:06
admin#
```

Figure 30. Displaying routing table of router r4.

Step 12. Router r5 is configured similarly to router r1, but with different IP addresses (see Table 2). Those steps are summarized in the following figure. To proceed, in router r5's terminal issue the commands depicted below. At the end, you will verify all the directly connected networks of router r5.

```

Host: r5
root@admin:/etc/routers/r5# zebra
root@admin:/etc/routers/r5# vtysh

Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

admin# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued route, r - rejected route

C>* 192.168.25.0/30 is directly connected, r5-eth1, 00:00:06
C>* 192.168.45.0/30 is directly connected, r5-eth0, 00:00:06
admin#
    
```

Figure 31. Displaying routing table of router r5.

3 Configuring static MPLS from router r1 to router r2

In this section, you will configure static MPLS on routers so that host h1 can reach host h2. Router r1 will push a label to router r3. Routers r3, r4 and r5 will swap the labels and the data packet will reach router r2. Router r2 will pop the label and the data packet will be delivered to the destination host h2.

3.1 Pushing labels

Step 1. At this point, router r1 can reach the directly connected network 192.168.13.0/30. To communicate with other networks, you will configure static routing on router r1. To configure static routing, you need to enable the static daemon first. In router r1, type the following command to exit the vtysh session:

```
exit
```

```

Host: r1
admin# exit
root@admin:/etc/routers/r1#
    
```

Figure 32. Exiting the vtysh session.

Step 2. Type the following command to enable the static routing daemon on router r1.

```
staticd
```

```

Host: r1
admin# exit
root@admin:/etc/routers/r1# staticd
root@admin:/etc/routers/r1#
    
```

Figure 33. Starting the staticd daemon.

Step 3. In order to enter router r1 terminal, issue the following command:

```
vttysh
```

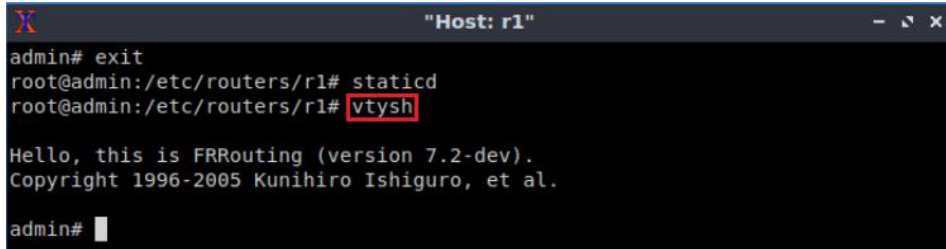


Figure 34. Starting vtysh on router r1.

Step 4. To enable router r1 configuration mode, issue the following command:

```
configure terminal
```

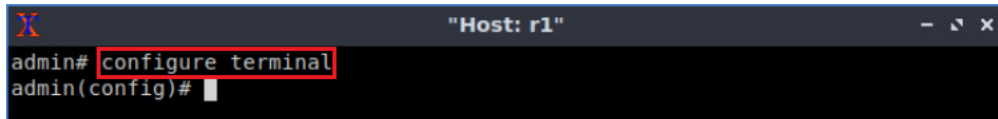


Figure 35. Enabling configuration mode on router r1.

Step 5. Type the following command to add a static route to the network 192.168.34.0/30. Router r1 will communicate with router r3 via 192.168.13.2 and the label 100 will be assigned.

```
ip route 192.168.34.0/30 192.168.13.2 label 100
```

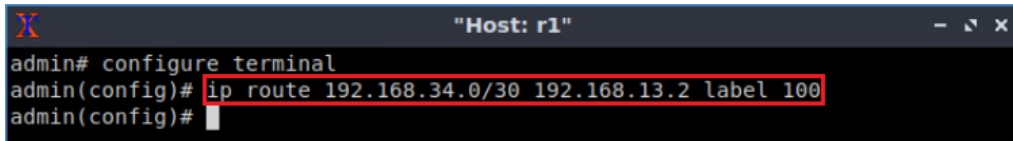


Figure 36. Pushing label for the network 192.168.34.0/30.

Step 6. Type the following command to add a static route to the network 192.168.45.0/30. Router r1 will communicate with router r3 via 192.168.13.2 and the label 100 will be assigned.

```
ip route 192.168.45.0/30 192.168.13.2 label 100
```

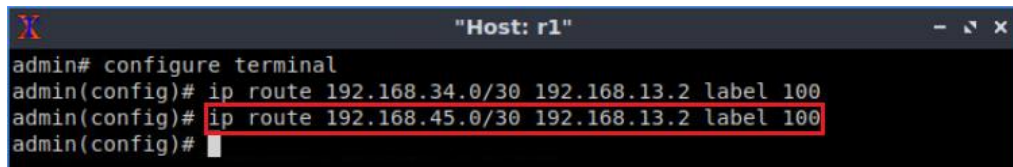
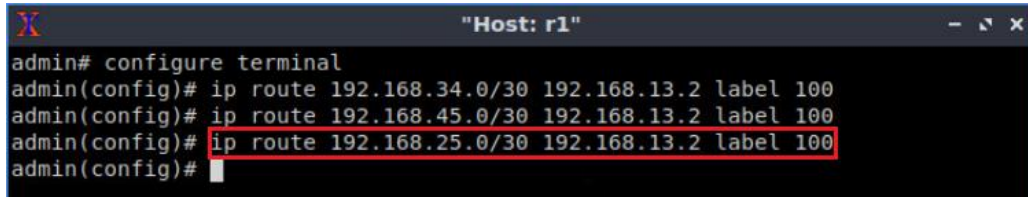


Figure 37. Pushing label for the network 192.168.45.0/30.

Step 7. Type the following command to configure static routing to the network 192.168.25.0/30. Router r1 will communicate with router r3 via 192.168.13.2 and the label 100 will be assigned.

```
ip route 192.168.25.0/30 192.168.13.2 label 100
```

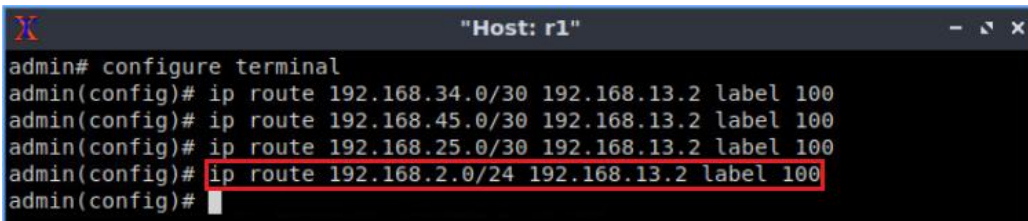


```
admin# configure terminal
admin(config)# ip route 192.168.34.0/30 192.168.13.2 label 100
admin(config)# ip route 192.168.45.0/30 192.168.13.2 label 100
admin(config)# ip route 192.168.25.0/30 192.168.13.2 label 100
admin(config)#
```

Figure 38. Pushing label for the network 192.168.25.0/30.

Step 8. Type the following command to configure static routing to the network 192.168.2.0/24. Router r1 will communicate with router r3 via 192.168.13.2 and the label 100 will be assigned.

```
ip route 192.168.2.0/24 192.168.13.2 label 100
```

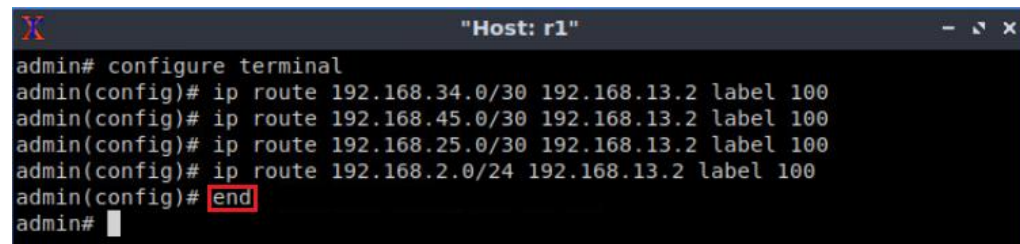


```
admin# configure terminal
admin(config)# ip route 192.168.34.0/30 192.168.13.2 label 100
admin(config)# ip route 192.168.45.0/30 192.168.13.2 label 100
admin(config)# ip route 192.168.25.0/30 192.168.13.2 label 100
admin(config)# ip route 192.168.2.0/24 192.168.13.2 label 100
admin(config)#
```

Figure 39. Pushing label for the network 192.168.2.0/24.

Step 9. Type the following command to exit configuration mode.

```
end
```

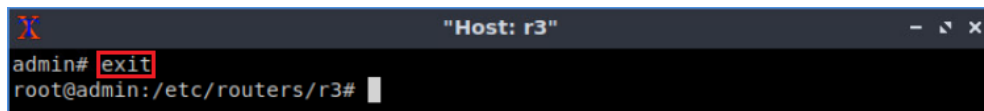


```
admin# configure terminal
admin(config)# ip route 192.168.34.0/30 192.168.13.2 label 100
admin(config)# ip route 192.168.45.0/30 192.168.13.2 label 100
admin(config)# ip route 192.168.25.0/30 192.168.13.2 label 100
admin(config)# ip route 192.168.2.0/24 192.168.13.2 label 100
admin(config)# end
admin#
```

Figure 40. Exiting configuration mode.

Step 10. In the following step, you will verify if router r1 can communicate with router r3. In router r3, type the following command to exit the vtysh session:

```
exit
```



```
admin# exit
root@admin:/etc/routers/r3#
```

Figure 41. Exiting the vtysh session.

Step 11. In router r3, type the following command in order to capture and analyze network traffic received by the interface *r3-eth0*.

```
tcpdump -i r3-eth0
```

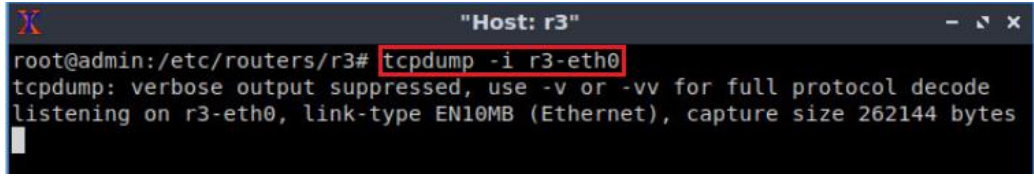


Figure 42. Enabling tcpdump to capture packets in router r3.

Step 12. Test the connectivity between router r1 and router r3 using the `ping` command. In router r1, type the command specified below. To stop the test, press `Ctrl+c`. The figure below shows a successful connectivity test.

```
ping 192.168.13.2
```

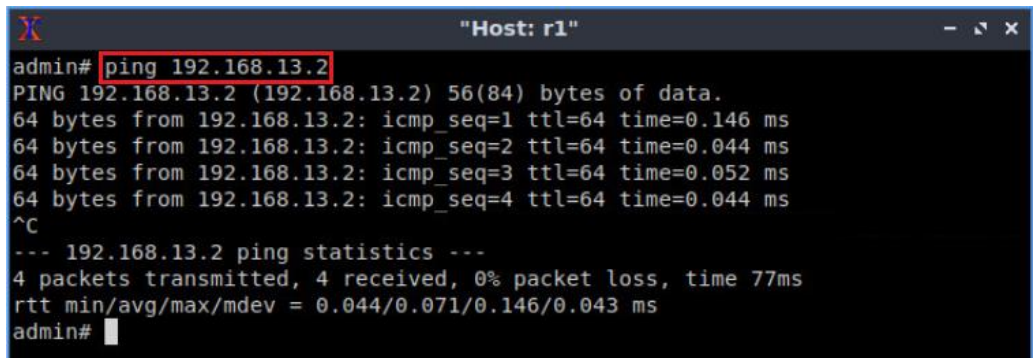


Figure 43. Output of the `ping` command in router r1.

Step 13. In router r3, you will notice that router r1 (192.168.13.1) can communicate with router r3 (192.168.13.2) through IP routing. Router r1 will not use any label to reach router r3 as the network is directly connected. To stop the test, press `Ctrl+c`.

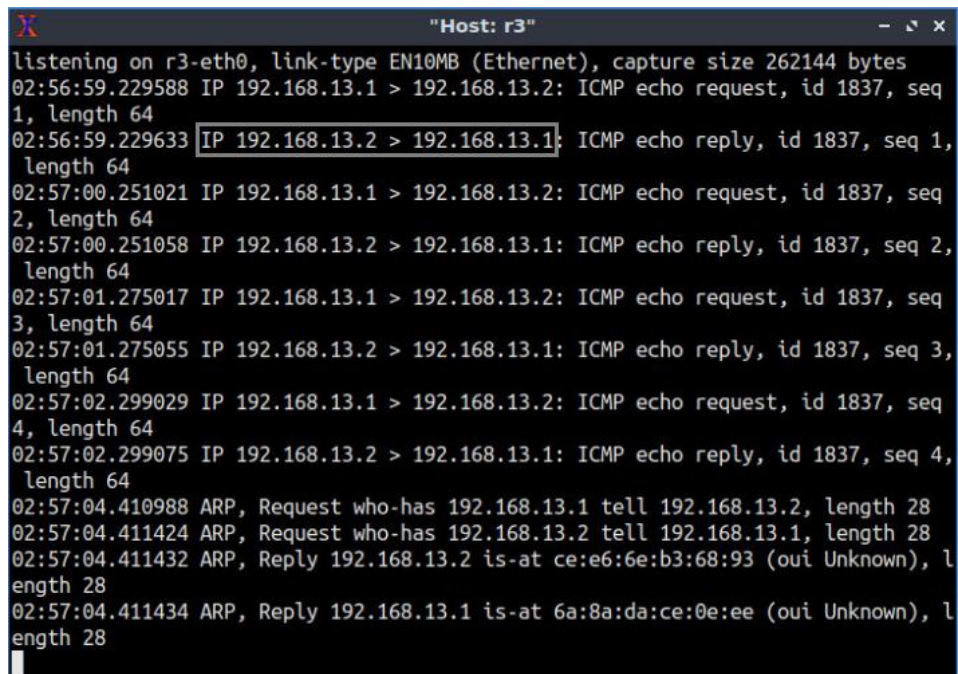


Figure 44. Verifying connectivity in router r3.

3.2 Swapping labels

Step 1. Type the following command in order to capture and analyze network traffic received by the interface, *r3-eth1*.

```
tcpdump -i r3-eth1
```

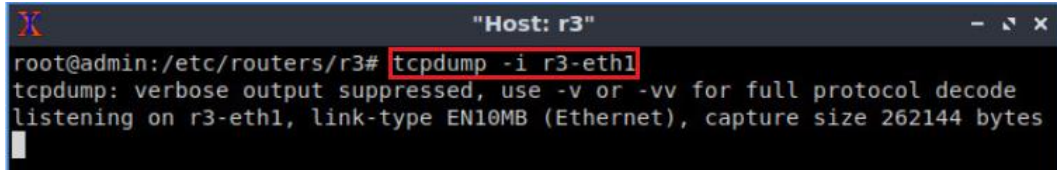


Figure 45. Enabling tcpdump to capture packets in router r3.

Step 2. Test the connectivity between router r1 and router r3 using the `ping` command. In router r1, type the command specified below. To stop the test, press `Ctrl+c`.

```
ping 192.168.34.1
```

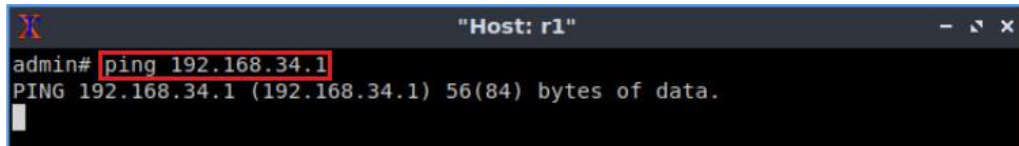


Figure 46. Output of the `ping` command in router r1.

At this point, router r1 will use label 100 to reach router r3 but router r3 cannot reach the network 192.168.34.0/30 as there is no label assigned to router r3. To stop the test, press `Ctrl+c` in router r3.

Step 3. In order to enter router r3 terminal, issue the following command:

```
vttysh
```

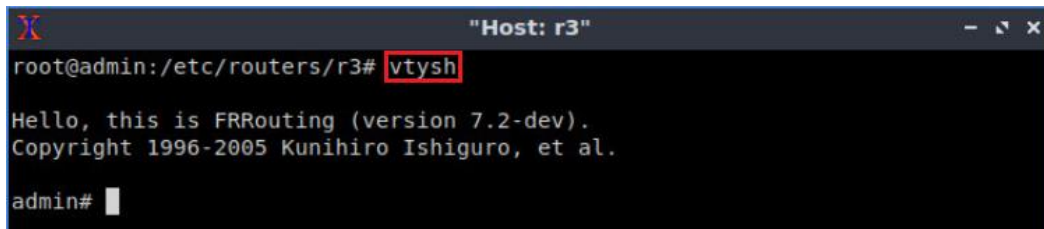
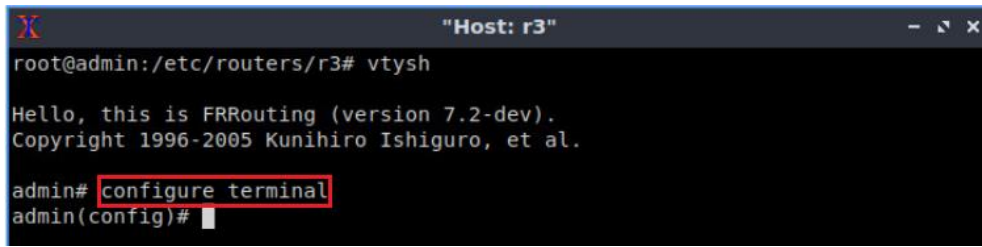


Figure 47. Starting vtysh on router r3.

Step 4. To enable router r3 configuration mode, issue the following command:

```
configure terminal
```

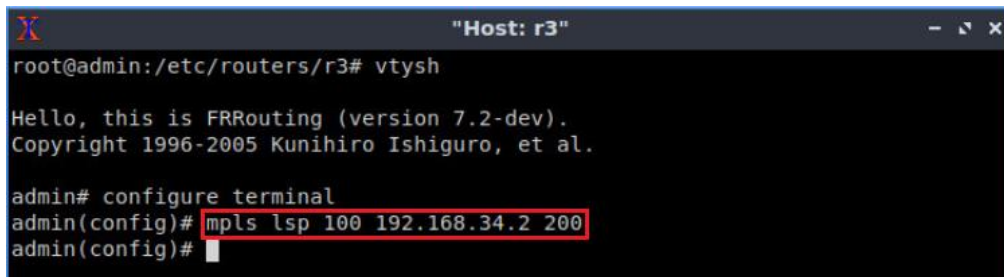


```
Host: r3
root@admin:/etc/routers/r3# vtysh
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.
admin# configure terminal
admin(config)#
```

Figure 48. Enabling configuration mode in router r3.

Step 5. In this step, you will configure label swapping in router r3. Router r3 will receive label 100 from router r1 and swap the label with another one (200) to reach the interface *r4-eth0* (192.168.34.2). Type the following command to enable label swapping in router r3.

```
mpls lsp 100 192.168.34.2 200
```

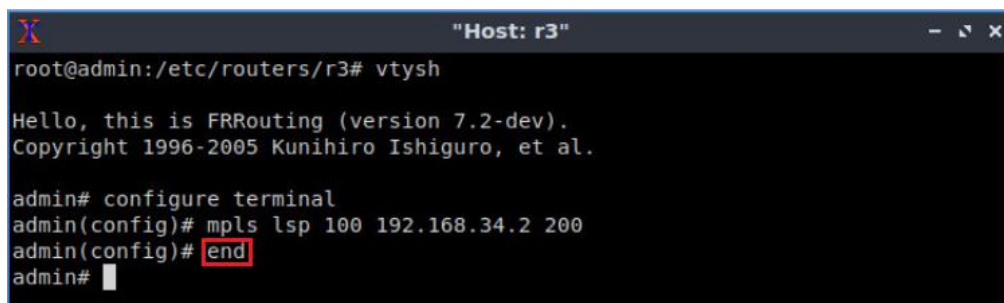


```
Host: r3
root@admin:/etc/routers/r3# vtysh
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.
admin# configure terminal
admin(config)# mpls lsp 100 192.168.34.2 200
admin(config)#
```

Figure 49. Enabling label swapping on router r3.

Step 6. Type the following command to exit configuration mode.

```
end
```

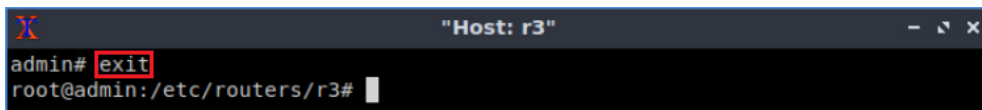


```
Host: r3
root@admin:/etc/routers/r3# vtysh
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.
admin# configure terminal
admin(config)# mpls lsp 100 192.168.34.2 200
admin(config)# end
admin#
```

Figure 50. Exiting from configuration mode.

Step 7. In the following step, you will verify that router r1 can communicate with router r3 (interface *r3-eth1*). In router r3, type the following command to exit the vtysh session:

```
exit
```



```
Host: r3
admin# exit
root@admin:/etc/routers/r3#
```

Figure 51. Exiting the vtysh session.

Step 8. Type the following command in order to capture and analyze network traffic received by the interface *r3-eth1*.

```
tcpdump -i r3-eth1
```

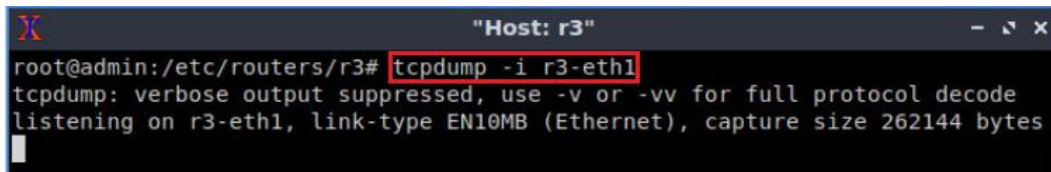


Figure 52. Enabling tcpdump to capture packets in router r3.

Step 9. Test the connectivity between router r1 and router r3 using the `ping` command. In router r1, type the command specified below. To stop the test, press `Ctrl+c`. Verify the connectivity in router r3's terminal. You will notice that router r3 is receiving packets from router r1.

```
ping 192.168.34.1
```

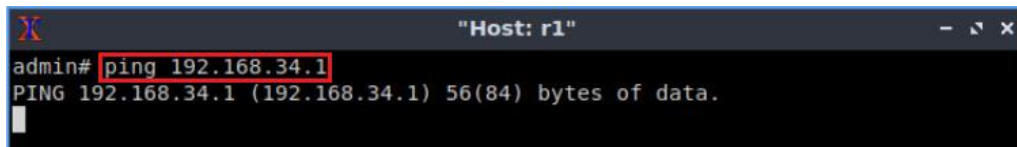


Figure 53. Output of the `ping` command in router r1.

Step 10. If you observe router r3, you will notice that router r1 (192.168.13.1) will communicate with router r3 (192.168.34.1) through MPLS routing. Router r1 uses label 100 to reach router r3 and router r3 swaps the label to 200 to reach network 192.168.34.0/30. To stop the test, press `Ctrl+c`.

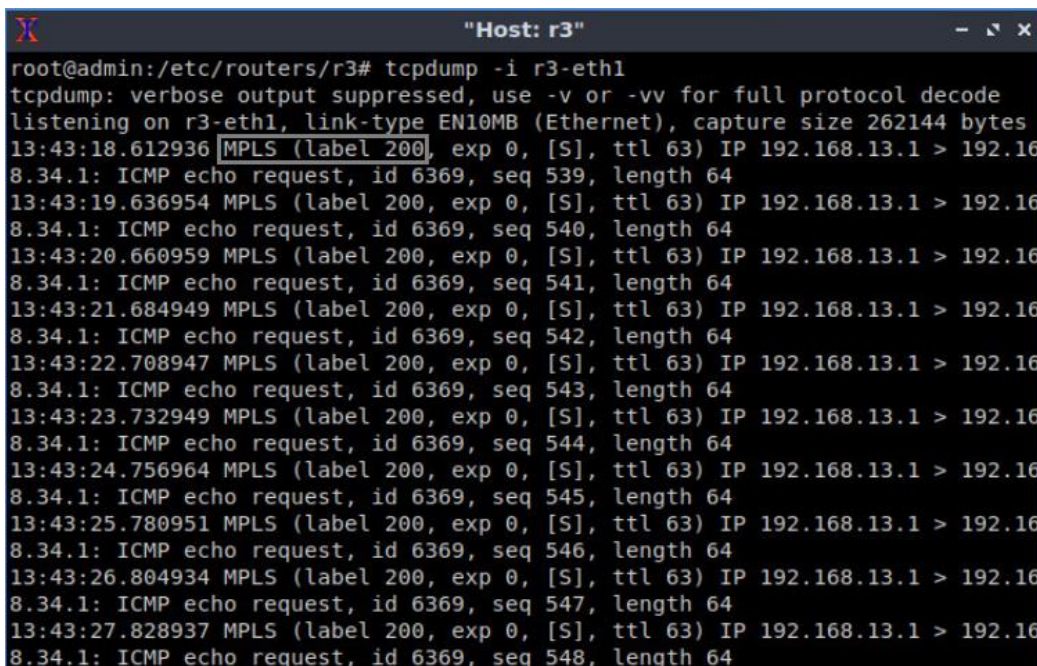


Figure 54. Verifying connectivity in router r3.

At this point, an MPLS packet can reach the network 192.168.34.0/30. Router r4 will not be able to connect to the network 192.168.45.0/30 as there is no label assigned for the network.

Step 11. To enable router r4 configuration mode, issue the following command:

```
configure terminal
```

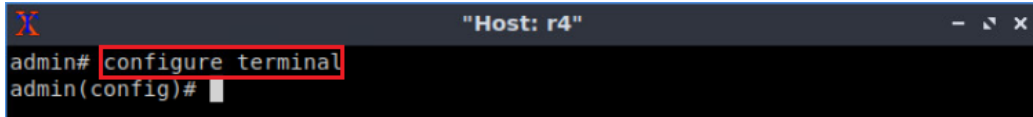


Figure 55. Enabling configuration mode in router r4.

Step 12. In this step, you will configure label swapping in router r4. Router r4 will receive label 200 from router r3 and swap the label with another one (300) to reach the interface r5-eth0 (192.168.45.2). Type the following command to enable label swapping on router r4.

```
mpls lsp 200 192.168.45.2 300
```

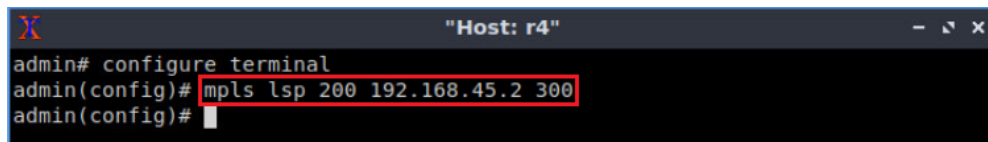


Figure 56. Enabling label swapping in router r4.

Step 13. Type the following command to exit configuration mode.

```
end
```

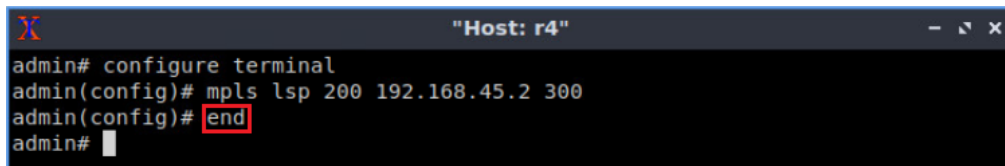


Figure 57. Exiting from configuration mode.

Step 14. In the following step, you will verify that router r1 can communicate with router r4 (interface r4-eth1). In router r4, type the following command to exit the vtysh session:

```
exit
```

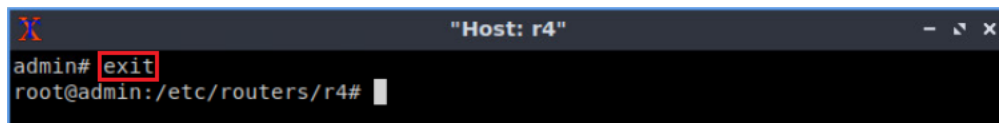


Figure 58. Exiting the vtysh session.

Step 15. Type the following command in order to capture and analyze network traffic received by the interface r4-eth1.

```
tcpdump -i r4-eth1
```

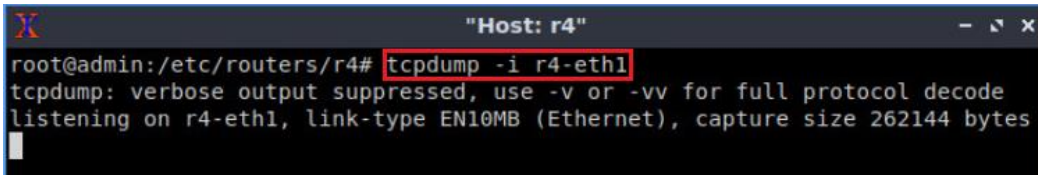


Figure 59. Enabling tcpdump to capture packets in router r4.

Step 16. Test the connectivity between router r1 and router r4 using the `ping` command. In router r1, type the command specified below. To stop the test, press `Ctrl+c`. Verify the connectivity in router r4's terminal. You will notice that router r4 is receiving packets from router r1.

```
ping 192.168.45.1
```

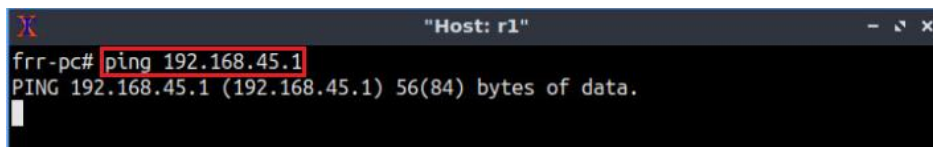


Figure 60. Output of the `ping` command on router r1.

Step 17. If you observe router r4, you will notice that router r1 (192.168.13.1) will communicate with router r4 (192.168.45.1) through MPLS routing. Router r1 uses label 100 to reach router r3 and router r3 swaps the label to 200 to reach network 192.168.34.0/30. Router r4 receives label 200 and swaps to label 300 to reach the network 192.168.45.0/30. To stop the test, press `Ctrl+c`.

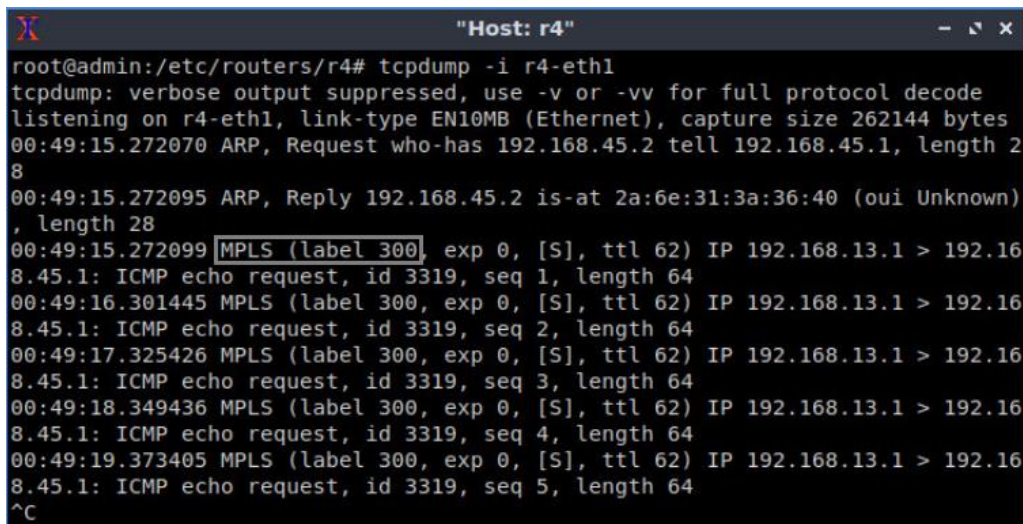


Figure 61. Verifying connectivity in router r4.

At this point, an MPLS packet can reach the network 192.168.45.0/30. Router r5 needs to assign label for the network 192.168.25.0/30.

Step 18. To enable router r5 configuration mode, issue the following command:

```
configure terminal
```

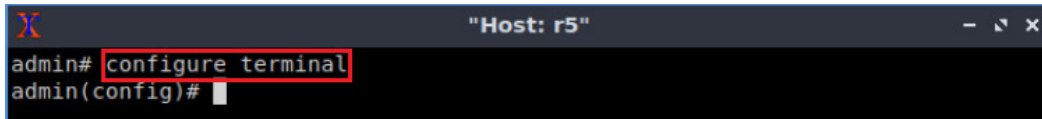


Figure 62. Enabling configuration mode in router r5.

Step 19. In this step, you will configure label swapping in router r5. Router r5 will receive label 300 from router r4 and swap the label with another one (220) to reach the interface *r2-eth1* (192.168.25.1). Type the following command to enable label swapping in router r4.

```
mpls lsp 300 192.168.25.1 220
```

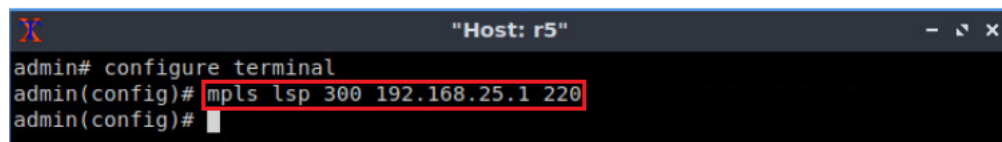


Figure 63. Enabling label swapping in router r5.

Step 20. Type the following command to exit configuration mode.

```
end
```

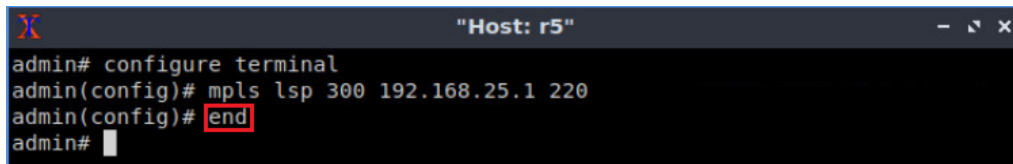


Figure 64. Exiting configuration mode.

Step 21. In the following step, you will verify that router r1 can communicate with router r5 (interface *r5-eth1*). In router r5, type the following command to exit the vtysh session:

```
exit
```

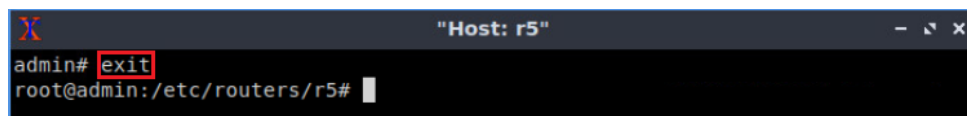


Figure 65. Exiting the vtysh session.

Step 22. Type the following command in order to capture and analyze network traffic received by the interface *r5-eth1*.

```
tcpdump -i r5-eth1
```

```

Host: r5
root@admin:/etc/routers/r5# tcpdump -i r5-eth1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on r5-eth1, link-type EN10MB (Ethernet), capture size 262144 bytes

```

Figure 66. Enabling tcpdump to capture packets on router r5.

Step 23. Test the connectivity between router r1 and router r5 using the `ping` command. In router r1, type the command specified below. To stop the test, press `Ctrl+c`. Verify the connectivity in router r5's terminal. You will notice that router r5 is receiving packets from router r1.

```
ping 192.168.25.2
```

```

Host: r1
admin# ping 192.168.25.2
PING 192.168.25.2 (192.168.25.2) 56(84) bytes of data.

```

Figure 67. Output of the `ping` command on router r1.

Step 24. If you observe router r5, you will notice that router r1 (192.168.13.1) can communicate with router r5 (192.168.25.2) through MPLS routing. Router r1 uses label 100 to reach router r3 and router r3 swaps the label to 200 to reach network 192.168.34.0/30. Router r4 receives label 200 and swaps to label 300 to reach the network 192.168.45.0/30. Finally, router r5 swaps the label to 220 and delivers the packet to router r2. To stop the test, press `Ctrl+c`.

```

Host: r5
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on r5-eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
00:52:55.225366 ARP, Request who-has 192.168.25.1 tell 192.168.25.2, length 28
00:52:55.225405 ARP, Reply 192.168.25.1 is-at 9e:de:eb:a4:00:f4 (oui Unknown), length 28
00:52:55.225410 MPLS (label 220, exp 0, [S], ttl 61) IP 192.168.13.1 > 192.168.25.2: ICMP echo request, id 3323, seq 1, length 64
00:52:56.238604 MPLS (label 220, exp 0, [S], ttl 61) IP 192.168.13.1 > 192.168.25.2: ICMP echo request, id 3323, seq 2, length 64
00:52:57.261429 MPLS (label 220, exp 0, [S], ttl 61) IP 192.168.13.1 > 192.168.25.2: ICMP echo request, id 3323, seq 3, length 64
00:52:58.287236 MPLS (label 220, exp 0, [S], ttl 61) IP 192.168.13.1 > 192.168.25.2: ICMP echo request, id 3323, seq 4, length 64
00:52:59.309461 MPLS (label 220, exp 0, [S], ttl 61) IP 192.168.13.1 > 192.168.25.2: ICMP echo request, id 3323, seq 5, length 64
00:53:00.333436 MPLS (label 220, exp 0, [S], ttl 61) IP 192.168.13.1 > 192.168.25.2: ICMP echo request, id 3323, seq 6, length 64
^C

```

Figure 68. Verifying connectivity in router r5.

3.3 Popping labels

Step 1. To enable router r2 configuration mode, issue the following command:

```
configure terminal
```

```
Host: r2
admin# configure terminal
admin(config)#
```

Figure 69. Enabling configuration mode in router r2.

Step 2. Type the following command to assign label 220 in router r2. The label is the same as the one router r2 received from router r5. Router r2 will pop the label (220) and the IP packet will be delivered to the destination host, h2 (192.168.2.10).

```
mpls lsp 220 192.168.2.10 implicit-null
```

```
Host: r2
admin# configure terminal
admin(config)# mpls lsp 220 192.168.2.10 implicit-null
admin(config)#
```

Figure 70. Popping the label in router r2.

The keyword *implicit-null* indicates that the router will not perform a swap; instead it performs a pop and delivers the IP packet to the destination.

Step 3. Type the following command to exit configuration mode.

```
end
```

```
Host: r2
admin# configure terminal
admin(config)# mpls lsp 220 192.168.2.10 implicit-null
admin(config)# end
admin#
```

Figure 71. Exiting configuration mode.

Step 4. Type the following command to exit the vtysh session:

```
exit
```

```
Host: r2
admin# exit
root@admin:/etc/routers/r2#
```

Figure 72. Exiting the vtysh session.

Step 5. Type the following command in order to capture and analyze network traffic received by the interface *r2-eth1*.

```
tcpdump -i r2-eth1
```

```

"Host: r2"
root@admin:/etc/routers/r2# tcpdump -i r2-eth1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on r2-eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
    
```

Figure 73. Enabling tcpdump to capture packets on router r2.

Step 6. Test the connectivity between router r1 and router r2 using the `ping` command. In router r1, type the command specified below. To stop the test, press `Ctrl+c`. Verify the connectivity in router r2 terminal. You will notice that router r2 is receiving packets from router r1.

```
ping 192.168.25.1
```

```

"Host: r1"
admin# ping 192.168.25.1
PING 192.168.25.1 (192.168.25.1) 56(84) bytes of data.
    
```

Figure 74. Output of the `ping` command on router r1.

Step 7. If you observe router r2, you will notice that label 220 is used to reach router r2. To stop the test, press `Ctrl+c`.

```

"Host: r2"
root@admin:/etc/routers/r2# tcpdump -i r2-eth1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on r2-eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
00:56:18.338965 MPLS (label 220, exp 0, [S], ttl 61) IP 192.168.13.1 > 192.16
8.25.1: ICMP echo request, id 3327, seq 1, length 64
00:56:19.341438 MPLS (label 220, exp 0, [S], ttl 61) IP 192.168.13.1 > 192.16
8.25.1: ICMP echo request, id 3327, seq 2, length 64
00:56:20.365472 MPLS (label 220, exp 0, [S], ttl 61) IP 192.168.13.1 > 192.16
8.25.1: ICMP echo request, id 3327, seq 3, length 64
00:56:21.389481 MPLS (label 220, exp 0, [S], ttl 61) IP 192.168.13.1 > 192.16
8.25.1: ICMP echo request, id 3327, seq 4, length 64
00:56:22.413419 MPLS (label 220, exp 0, [S], ttl 61) IP 192.168.13.1 > 192.16
8.25.1: ICMP echo request, id 3327, seq 5, length 64
00:56:23.437417 MPLS (label 220, exp 0, [S], ttl 61) IP 192.168.13.1 > 192.16
8.25.1: ICMP echo request, id 3327, seq 6, length 64
00:56:23.469349 ARP, Request who-has 192.168.25.1 tell 192.168.25.2, length 2
8
00:56:23.469442 ARP, Reply 192.168.25.1 is-at 9e:de:eb:a4:00:f4 (oui Unknown)
, length 28
    
```

Figure 75. Verifying connectivity in router r2.

Step 8. Type the following command in order to capture and analyze network traffic received by the interface `r2-eth0`.

```
tcpdump -i r2-eth0
```

```

"Host: r2"
root@admin:/etc/routers/r2# tcpdump -i r2-eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on r2-eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
    
```

Figure 76. Enabling tcpdump to capture packets on router r2.

Step 9. Test the connectivity between router r1 and router r2 using the `ping` command. In router r1, type the command specified below. To stop the test, press `Ctrl+c`. Verify the connectivity in router r2 terminal. You will notice that router r2 is receiving packets from router r1.

```
ping 192.168.2.1
```

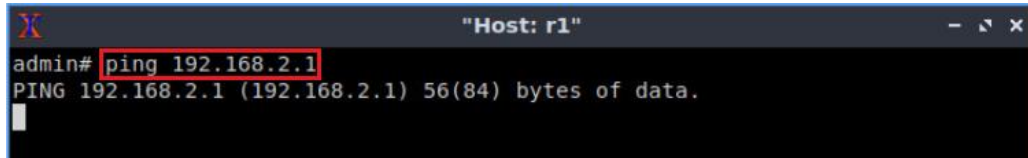


Figure 77. Output of the `ping` command on router r1.

Step 10. If you observe router r2, you will notice that router r1 (192.168.13.1) can communicate with router r2 (192.168.2.1). You will notice there is no MPLS label attached to it as router r2 popped the label and delivered the IP packet to the destination.

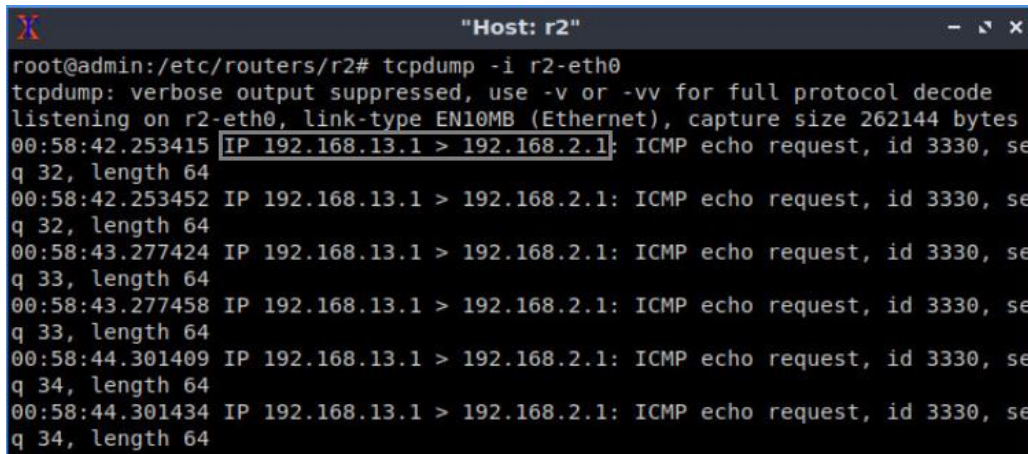


Figure 78. Verifying connectivity in router r2.

Step 11. Test the connectivity between router r1 and host h2 using the `ping` command. In router r1, type the command specified below. To stop the test, press `Ctrl+c`. Verify the connectivity in router r2 terminal. You will notice that router r2 is receiving packets from router r1.

```
ping 192.168.2.10
```

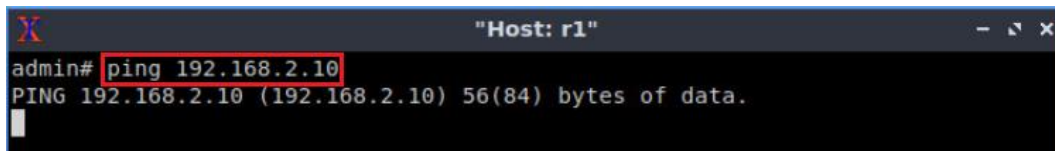


Figure 79. Output of the `ping` command on router r1.

Step 12. If you observe router r2, you will notice that router r1 (192.168.13.1) can finally reach the destination host, h2 (192.168.2.10).

```
tcpdump -i r2-eth0
```

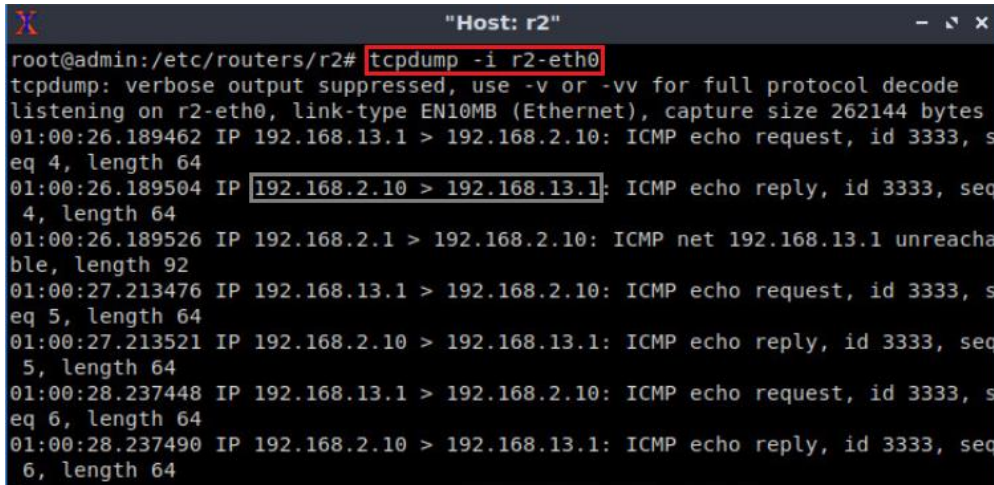


Figure 80. Verifying connectivity in router r2.

Step 13. Test the connectivity between host h1 and host h2 using the `ping` command. In host h1, type the command specified below. To stop the test, press `Ctrl+c`.

```
ping 192.168.2.10
```

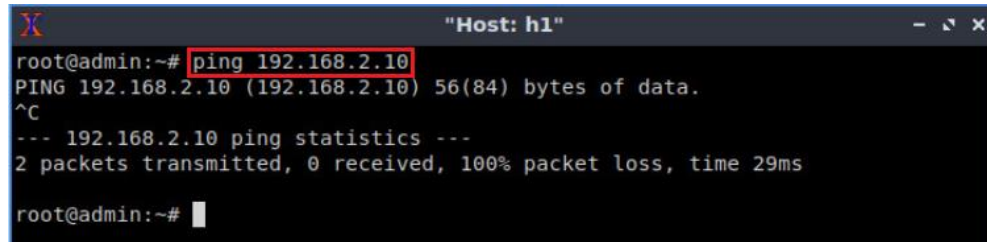


Figure 81. Output of the `ping` command on host h2.

Consider Figure 81. Host h1 cannot ping host h2 as the IP packet reaches host h2 but cannot reply back to host h1 as router r2 does not have the route back to router r1.

4 Configuring static MPLS from router r2 to router r1

In this section, you will configure static MPLS so that router r2 can communicate with all the routers and both the hosts, h1 and h2 can ping each other.

Step 1. Router r2 has a route to the network 192.168.25.0/30. Configure static routing with label 400 in router r1 in order to get the routes to other networks. All the steps are summarized in the following figure.


```
Host: r2
root@admin:/etc/routers/r2# staticd
root@admin:/etc/routers/r2# vtysh

Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

admin# configure terminal
admin(config)# ip route 192.168.45.0/30 192.168.25.2 label 400
admin(config)# ip route 192.168.34.0/30 192.168.25.2 label 400
admin(config)# ip route 192.168.13.0/30 192.168.25.2 label 400
admin(config)# ip route 192.168.1.0/24 192.168.25.2 label 400
admin(config)# end
admin#
```

Figure 82. Pushing labels in router r2.

Step 2. Type the following commands in router r5. Router r5 will receive label 400, swap the label to 300 which will be delivered to router r4 (192.168.45.1). All the commands are summarized in the following figure.

```
mpls lsp 400 192.168.45.1 300
```

```
Host: r5
root@admin:/etc/routers/r5# vtysh

Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

admin# configure terminal
admin(config)# mpls lsp 400 192.168.45.1 300
admin(config)# end
admin#
```

Figure 83. Enabling label swapping in router r5.

Step 3. Type the following commands in router r4 terminal. Router r4 will receive label 300, swap the label to 200 which will be delivered to router r3 (192.168.34.1). All the commands are summarized in the following figure.

```
mpls lsp 300 192.168.34.1 200
```

```
Host: r4
root@admin:/etc/routers/r4# vtysh

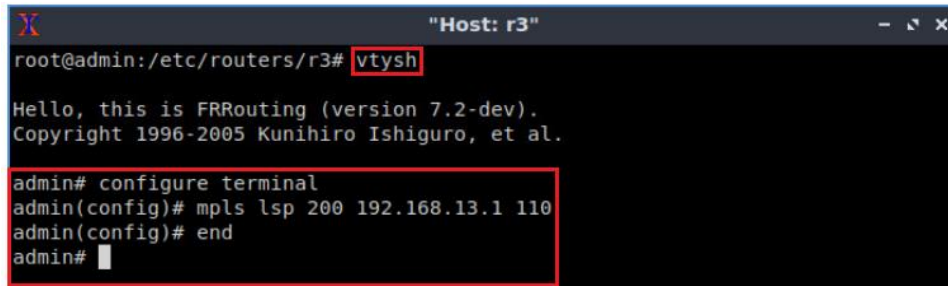
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

admin# configure terminal
admin(config)# mpls lsp 300 192.168.34.1 200
admin(config)# end
admin#
```

Figure 84. Enabling label swapping in router r4.

Step 4. Type the following commands in router r3 terminal. Router r3 will receive label 200, swap the label to 110 which will be delivered to router r1 (192.168.13.1). All the commands are summarized in the following figure.

```
mpls lsp 200 192.168.13.1 110
```



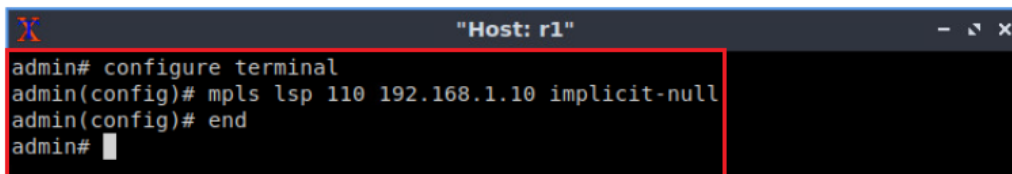
```
root@admin:/etc/routers/r3# vtysh
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

admin# configure terminal
admin(config)# mpls lsp 200 192.168.13.1 110
admin(config)# end
admin#
```

Figure 85. Enabling label swapping in router r3.

Step 5. Type the following commands to assign label 110 to router r1. The label is the same as the one router r1 received from router r3. Router r1 will pop the label (110) and the IP packet will be delivered to the destination host, h1 (192.168.1.10).

```
mpls lsp 110 192.168.1.10 implicit-null
```



```
admin# configure terminal
admin(config)# mpls lsp 110 192.168.1.10 implicit-null
admin(config)# end
admin#
```

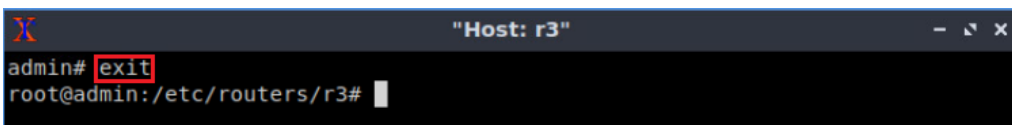
Figure 86. Popping the label in router r1.

5 Verifying configuration

In this section, you will verify the MPLS configuration and the connectivity between two hosts.

Step 1. You will verify MPLS configuration in router r3. Router r3 uses the MPLS label 200 to reach interface *r3-eth1* (192.168.34.1). Type the following command to exit the vtysh session in router r3:

```
exit
```



```
admin# exit
root@admin:/etc/routers/r3#
```

Figure 87. Exiting configuration mode.

Step 2. Type the following command in order to capture and analyze network traffic received by the interface *r3-eth1*.

```
tcpdump -i r3-eth1
```

```

"Host: r3"
root@admin:/etc/routers/r3# tcpdump -i r3-eth1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on r3-eth1, link-type EN10MB (Ethernet), capture size 262144 bytes

```

Figure 88. Enabling tcpdump to capture packets in router r3.

Step 3. Test the connectivity between router r2 and router r3 using the `ping` command. In router r2, type the command specified below. To stop the test, press `Ctrl+c`. Verify the connectivity in router r3's terminal. You will notice that router r3 is receiving packets from router r2.

```
ping 192.168.34.1
```

```

"Host: r2"
admin# ping 192.168.34.1
PING 192.168.34.1 (192.168.34.1) 56(84) bytes of data.

```

Figure 89. Output of the `ping` command on router r2.

Step 4. If you observe router r3, you will notice that router r2 (192.168.25.1) can communicate with router r3 (192.168.34.1) through MPLS routing. Router r2 uses label 400 to reach router r5 and router r5 swaps the label to 300 to reach router r4. Router r4 uses label 200 to reach network 192.168.34.0/30. To stop the test, press `Ctrl+c`.

```

"Host: r3"
root@admin:/etc/routers/r3# tcpdump -i r3-eth1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on r3-eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
01:18:05.421426 MPLS (label 200, exp 0, [S], ttl 62) IP 192.168.25.1 > 192.16
8.34.1: ICMP echo request, id 3365, seq 12, length 64
01:18:05.421480 MPLS (label 200, exp 0, [S], ttl 57) IP 192.168.25.1 > 192.16
8.34.1: ICMP echo request, id 3365, seq 12, length 64
01:18:06.445465 MPLS (label 200, exp 0, [S], ttl 62) IP 192.168.25.1 > 192.16
8.34.1: ICMP echo request, id 3365, seq 13, length 64
01:18:06.445513 MPLS (label 200, exp 0, [S], ttl 57) IP 192.168.25.1 > 192.16
8.34.1: ICMP echo request, id 3365, seq 13, length 64

```

Figure 90. Verifying connectivity in router r3.

Step 5. You can also verify the configuration through wireshark. Type the following command in router r3 to open wireshark.

```
wireshark
```

```

"Host: r3"
root@admin:/etc/routers/r3# wireshark

```

Figure 91. Opening wireshark in router r3.

Step 6. Click on `r3-eth1` in order to see all the traffic received by the interface `r3-eth1`.

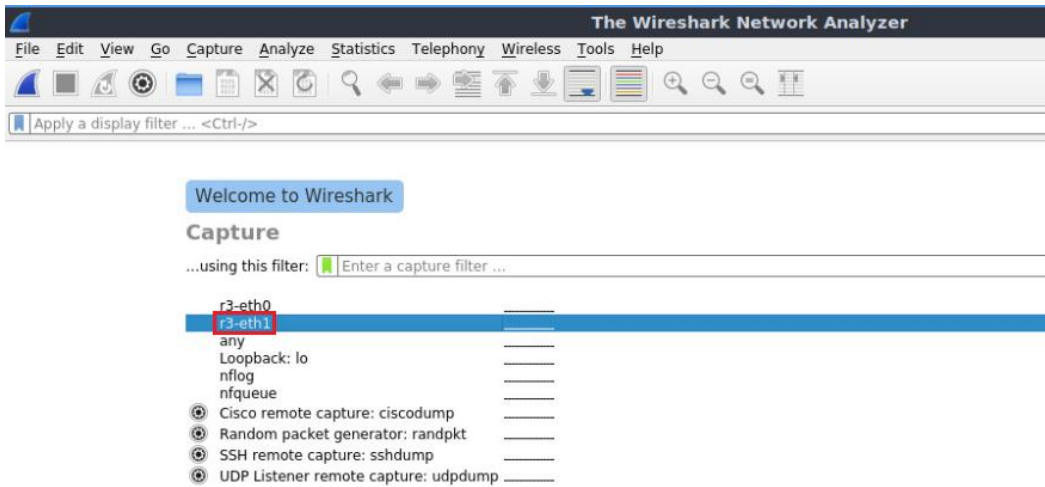


Figure 92. Selecting interface *r3-eth1* in wireshark.

Step 7. Test the connectivity between router r2 and router r3 using the `ping` command. In router r2, type the command specified below. To stop the test, press `Ctrl+c`. Verify the connectivity on wireshark.

```
ping 192.168.34.1
```

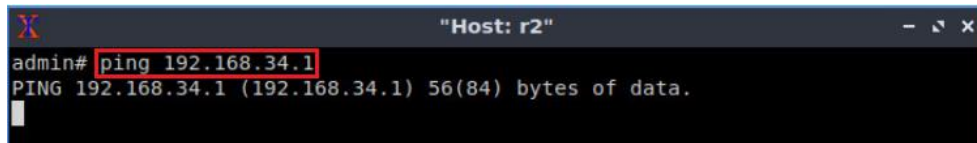


Figure 93. Output of the `ping` command on router r2.

Step 8. If you verify packets on wireshark, you will notice the source address, 192.168.25.1 can ping to the destination address, 192.168.34.1. You can also notice the MPLS header used to reach the destination, 192.168.34.1 (label: 200, Exp: 0, S: 1, TTL: 62).

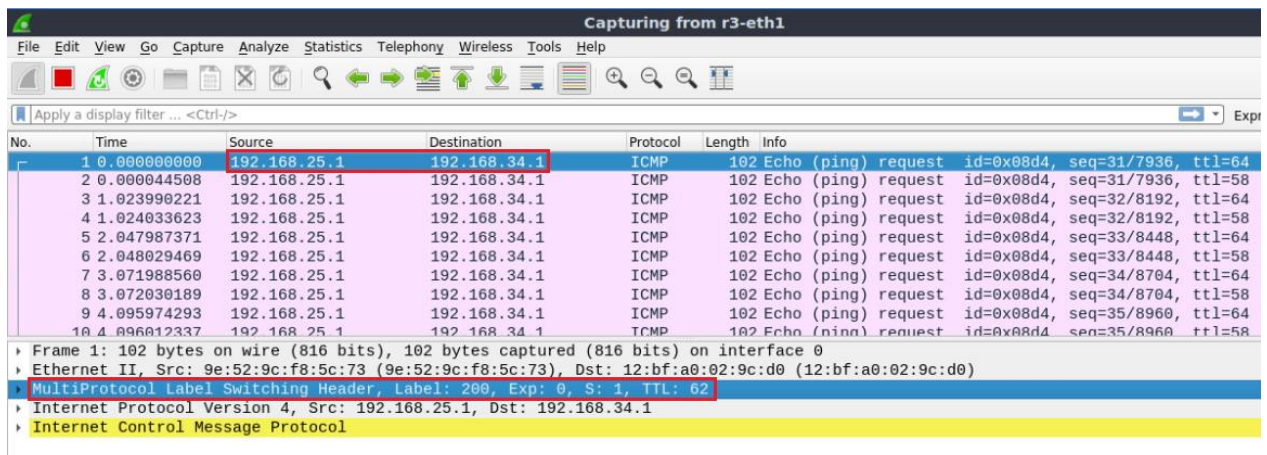


Figure 94. Output of the `ping` command in wireshark.

Step 9. Type the following command to display the MPLS table of router r1. You will notice the label assigned to router r1.

```
show mpls table
```

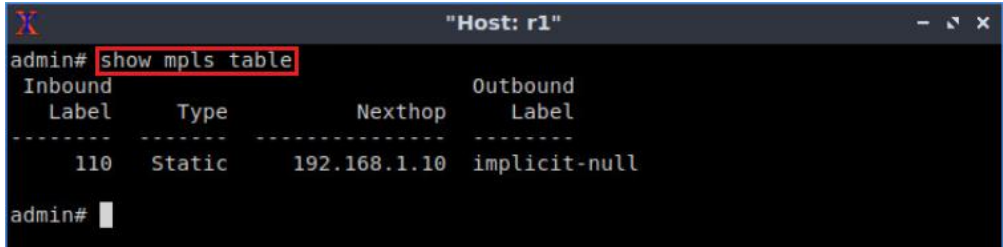


Figure 95. Displaying the MPLS table of router r1.

Step 10. Type the following command to display the routing table (control plane) of router r1. The control plane makes use of a label exchange protocol to create and maintain labels internally, and to exchange these labels with other devices.

```
show ip route
```

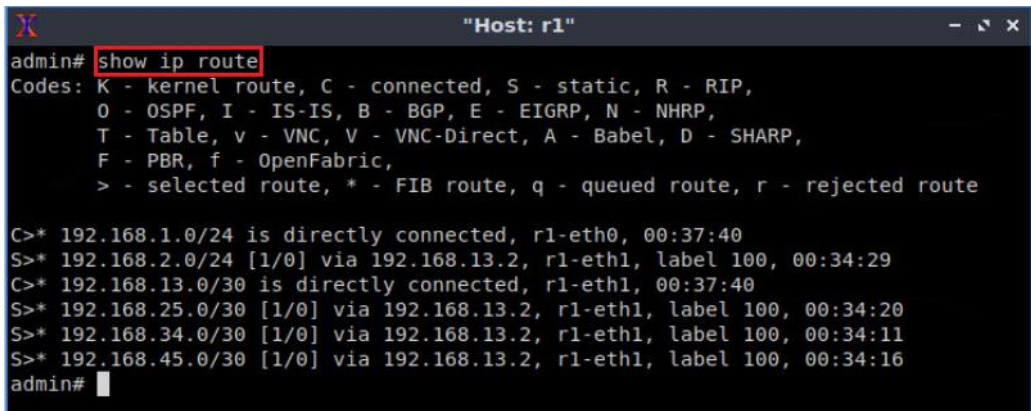


Figure 96. displaying the routing table of router r1.

Step 11. Type the following command to exit the vtysh session:

```
exit
```

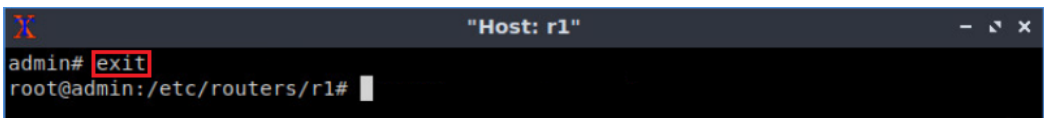


Figure 97. Exiting the vtysh session.

Step 12. Type the following command to display the kernel routes (data plane) in router r1. The data plane indicates how the data will flow from router r1 (192.168.1.0/24) to router r2 (192.168.2.0/24).

```
route
```

```

Host: r1
root@admin:/etc/routers/r1# route
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.1.0     0.0.0.0         255.255.255.0   U        0      0      0 r1-eth0
192.168.2.0     192.168.13.2   255.255.255.0   UG       20     0      0 r1-eth1
192.168.13.0    0.0.0.0         255.255.255.252 U        0      0      0 r1-eth1
192.168.25.0    192.168.13.2   255.255.255.252 UG       20     0      0 r1-eth1
192.168.34.0    192.168.13.2   255.255.255.252 UG       20     0      0 r1-eth1
192.168.45.0    192.168.13.2   255.255.255.252 UG       20     0      0 r1-eth1
root@admin:/etc/routers/r1#
    
```

Figure 98. Displaying kernel routes in router r1.

Consider Figure 98. An IP lookup for the destination, 192.168.2.0/24 will be examined at this point. Other routers (r2, r3, r4 and r5) will forward the packet according to label information. The label information is removed at the exit router (r2), and the IP packet will be delivered to the destination, 192.168.2.10.

Step 13. Test the connectivity between host h1 and host h2 using the `ping` command. In host h1, type the command specified below. To stop the test, press `Ctrl+c`. The figure below shows a successful connectivity test.

```
ping 192.168.2.10
```

```

Host: h1
root@admin:~# ping 192.168.2.10
PING 192.168.2.10 (192.168.2.10) 56(84) bytes of data:
64 bytes from 192.168.2.10: icmp_seq=1 ttl=59 time=0.125 ms
64 bytes from 192.168.2.10: icmp_seq=2 ttl=59 time=0.104 ms
64 bytes from 192.168.2.10: icmp_seq=3 ttl=59 time=0.114 ms
^C
--- 192.168.2.10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 41ms
rtt min/avg/max/mdev = 0.104/0.114/0.125/0.012 ms
root@admin:~#
    
```

Figure 99. Output of the `ping` command in host h2.

This concludes Lab 3. Stop the emulation and then exit out of MiniEdit.

References

1. Paul Goransson, Chuck Black, *“Software Defined Networks: A Comprehensive Approach”*, 2014.
2. Luc De Ghein, *“MPLS Fundamentals”*, 1st Edition, Pearson, 2006.
3. Benzekki, Kamal; El Fergougui, Abdeslam; Elbelrhiti Elalaoui, Abdelbaki, *“Software-defined networking (SDN): A survey”*, 2016.
4. Juniper, *“MPLS Label Distribution Protocols Overview”*, 2018. [Online]. Available: https://www.juniper.net/documentation/en_US/junose15.1/topics/concept/mpls-label-distribution-protocols.html

5. Linux foundation collaborative projects, "*FRR Routing documentation*", 2017. [Online]. Available: <http://docs.frrouting.org/en/latest/zebra.html#mpls-commands>
6. Juniper, "*Understanding MPLS Label Operations*", 2020. [Online]. Available: https://www.juniper.net/documentation/en_US/junos/topics/concept/mpls-label-operations-qfx-series.html
7. B. Lantz, G. Gee, "*MiniEdit: a simple network editor for Mininet*," 2013. [Online]. Available: <https://github.com/Mininet/Mininet/blob/master/examples>.