



UNIVERSITY OF  
**SOUTH CAROLINA**

**OPEN SHORTEST PATH FIRST  
LAB SERIES**

Book Version: **10-24-2021**

Principal Investigator: Jorge Crichigno



Award 1829698

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

## Contents

- Lab 1: Introduction to Mininet
- Lab 2: Introduction to FRR
- Lab 3: Configuring Single-Area OSPFv2
- Lab 4: Configuring Multi-Area OSPFv2
- Exercise 1: Configuring Multi-Area OSPFv2
- Lab 5: Configuring OSPFv2 with Default Route
- Lab 6: OSPFv2 Virtual Link
- Exercise 2: Configuring OSPFv2 Virtual Link
- Lab 7: OSPFv2 Authentication
- Lab 8: Setting OSPFv2 Route Cost
- Lab 9: Configuring Multi-Area OSPFv3
- Exercise 3: Configuring Multi-Area OSPFv3
- Lab 10: Configuring Dual Stack OSPF Routing



UNIVERSITY OF  
**SOUTH CAROLINA**

## **OPEN SHORTEST PATH FIRST**

### **Lab 1: Introduction to Mininet**

Document Version: **10-11-2021**



Award 1829698

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

## Contents

Overview .....	3
Objectives.....	3
Lab settings .....	3
Lab roadmap .....	3
1 Introduction to Mininet .....	3
2 Invoke Mininet using the CLI .....	5
2.1 Invoke Mininet using the default topology.....	5
2.2 Test connectivity .....	9
3 Build and emulate a network in Mininet using the GUI .....	10
3.1 Build the network topology .....	10
3.2 Test connectivity .....	12
3.3 Automatic assignment of IP addresses .....	15
3.4 Save and load a Mininet topology .....	18
4 Configure router r1 .....	19
4.1 Verify end-hosts configuration.....	20
4.2 Configure router's interface.....	21
4.3 Verify router r1 configuration .....	25
4.4 Test connectivity between end-hosts.....	26
References .....	26

## Overview

This lab provides an introduction to Mininet, a virtual testbed used for testing network tools and protocols. It demonstrates how to invoke Mininet from the command-line interface (CLI) utility and how to build and emulate topologies using a graphical user interface (GUI) application.

## Objectives

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

1. Understand what Mininet is and why it is useful for testing network topologies.
2. Invoke Mininet from the CLI.
3. Construct network topologies using the GUI.
4. Save/load Mininet topologies using the GUI.
5. Configure the interfaces of a router using the CLI.

## Lab settings

The information in Table 1 provides the credentials of the Client machine.

Table 1. Credentials to access the Client machine.

Device	Account	Password
Client	admin	password

## Lab roadmap

This lab is organized as follows:

1. Section 1: Introduction to Mininet.
2. Section 2: Invoke Mininet using the CLI.
3. Section 3: Build and emulate a network in Mininet using the GUI.
4. Section 4: Configure router r1.

### 1 Introduction to Mininet

Mininet is a virtual testbed enabling the development and testing of network tools and protocols. With a single command, Mininet can create a realistic virtual network on any type of machine (Virtual Machine (VM), cloud-hosted, or native). Therefore, it provides an inexpensive solution and streamlined development running in line with production networks<sup>1</sup>. Mininet offers the following features:

- Fast prototyping for new networking protocols.
- Simplified testing for complex topologies without the need of buying expensive hardware.
- Realistic execution as it runs real code on the Unix and Linux kernels.
- Open-source environment backed by a large community contributing extensive documentation.

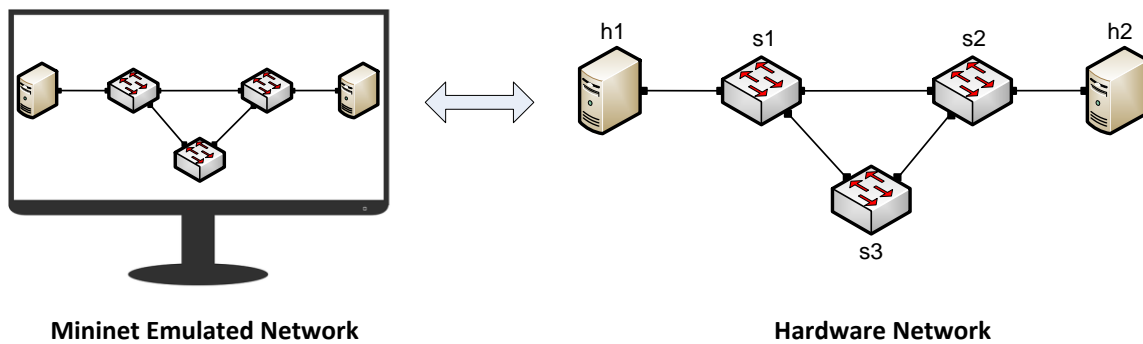


Figure 1. Hardware network vs. Mininet emulated network.

Mininet is useful for development, teaching, and research as it is easy to customize and interact with it through the CLI or the GUI. Mininet was originally designed to experiment with *OpenFlow*<sup>2</sup> and *Software-Defined Networking (SDN)*<sup>3</sup>. This lab, however, only focuses on emulating a simple network environment without SDN-based devices.

Mininet’s logical nodes can be connected into networks. These nodes are sometimes called containers, or more accurately, *network namespaces*. Containers consume sufficiently fewer resources than networks of over a thousand nodes have created, running on a single laptop. A Mininet container is a process (or group of processes) that no longer has access to all the host system’s native network interfaces. Containers are then assigned virtual Ethernet interfaces, which are connected to other containers through a virtual switch<sup>4</sup>. Mininet connects a host and a switch using a virtual Ethernet (veth) link. The veth link is analogous to a wire connecting two virtual interfaces, as illustrated below.

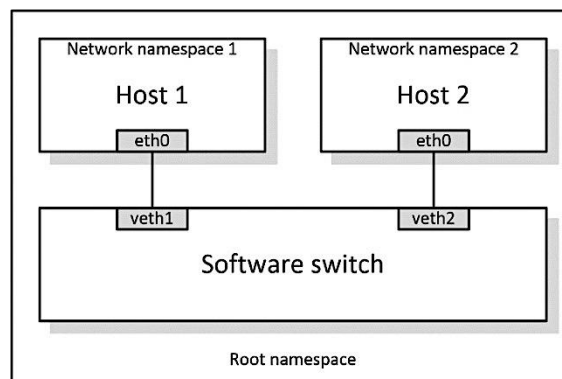


Figure 2. Network namespaces and virtual Ethernet links.

Each container is an independent network namespace, a lightweight virtualization feature that provides individual processes with separate network interfaces, routing tables, and Address Resolution Protocol (ARP) tables.

Mininet provides network emulation opposed to simulation, allowing all network software at any layer to be simply run *as is*; i.e. nodes run the native network software of the physical machine. On the other hand, in a simulated environment applications and protocol implementations need to be ported to run within the simulator before they can be used.

## 2 Invoke Mininet using the CLI

The first step to start Mininet using the CLI is to start a Linux terminal.

### 2.1 Invoke Mininet using the default topology

**Step 1.** Launch a Linux terminal by holding the `Ctrl+Alt+T` keys or by clicking on the Linux terminal icon.

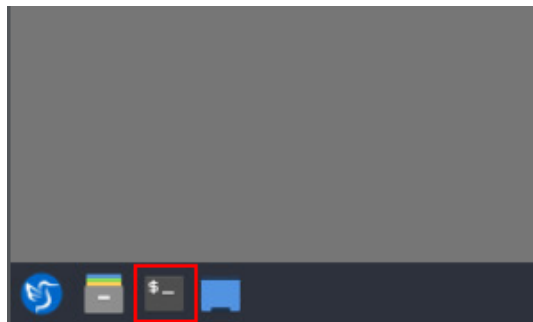


Figure 3. Linux terminal icon.

The Linux terminal is a program that opens a window and permits you to interact with a command-line interface (CLI). A CLI is a program that takes commands from the keyboard and sends them to the operating system for execution.

**Step 2.** To start a minimal topology, enter the command shown below. When prompted for a password, type `password` and hit enter. Note that the password will not be visible as you type it.

```
sudo mn
```

```

frr@frr-pc: ~
File Actions Edit View Help
frr@frr-pc: ~
frr@frr-pc:~$ sudo mn
[sudo] password for frr:
*** No default OpenFlow controller found for default switch!
*** Falling back to OVS Bridge
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
    
```

Figure 4. Starting Mininet using the CLI.

The above command starts Mininet with a minimal topology, which consists of a switch connected to two hosts as shown below.

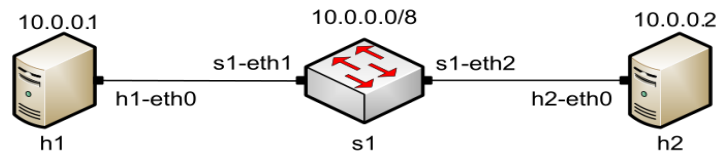


Figure 5. Mininet's default minimal topology.

When issuing the `sudo mn` command, Mininet initializes the topology and launches its command line interface which looks like this:

```
mininet>
```

**Step 3.** To display the list of Mininet CLI commands and examples on their usage, type the following command:

```
help
```



```
frr@frr-pc: ~
File Actions Edit View Help
frr@frr-pc: ~
mininet> help

Documented commands (type help <topic>):
=====
EOF      gterm  iperfudp  nodes      pingpair    py      switch
dpctl    help   link      noecho     pingpairfull  quit    time
dump     intfs  links     pingall    ports       sh      x
exit     iperf  net       pingallfull  px          source  xterm

You may also send a command to a node using:
  <node> command {args}
For example:
  mininet> h1 ifconfig

The interpreter automatically substitutes IP addresses
for node names when a node is the first arg, so commands
like
  mininet> h2 ping h3
should work.

Some character-oriented interactive commands require
noecho:
  mininet> noecho h2 vi foo.py
However, starting up an xterm/gterm is generally better:
  mininet> xterm h2

mininet> 
```

Figure 6. Mininet's `help` command.

**Step 4.** To display the available nodes, type the following command:

```
nodes
```

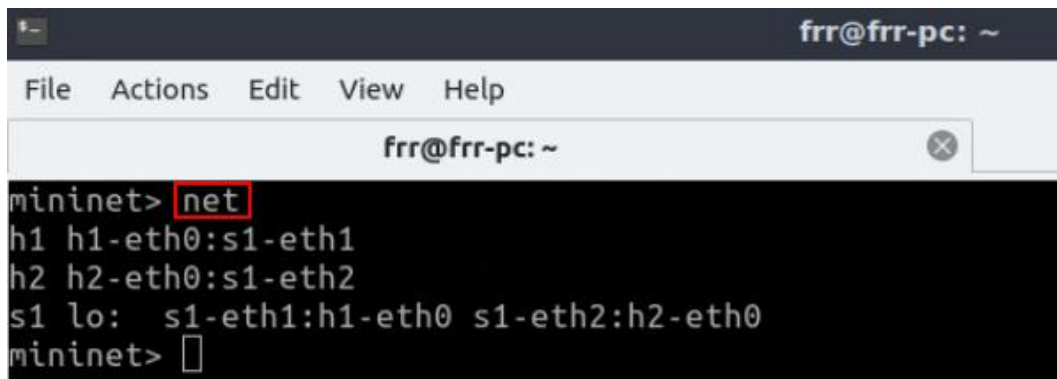
```
frr@frr-pc: ~
File Actions Edit View Help
frr@frr-pc: ~
mininet> nodes
available nodes are:
h1 h2 s1
mininet> 
```

Figure 7. Mininet's `nodes` command.

The output of this command shows that there are two hosts (host h1 and host h2) and a switch (s1).

**Step 5.** It is useful sometimes to display the links between the devices in Mininet to understand the topology. Issue the command shown below to see the available links.

```
net
```



```
frr@frr-pc: ~
File Actions Edit View Help
frr@frr-pc: ~
mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
s1 lo: s1-eth1:h1-eth0 s1-eth2:h2-eth0
mininet> 
```

Figure 8. Mininet's `net` command.

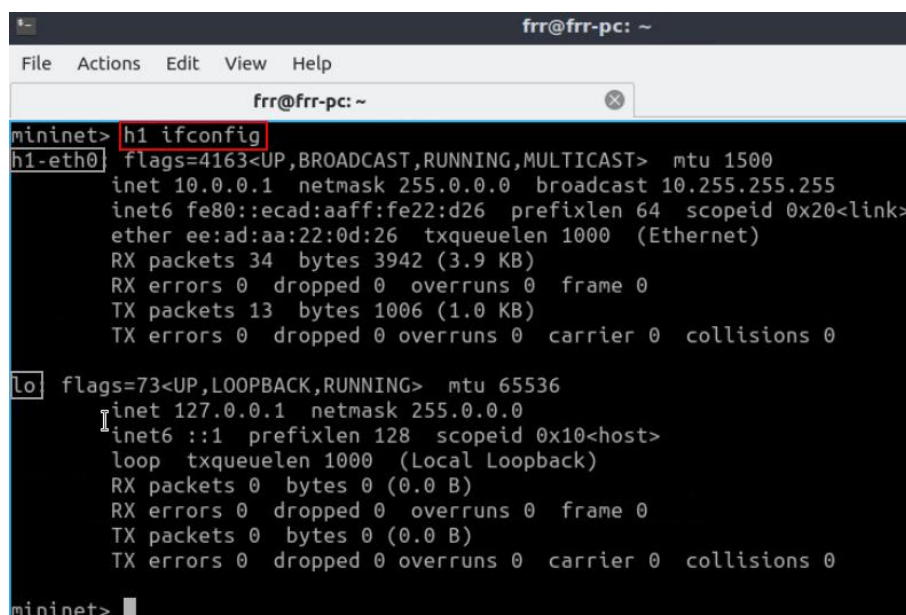
The output of this command shows that:

1. Host `h1` is connected using its network interface `h1-eth0` to the switch on interface `s1-eth1`.
2. Host `h2` is connected using its network interface `h2-eth0` to the switch on interface `s1-eth2`.
3. Switch `s1`:
  - a. has a loopback interface `lo`.
  - b. connects to `h1-eth0` through interface `s1-eth1`.
  - c. connects to `h2-eth0` through interface `s1-eth2`.

Mininet allows you to execute commands on a specific device. To issue a command for a specific node, you must specify the device first, followed by the command.

**Step 6.** To proceed, issue the command:

```
h1 ifconfig
```



```
frr@frr-pc: ~
File Actions Edit View Help
frr@frr-pc: ~
mininet> h1 ifconfig
h1-eth0 flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 10.0.0.1 netmask 255.0.0.0 broadcast 10.255.255.255
inet6 fe80::ecad:aaff:fe22:d26 prefixlen 64 scopeid 0x20<link>
ether ee:ad:aa:22:0d:26 txqueuelen 1000 (Ethernet)
RX packets 34 bytes 3942 (3.9 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 13 bytes 1006 (1.0 KB)
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
mininet> 
```

Figure 9. Output of `h1 ifconfig` command.

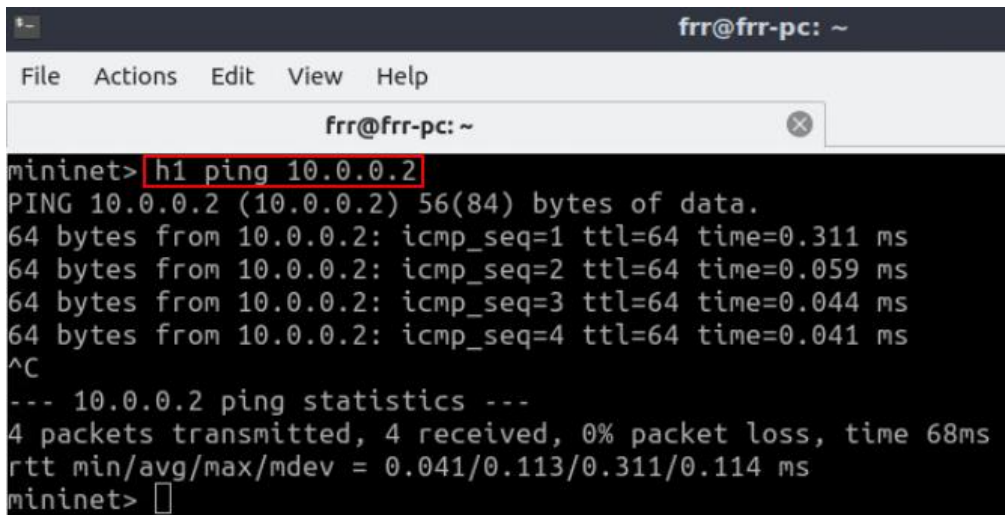
This command executes the `ifconfig` Linux command on host h1. The command shows host h1's interfaces. The display indicates that host h1 has an interface `h1-eth0` configured with IP address 10.0.0.1, and another interface `lo` configured with IP address 127.0.0.1 (loopback interface).

## 2.2 Test connectivity

Mininet's default topology assigns the IP addresses 10.0.0.1/8 and 10.0.0.2/8 to host h1 and host h2 respectively. To test connectivity between them, you can use the command `ping`. The `ping` command operates by sending Internet Control Message Protocol (ICMP) Echo Request messages to the remote computer and waiting for a response or reply. Information available includes how many responses are returned and how long it takes for them to return.

**Step 1.** On the CLI, type the command shown below. This command tests the connectivity between host h1 and host h2. To stop the test, press `Ctrl+c`. The figure below shows a successful connectivity test. Host h1 (10.0.0.1) sent four packets to host h2 (10.0.0.2) and successfully received the expected responses.

```
h1 ping 10.0.0.2
```



```
frr@frr-pc: ~  
File Actions Edit View Help  
frr@frr-pc: ~  
mininet> h1 ping 10.0.0.2  
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.  
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.311 ms  
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.059 ms  
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.044 ms  
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.041 ms  
^C  
--- 10.0.0.2 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 68ms  
rtt min/avg/max/mdev = 0.041/0.113/0.311/0.114 ms  
mininet> 
```

Figure 10. Connectivity test between host h1 and host h2.

**Step 2.** Stop the emulation by typing the following command:

```
exit
```

```

frr@frr-pc: ~
File Actions Edit View Help
frr@frr-pc: ~
mininet> exit
*** Stopping 0 controllers
*** Stopping 2 links
. .
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 771.153 seconds
frr@frr-pc:~$
    
```

Figure 11. Stopping the emulation using `exit`.

The command `sudo mn -c` is often used on the Linux terminal (not on the Mininet CLI) to clean a previous instance of Mininet (e.g., after a crash).

### 3 Build and emulate a network in Mininet using the GUI

In this section, you will use the application MiniEdit<sup>5</sup> to deploy the topology illustrated below. MiniEdit is a simple GUI network editor for Mininet.

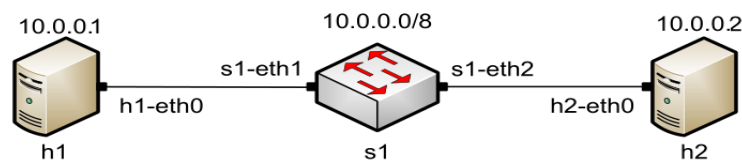


Figure 12. Lab topology.

#### 3.1 Build the network topology

**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 13. MiniEdit Desktop shortcut.

MiniEdit will start, as illustrated below.

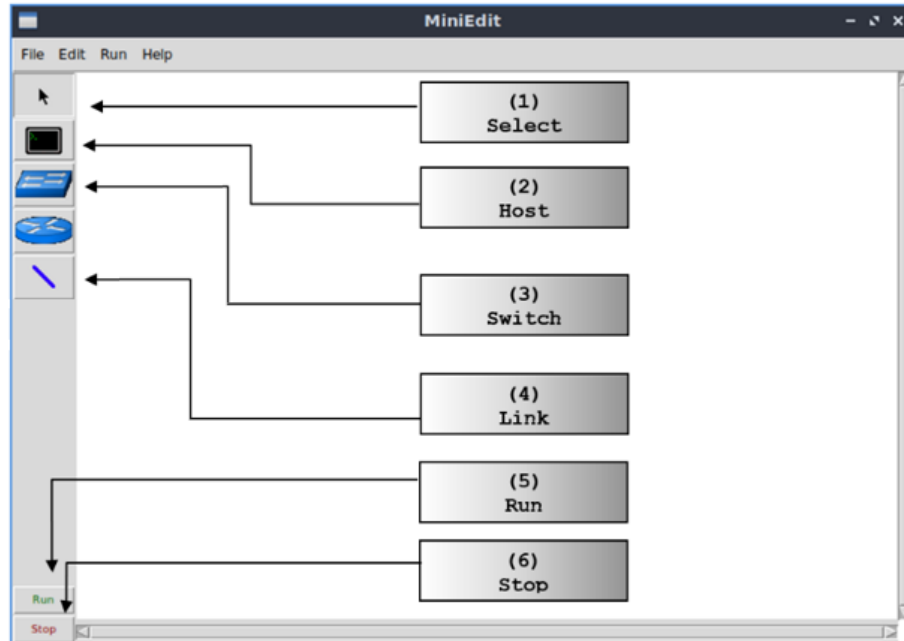


Figure 14. MiniEdit Graphical User Interface (GUI).

The main buttons are:

1. *Select*: allows selection/movement of the devices. Pressing *Del* on the keyboard after selecting the device removes it from the topology.
2. *Host*: allows addition of a new host to the topology. After clicking this button, click anywhere in the blank canvas to insert a new host.
3. *Switch*: allows addition of a new switch to the topology. After clicking this button, click anywhere in the blank canvas to insert the switch.
4. *Link*: connects devices in the topology (mainly switches and hosts). After clicking this button, click on a device and drag to the second device to which the link is to be established.
5. *Run*: starts the emulation. After designing and configuring the topology, click the run button.
6. *Stop*: stops the emulation.

**Step 2.** To build the topology illustrated in Figure 12, two hosts and one switch must be deployed. Deploy these devices in MiniEdit, as shown below.

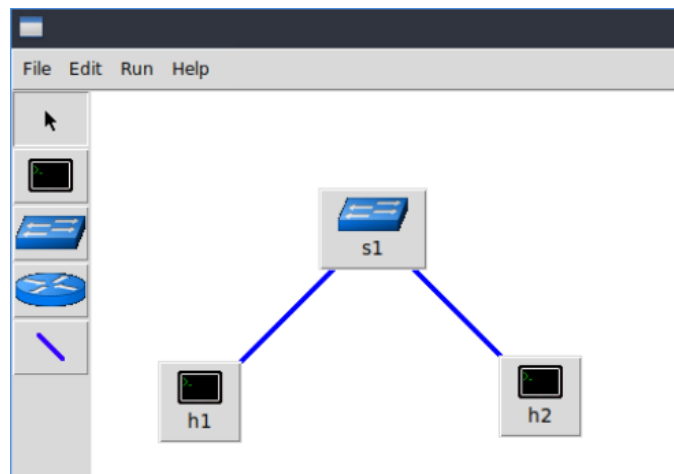


Figure 15. MiniEdit's topology.

Use the buttons described in the previous step to add and connect devices. The configuration of IP addresses is described in Step 3.

**Step 3.** Configure the IP addresses of host h1 and host h2. Host h1's IP address is 10.0.0.1/8 and host h2's IP address is 10.0.0.2/8. A host can be configured by holding the right click and selecting properties on the device. For example, host h2 is assigned the IP address 10.0.0.2/8 in the figure below. Click **OK** for the settings to be applied.

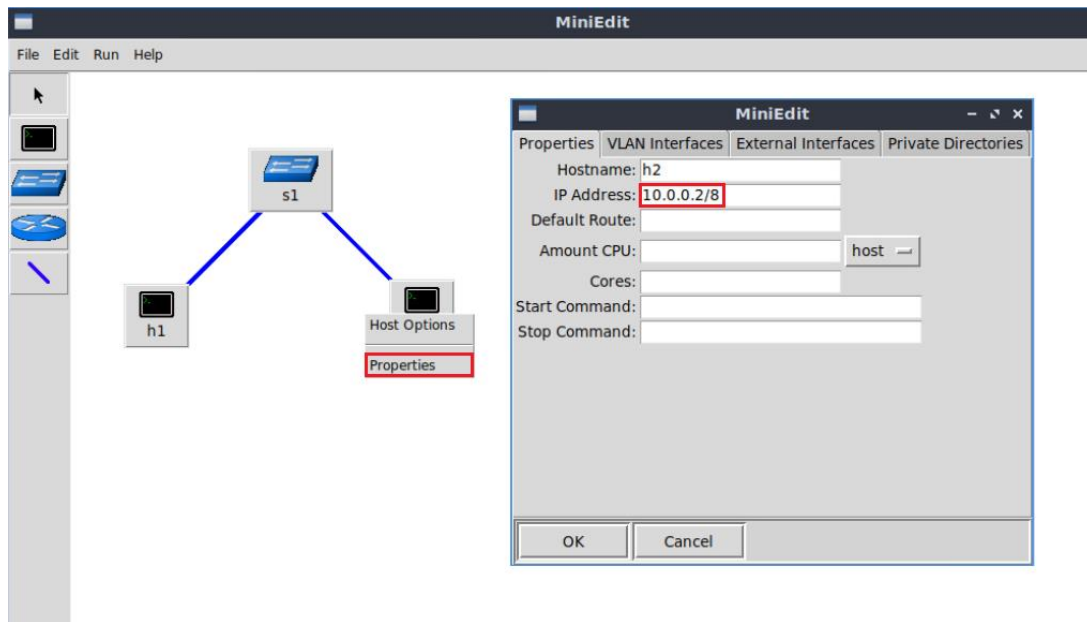


Figure 16. Configuration of a host's properties.

### 3.2 Test connectivity

Before testing the connection between host h1 and host h2, the emulation must be started.

**Step 1.** Click the *Run* button to start the emulation. The emulation will start and the buttons of the MiniEdit panel will gray out, indicating that they are currently disabled.

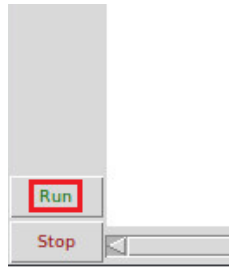


Figure 17. Starting the emulation.

**Step 2.** Open a terminal on host h1 by holding the right click on host h1 and selecting *Terminal*. This opens a terminal on host h1 and allows the execution of commands on the host h1. Repeat the procedure on host h2.

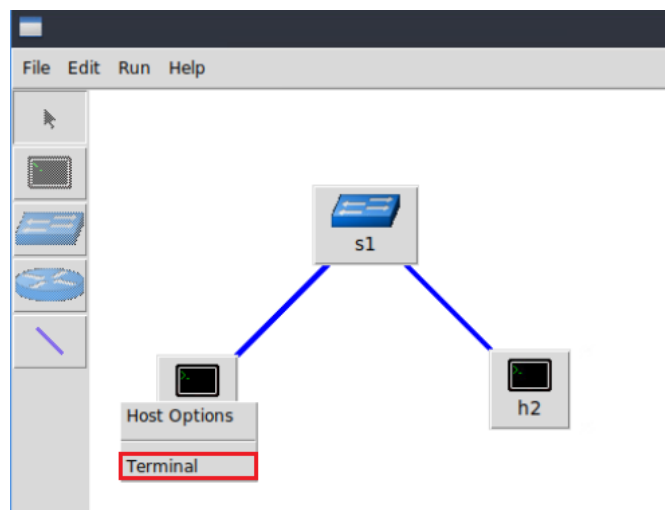


Figure 18. Opening a terminal on host h1.

The network and terminals at host h1 and host h2 will be available for testing.

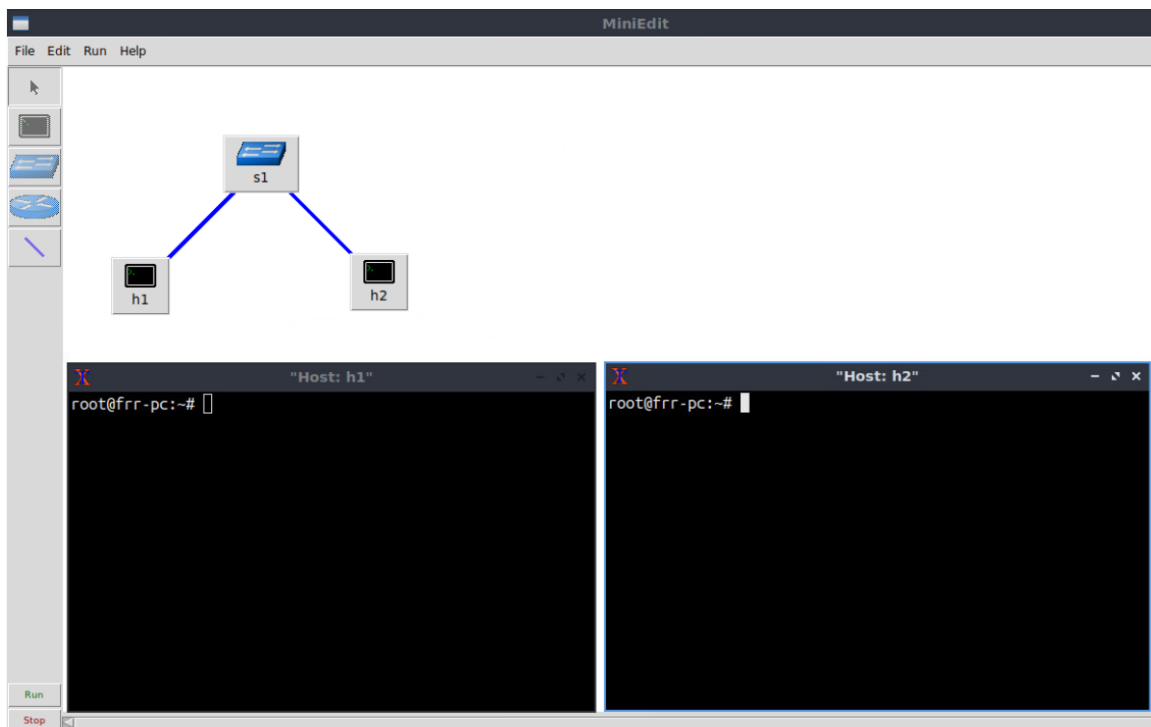


Figure 19. Terminals at host h1 and host h2.

**Step 3.** On host h1's terminal, type the command shown below to display its assigned IP addresses. The interface *h1-eth0* at host h1 should be configured with the IP address 10.0.0.1 and subnet mask 255.0.0.0.

```
ifconfig
```

```

root@admin-pc:~# ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.1 netmask 255.0.0.0 broadcast 10.255.255.255
    inet6 fe80::f0d6:67ff:fe01:6041 prefixlen 64 scopeid 0x20<link>
    ether f2:d6:67:01:60:41 txqueuelen 1000 (Ethernet)
    RX packets 51 bytes 5112 (5.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 21 bytes 1678 (1.6 KB)
    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-pc:~#

```

Figure 20. Output of `ifconfig` command on host h1.

Repeat Step 3 on host h2. Its interface *h2-eth0* should be configured with IP address 10.0.0.2 and subnet mask 255.0.0.0.

**Step 4.** On host h1's terminal, type the command shown below. This command tests the connectivity between host h1 and host h2. To stop the test, press `Ctrl+c`. The figure



below shows a successful connectivity test. Host h1 (10.0.0.1) sent six packets to host h2 (10.0.0.2) and successfully received the expected responses.

```
ping 10.0.0.2
```

```

Host: h1
root@admin-pc:~# ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.541 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.048 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.033 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.052 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.035 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.042 ms
^C
--- 10.0.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 110ms
rtt min/avg/max/mdev = 0.033/0.125/0.541/0.186 ms
root@admin-pc:~#
    
```

Figure 21. Connectivity test using `ping` command.

**Step 5.** Stop the emulation by clicking on the *Stop* button.



Figure 22. Stopping the emulation.

### 3.3 Automatic assignment of IP addresses

In the previous section, you manually assigned IP addresses to host h1 and host h2. An alternative is to rely on Mininet for an automatic assignment of IP addresses (by default, Mininet uses automatic assignment), which is described in this section.

**Step 1.** Remove the manually assigned IP address from host h1. Hold right-click on host h1, *Properties*. Delete the IP address, leaving it unassigned, and press the *OK* button as shown below. Repeat the procedure on host h2.

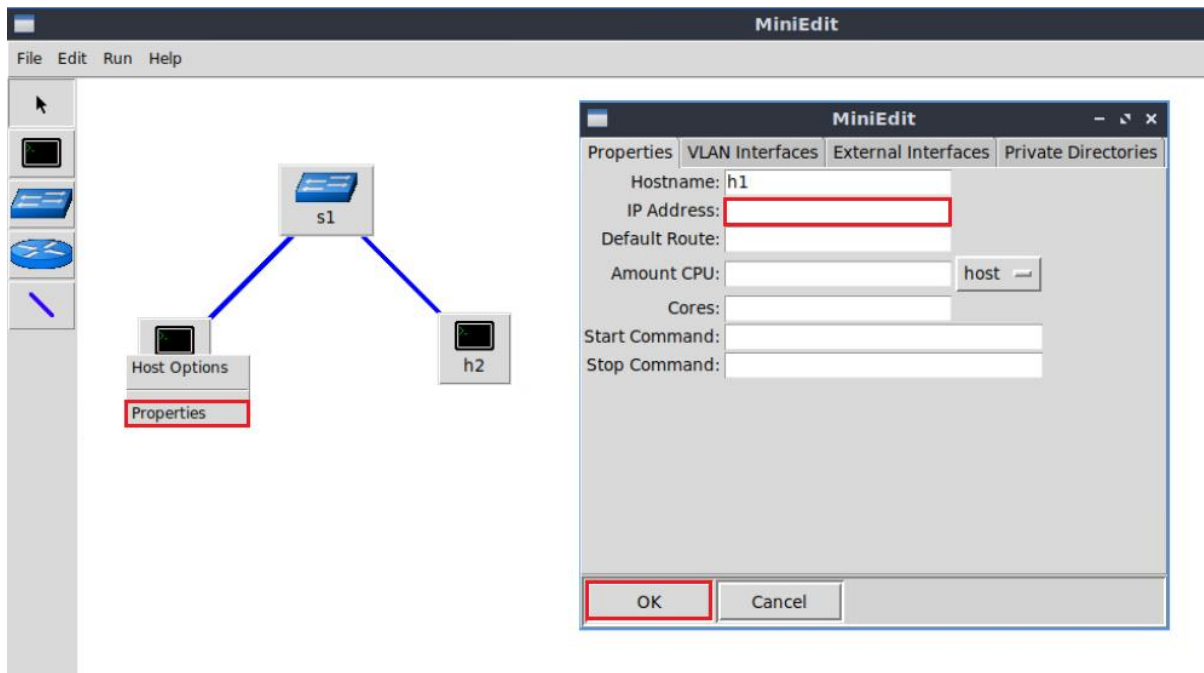


Figure 23. Host h1 properties.

**Step 2.** Click on *Edit, Preferences* button. The default IP base is 10.0.0.0/8. Modify this value to 15.0.0.0/8, and then press the *OK* button.

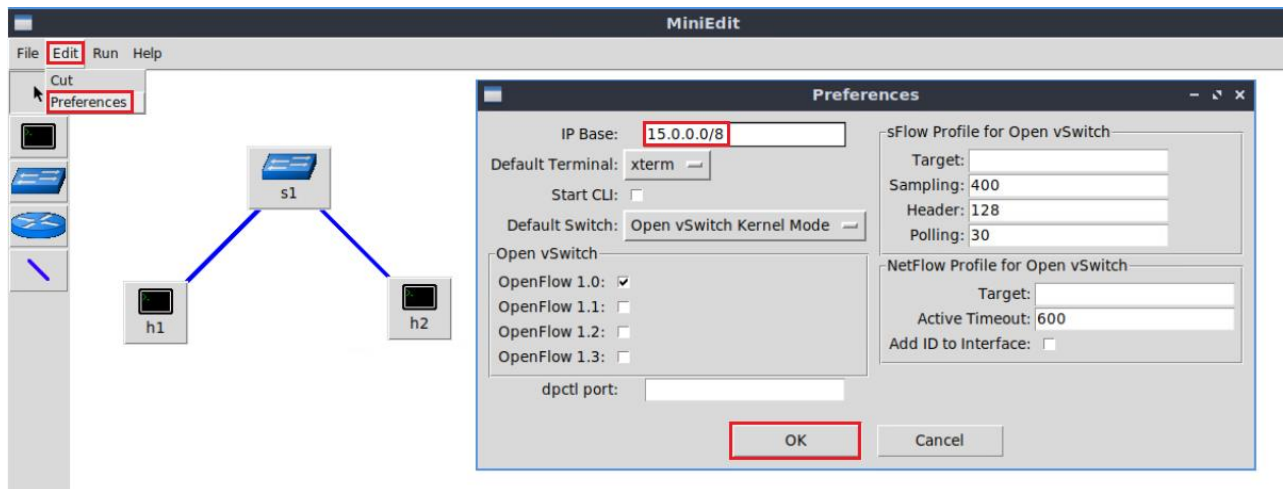


Figure 24. Modification of the IP Base (network address and prefix length).

**Step 3.** Run the emulation again by clicking on the *Run* button. The emulation will start and the buttons of the MiniEdit panel will be disabled.

**Step 4.** Open a terminal on host h1 by holding the right click on host h1 and selecting Terminal.

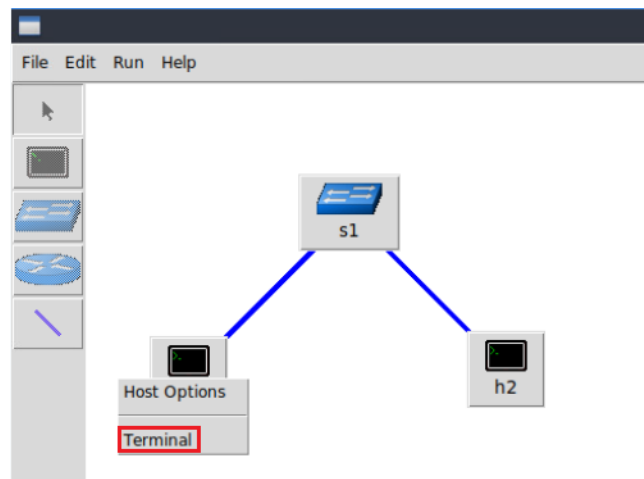


Figure 25. Opening a terminal on host h1.

**Step 5.** Type the command shown below to display the IP addresses assigned to host h1. The interface *h1-eth0* at host h1 now has the IP address 15.0.0.1 and subnet mask 255.0.0.0.

```
ifconfig
```

```

root@admin-pc:~# ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
  inet 15.0.0.1 netmask 255.0.0.0 broadcast 15.255.255.255
  inet6 fe80::5c52:56ff:febc:848b prefixlen 64 scopeid 0x20<link>
  ether 5e:52:56:bc:84:8b txqueuelen 1000 (Ethernet)
  RX packets 24 bytes 2851 (2.8 KB)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 7 bytes 586 (586.0 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-pc:~#

```

Figure 26. Output of `ifconfig` command on host h1.

You can also verify the IP address assigned to host h2 by repeating Steps 4 and 5 on host h2's terminal. The corresponding interface *h2-eth0* at host h2 has now the IP address 15.0.0.2 and subnet mask 255.0.0.0.

**Step 6.** Stop the emulation by clicking on *Stop* button.



Figure 27. Stopping the emulation.

### 3.4 Save and load a Mininet topology

In this section you will save and load a Mininet topology. It is often useful to save the network topology, particularly when its complexity increases. MiniEdit enables you to save the topology to a file.

**Step 1.** Save the current topology by clicking on *File* then *Save*. Provide a name for the topology and save it in the local folder. In this case, we used *myTopology* as the topology name.

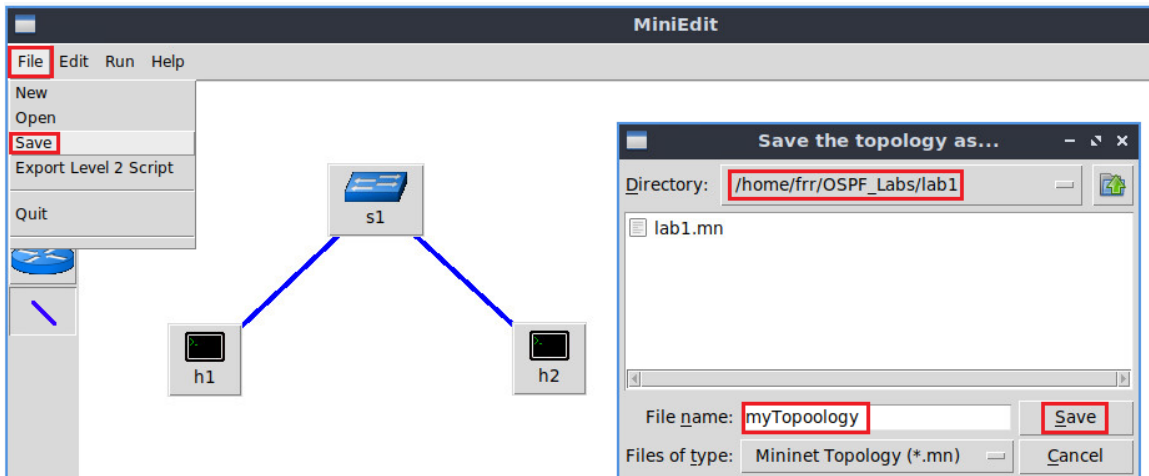


Figure 28. Saving the topology.

**Step 2.** Load the topology by clicking on *File* then *Open*. Search for the topology file called *lab1.mn* and click on *Open*. A new topology will be loaded to MiniEdit.

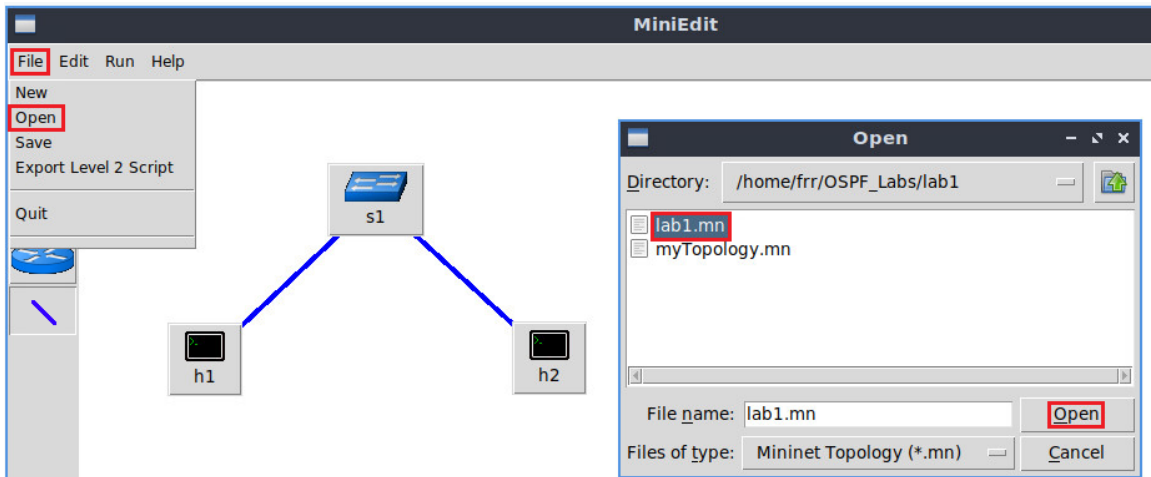


Figure 29. Opening a topology.

#### 4 Configure router r1

In the previous step, you loaded a topology that consists in two networks directly connected to router r1. Look at Figure 30. In this topology two LANs, defined by switch s1 and switch s2 are connected to router r1. Initially, host h1 and host h2 do not have connectivity thus, you will configure router r1's interfaces in order to establish connectivity between the two networks.

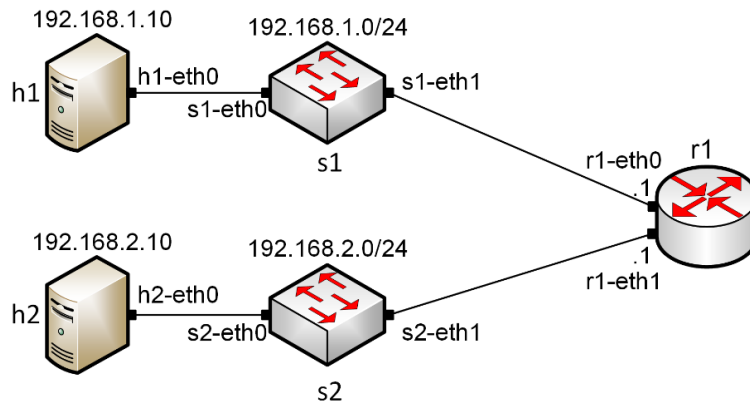


Figure 30. Topology.

Table 2 summarized the IP addresses used to configure router r1 and the end-hosts.

Table 2. Topology information.

Device	Interface	IP Address	Subnet	Default gateway
r1	r1-eth0	192.168.1.1	/24	N/A
	r1-eth1	192.168.2.1	/24	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

**Step 1.** Click on the *Run* button to start the emulation. The emulation will start and the buttons of the MiniEdit panel will gray out, indicating that they are currently disabled.

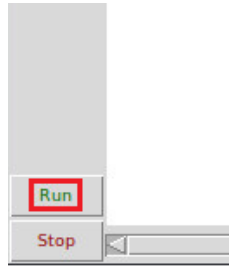


Figure 31. Starting the emulation.

#### 4.1 Verify end-hosts configuration

In this section, you will verify that the IP addresses are assigned according to Table 2. Additionally, you will check routing information.

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

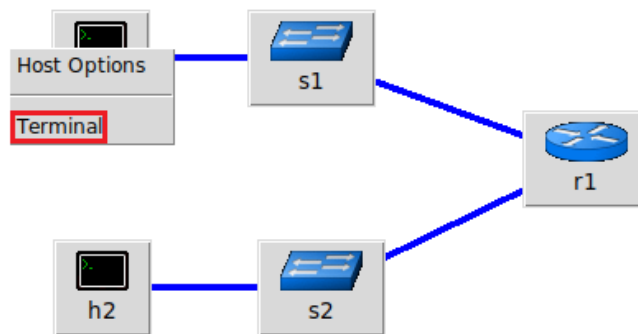


Figure 32. Opening a terminal on host h1.

**Step 2.** In host h1 terminal, type the command shown below to verify that the IP address was assigned successfully. You will verify that host h1 has two interfaces, *h1-eth0* configured with the IP address 192.168.1.10 and the subnet mask 255.255.255.0 and, the loopback interface *lo* configured with the IP address 127.0.0.1.

```
ifconfig
```

```

root@frr-pc:~# ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.10 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::7c11:30ff:fea5:d022 prefixlen 64 scopeid 0x20<link>
    ether 7e:11:30:a5:d0:22 txqueuelen 1000 (Ethernet)
    RX packets 32 bytes 3781 (3.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 12 bytes 936 (936.0 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@frr-pc:~#

```

Figure 33. Output of `ifconfig` command.

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

```
route
```

```

root@frr-pc:~# ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.10 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::7c11:30ff:fea5:d022 prefixlen 64 scopeid 0x20<link>
    ether 7e:11:30:a5:d0:22 txqueuelen 1000 (Ethernet)
    RX packets 32 bytes 3781 (3.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 12 bytes 936 (936.0 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@frr-pc:~# 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@frr-pc:~#

```

Figure 34. Output of `route` command.

**Step 4.** In order to verify host 2 default route, proceed similarly by repeating from step 1 to step 3 in host h2 terminal. Similar results should be observed.

## 4.2 Configure router's interface

**Step 1.** In order to configure router r1, hold right-click on router r1 and select *Terminal*.

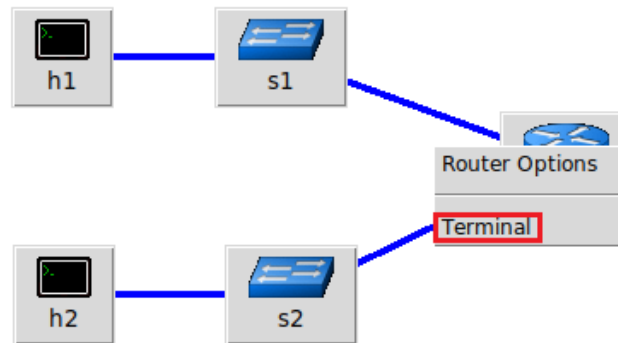


Figure 35. Opening a terminal on router r1.

**Step 2.** In this step, you will start the zebra daemon, a multi-server routing software that provides TCP/IP based routing protocols. The configuration will not be working if you do not enable zebra daemon initially. To activate the zebra daemon, type the following command:

```
zebra
```

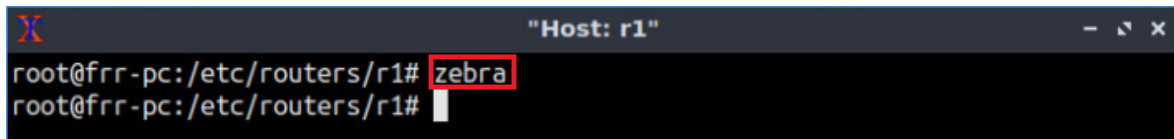


Figure 36. Starting zebra daemon.

**Step 3.** 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:

```
vtys
```

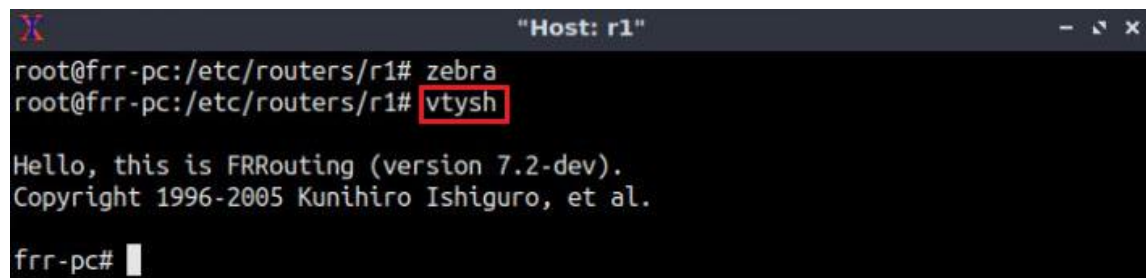
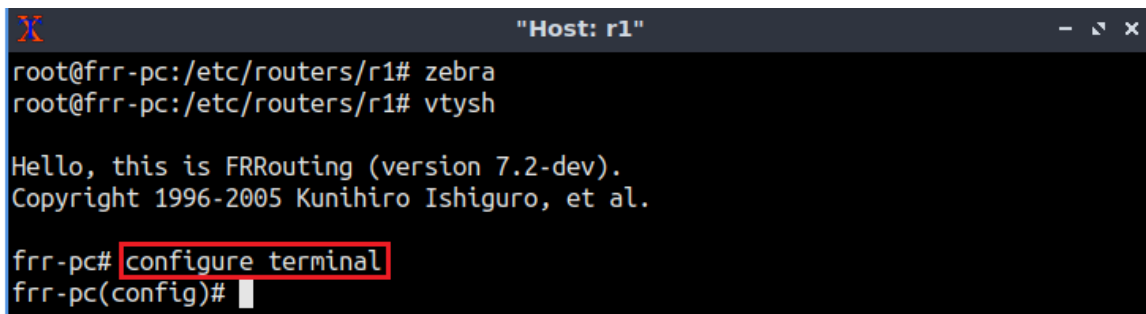


Figure 37. Starting vtysh on router r1.

**Step 4.** Type the following command in the router r1 terminal to enter in configuration mode.

```
configure terminal
```





```

Host: r1
root@frr-pc:/etc/routers/r1# zebra
root@frr-pc:/etc/routers/r1# vtysh

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

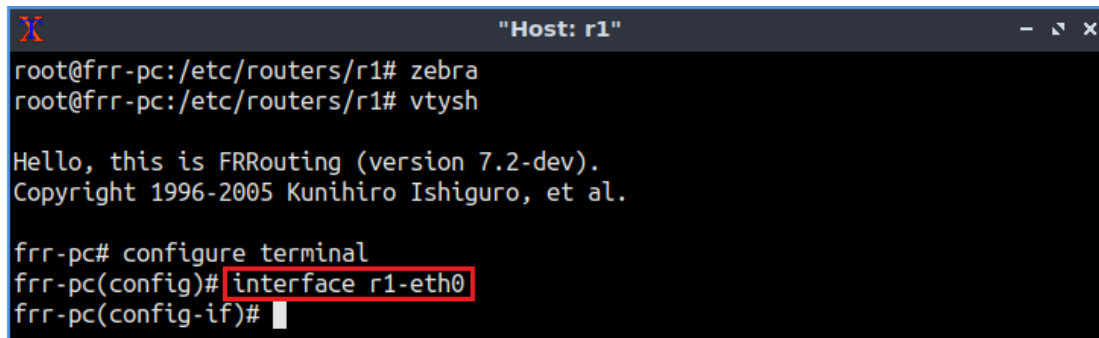
frr-pc# configure terminal
frr-pc(config)#

```

Figure 38. Entering in configuration mode.

**Step 5.** Type the following command in the router *r1* terminal to configure interface *r1-eth0*.

```
interface r1-eth0
```



```

Host: r1
root@frr-pc:/etc/routers/r1# zebra
root@frr-pc:/etc/routers/r1# vtysh

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

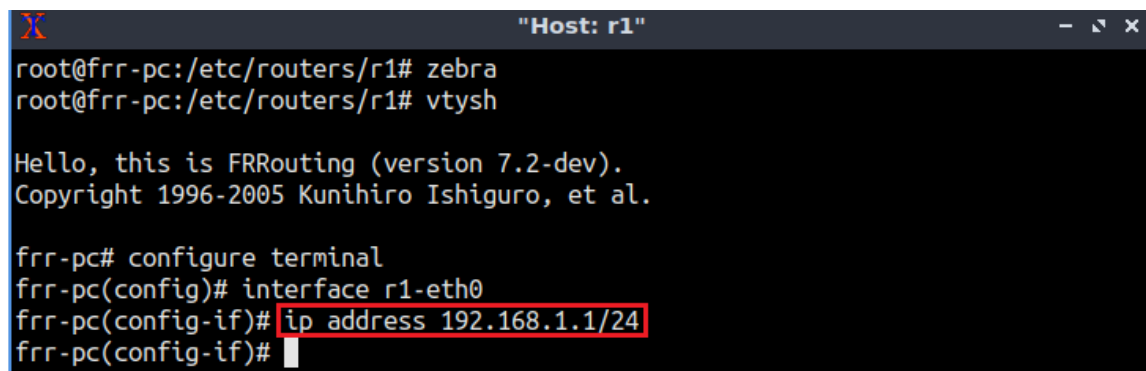
frr-pc# configure terminal
frr-pc(config)# interface r1-eth0
frr-pc(config-if)#

```

Figure 39. Configuring interface *r1-eth0*.

**Step 6.** Type the following command on router *r1* terminal to configure the IP address of the interface *r1-eth0*.

```
ip address 192.168.1.1/24
```



```

Host: r1
root@frr-pc:/etc/routers/r1# zebra
root@frr-pc:/etc/routers/r1# vtysh

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

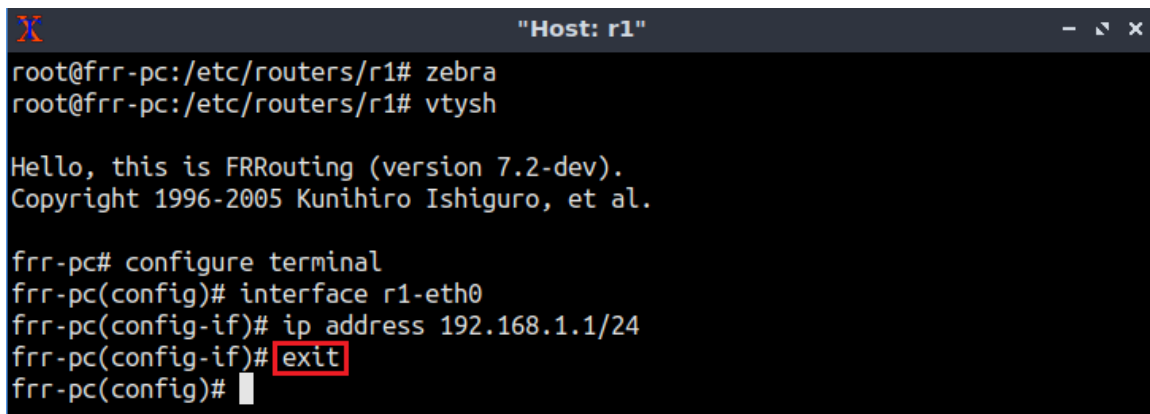
frr-pc# configure terminal
frr-pc(config)# interface r1-eth0
frr-pc(config-if)# ip address 192.168.1.1/24
frr-pc(config-if)#

```

Figure 40. Configuring an IP address to interface *r1-eth0*.

**Step 7.** Type the following command exit from interface *r1-eth0* configuration.

```
exit
```



```
root@frr-pc:/etc/routers/r1# zebra
root@frr-pc:/etc/routers/r1# vtysh

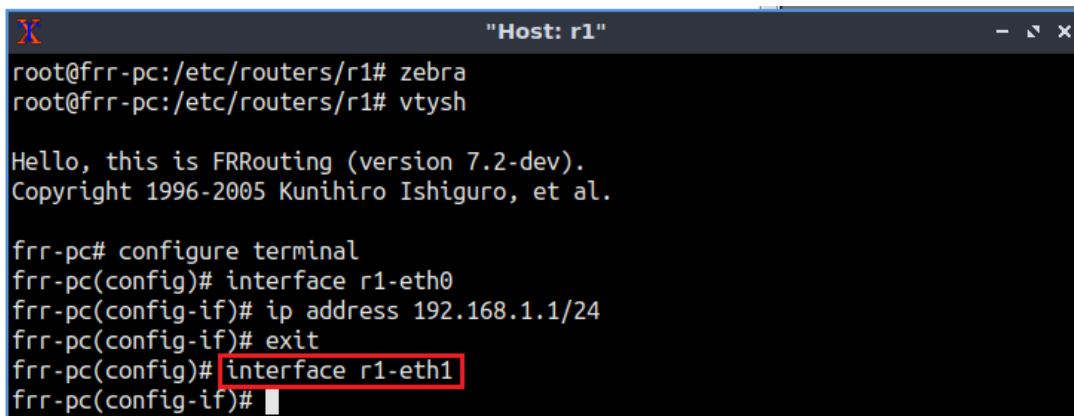
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# interface r1-eth0
frr-pc(config-if)# ip address 192.168.1.1/24
frr-pc(config-if)# exit
frr-pc(config)#
```

Figure 41. Exiting from configuring interface *r1-eth0*.

**Step 8.** Type the following command on router *r1* terminal to configure the interface *r1-eth1*.

```
interface r1-eth1
```



```
root@frr-pc:/etc/routers/r1# zebra
root@frr-pc:/etc/routers/r1# vtysh

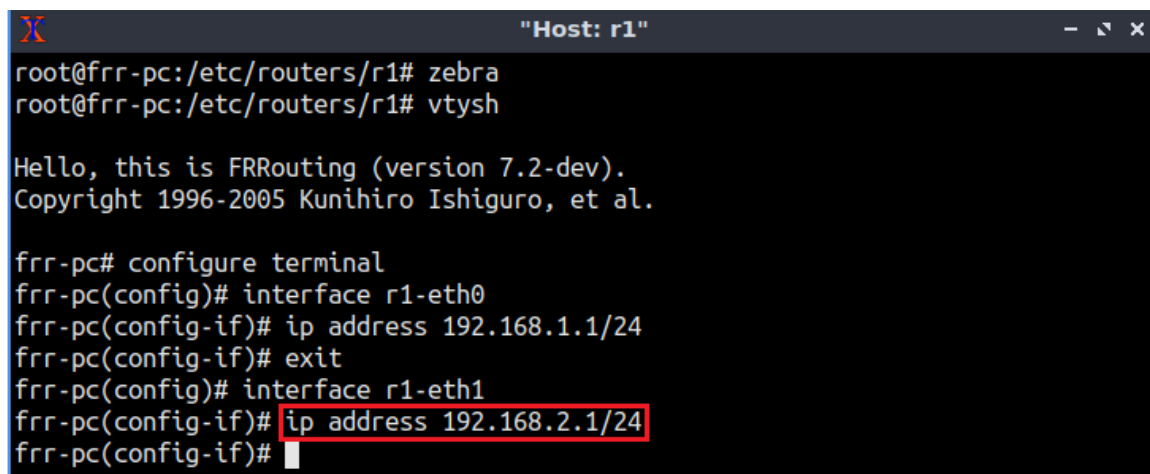
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# interface r1-eth0
frr-pc(config-if)# ip address 192.168.1.1/24
frr-pc(config-if)# exit
frr-pc(config)# interface r1-eth1
frr-pc(config-if)#
```

Figure 42. Configuring interface *r1-eth1*.

**Step 9.** Type the following command on router *r1* terminal to configure the IP address of the interface *r1-eth1*.

```
ip address 192.168.2.1/24
```



```
root@frr-pc:/etc/routers/r1# zebra
root@frr-pc:/etc/routers/r1# vtysh

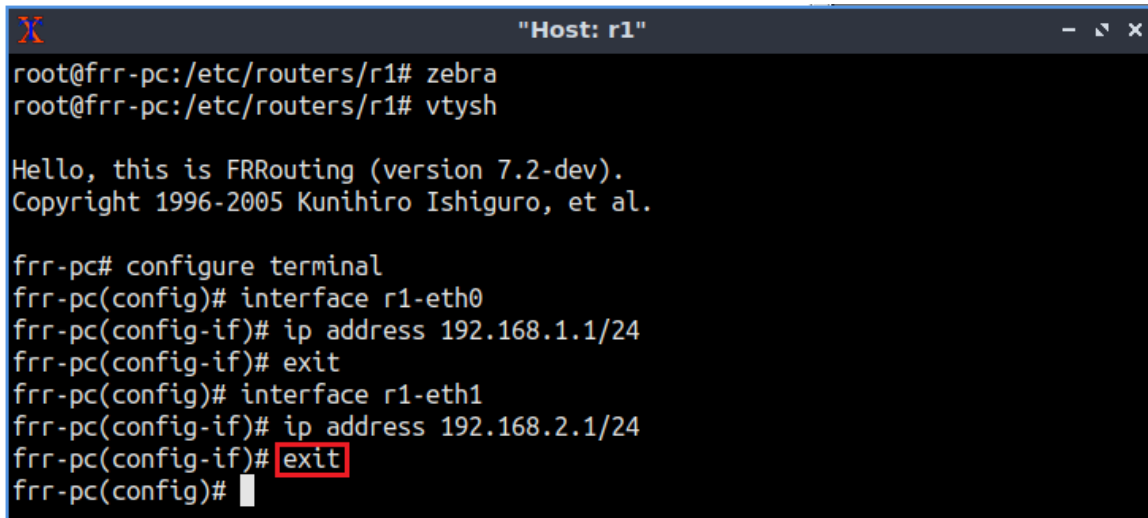
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# interface r1-eth0
frr-pc(config-if)# ip address 192.168.1.1/24
frr-pc(config-if)# exit
frr-pc(config)# interface r1-eth1
frr-pc(config-if)# ip address 192.168.2.1/24
frr-pc(config-if)#
```

Figure 43. Configuring an IP address to interface *r1-eth1*.

**Step 10.** Type the following command to exit from *r1-eth1* interface configuration.

```
exit
```



```
Host: r1
root@frr-pc:/etc/routers/r1# zebra
root@frr-pc:/etc/routers/r1# vtysh

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

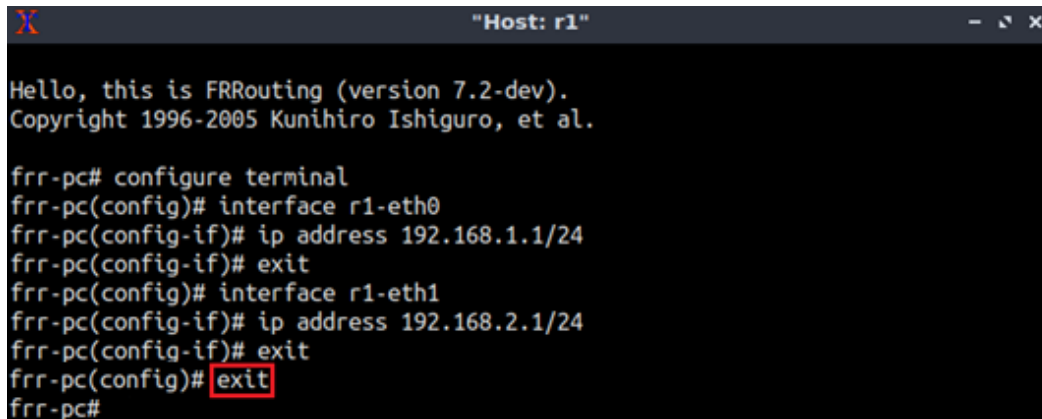
frr-pc# configure terminal
frr-pc(config)# interface r1-eth0
frr-pc(config-if)# ip address 192.168.1.1/24
frr-pc(config-if)# exit
frr-pc(config)# interface r1-eth1
frr-pc(config-if)# ip address 192.168.2.1/24
frr-pc(config-if)# exit
frr-pc(config)#
```

Figure 44. Exiting from configuring interface *r1-eth1*.

### 4.3 Verify router r1 configuration

**Step 1.** Exit from router r1 configuration mode issuing the following command:

```
exit
```



```
Host: r1

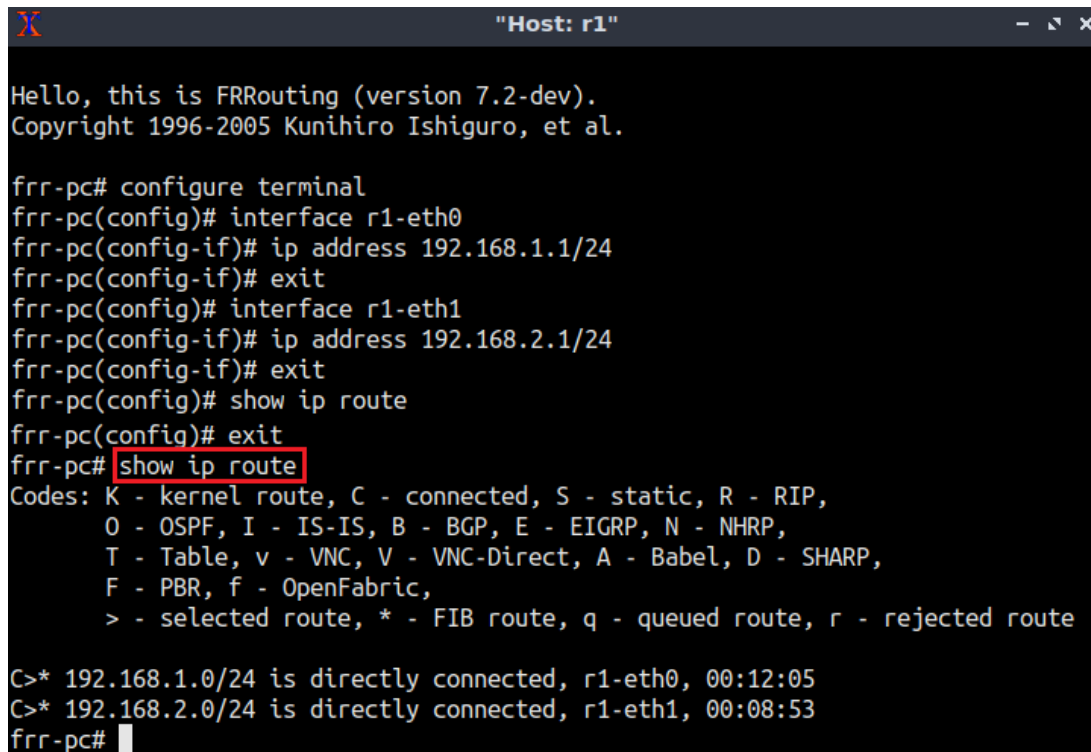
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# interface r1-eth0
frr-pc(config-if)# ip address 192.168.1.1/24
frr-pc(config-if)# exit
frr-pc(config)# interface r1-eth1
frr-pc(config-if)# ip address 192.168.2.1/24
frr-pc(config-if)# exit
frr-pc(config)# exit
frr-pc#
```

Figure 45. Exiting from configuration mode.

**Step 2.** Type the following command on router r1 terminal to verify the routing information of router r1. It will be showing all the directly connected networks.

```
show ip route
```



```

Host: r1
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# interface r1-eth0
frr-pc(config-if)# ip address 192.168.1.1/24
frr-pc(config-if)# exit
frr-pc(config)# interface r1-eth1
frr-pc(config-if)# ip address 192.168.2.1/24
frr-pc(config-if)# exit
frr-pc(config)# show ip route
frr-pc(config)# exit
frr-pc# 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, 00:12:05
C>* 192.168.2.0/24 is directly connected, r1-eth1, 00:08:53
frr-pc#

```

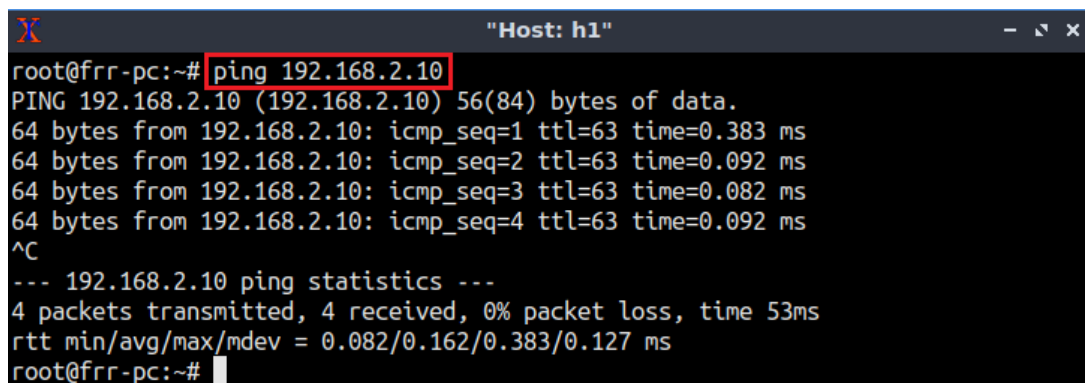
Figure 46. Displaying routing information of router r1.

#### 4.4 Test connectivity between end-hosts

In this section you will run a connectivity test between host h1 and host h2.

**Step 1.** In host h1 terminal type the command shown below. Notice that according to Table 2, the IP address 192.168.2.10 is assigned to host h2. To stop the test press `ctrl+c`

```
ping 192.168.2.10
```



```

Host: h1
root@frr-pc:~# 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=63 time=0.383 ms
64 bytes from 192.168.2.10: icmp_seq=2 ttl=63 time=0.092 ms
64 bytes from 192.168.2.10: icmp_seq=3 ttl=63 time=0.082 ms
64 bytes from 192.168.2.10: icmp_seq=4 ttl=63 time=0.092 ms
^C
--- 192.168.2.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 53ms
rtt min/avg/max/mdev = 0.082/0.162/0.383/0.127 ms
root@frr-pc:~#

```

Figure 47. Connectivity test between host h1 and host h2.

This concludes lab 1. Stop the emulation and then exit out of MiniEdit and the Linux terminal.

## References

1. Mininet walkthrough. [Online]. Available: <http://Mininet.org>.
2. Mckeown N., Anderson T., Balakrishnan H., Parulkar G., Peterson L., Rexford J., Shenker S., Turner J., "*OpenFlow*," ACM SIGCOMM Computer Communication Review, vol. 38, no. 2, p. 69, 2008.
3. Esch J., "*Prolog to, software-defined networking: a comprehensive survey*," Proceedings of the IEEE, vol. 103, no. 1, pp. 10–13, 2015.
4. Dordal P., "*An Introduction to computer networks*,". [Online]. Available: <https://intronetworks.cs.luc.edu/>.
5. Lantz B., Gee G. "*MiniEdit: a simple network editor for Mininet*." 2013. [Online]. Available: <https://github.com/Mininet/Mininet/blob/master/examples>.



UNIVERSITY OF  
**SOUTH CAROLINA**

## OPEN SHORTEST PATH FIRST

### Lab 2: Introduction to Free Range Routing (FRR)

Document Version: **05-25-2021**



Award 1829698

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

## Contents

Overview .....	3
Objectives.....	3
Lab settings .....	3
Lab roadmap .....	3
1 Introduction to FRR.....	3
1.1 FRR architecture .....	4
1.2 FRR and Mininet integration .....	5
2 Lab topology.....	6
2.1 Lab settings.....	6
2.2 Open the topology .....	7
2.3 Load the configuration file .....	8
2.4 Run the emulation.....	9
2.5 Verify the configuration .....	10
2.6 Test connectivity between end-hosts .....	14
3 Configure a routing protocol .....	14
3.1 Enable a routing daemon .....	15
3.2 Configure static route .....	16
3.3 Verify the configuration .....	18
4 Test connectivity and verify routes between end-hosts .....	19
References .....	20

## Overview

This lab is an introduction to Free Range Routing (FRR), which is a routing software suite that provides TCP/IP based routing services with routing protocols support. FRR also coordinates tasks such as exchanging routing information with other routers, making routing and policy decisions and, managing packet forwarding. In this lab, you will explore FRR architecture, load basic configuration and conduct connectivity tests within a simple topology.

## Objectives

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

1. Understand the architecture of FRR.
2. Run FRR daemons in an emulated environment.
3. Enable routing features using the router's command line.
4. Navigate into FRR terminal using administrative commands.
5. Load a configuration file into the router.
6. Perform a connectivity test between end hosts.

## Lab settings

The information in Table 1 provides the credentials of the Client.

Table 1. Credentials to access the Client machine.

Device	Account	Password
Client	admin	password

## Lab roadmap

This lab is organized as follows:

1. Section 1: Introduction to FRR.
2. Section 2: Lab topology.
3. Section 3: Configure a routing protocol.
4. Section 4: Test connectivity and verify routes between end-hosts.

### 1 Introduction to FRR

Implementing IP routing usually involves buying expensive and vertically integrated equipment from specific companies. This approach has limitations such as the hardware's



cost, closed source software, and the training required to operate and configure the devices. Networking professionals, operators, and researchers sometimes are limited by the capabilities of such routing products. Moreover, combining routing functionalities with existing open-source software packages is usually constrained by the number of separate devices deployed.

For example, operators could be interested in collecting information about the behavior of routing devices, process them, and make them available. Therefore, to achieve such capabilities, additional storage and scripting capacities are required. Such resources are not available in existing routing products. On the other hand, researchers may be interested in developing routing protocols by extending an existing one without writing a complete implementation from scratch.

FRR suite<sup>1</sup> is a package of Unix/Linux software that implements standard network routing protocols, such as Routing Information Protocol<sup>4</sup> (RIP), Open Shortest Path First<sup>5</sup> (OSPF), Border Gateway Protocol<sup>6</sup> (BGP), and Intermediate System to Intermediate System IS-IS<sup>7</sup>. The package also includes a routing information management process to act as an intermediary between the routing protocols and the active routes installed with the kernel. A library provides support for configuration and an interactive command-line interface. The routing protocols supported by FRR can be extended to enable experimentation, logging, or custom processing. Also, libraries and kernel daemons provide a framework to facilitate the development of new routing protocol daemons. A wide range of functionalities can be obtained by combining other software packages to allow integration into a single device and enable innovative solutions to networking problems.

FRR is distributed under General Public License v2.0 (GPLv2). The community of operators, vendors, non-profits, and researchers is interested in increasing the visibility of FRR and a potential path to more comprehensive testing and deployment of proposed modifications to routing protocols or new routing protocols.

## 1.1 FRR architecture

FRR takes a different approach than traditional routing software, consisting of a single process program that provides all the routing protocol functionalities. FRR is composed of a suite of daemons that work together to build a routing table. Each routing protocol is implemented in its daemon. These daemons exchange information through another daemon called zebra, responsible for coordinating routing decisions and managing the data plane.

Since all the protocols are running independently, this architecture provides high resiliency, which means that an error, crash, or exploit in one protocol daemon will generally not affect the other protocols. It is also flexible and extensible since modularity makes it easy to implement new protocols and append them to the suite<sup>1</sup>. Additionally, each daemon implements a plugin system allowing new functionality to be loaded at runtime.

Figure 1 illustrates FRR architecture. It consists of a set of processes communicating via Inter-process Communication (IPC) protocol. This protocol refers to the mechanism provided by an operating system (OS) to manage shared data between different processes. Network routing protocols such as BGP, OSPF, and IS-IS are implemented in processes such as *bgpd*, *ripd*, *ospfd*, *ldpd*, etc. These processes are daemons that implement routing protocols. For example, the *bgpd* process implements the BGP daemon, the *ripd* process implements the RIP daemon, etc. Another daemon, called zebra, acts as an intermediary between the kernel's forwarding plane and the routing protocol processes. Additionally, an interactive command-line tool called *vtys* allows these processes to be monitored and configured. The *vtys* command-line tool communicates with other processes via a simple string passing protocol, where the strings are essentially identical to the commands entered.

The *zebra* process is a fundamental part of FRR architecture. Its purpose is to maintain a backup of packet forwarding state, such as the network interfaces and the table of currently active routes. The currently active routes are also referred to as the Forwarding Information Base (FIB)<sup>2</sup>. Usually, the kernel manages packet forwarding therefore, the kernel maintains these. The *zebra* process also collects routing information from the routing protocol processes and stores these, together with its shadow copy of the FIB, in its own Routing Information Base (RIB)<sup>2</sup> whereas, static routes are also configured. The *zebra* process then is responsible for selecting the best route from all those available for a destination and updating the FIB<sup>3</sup>. Additionally, information about the current best routes may be distributed to the protocol daemons. The *zebra* process maintains the routing daemons updated if any change occurs in the network interface state.

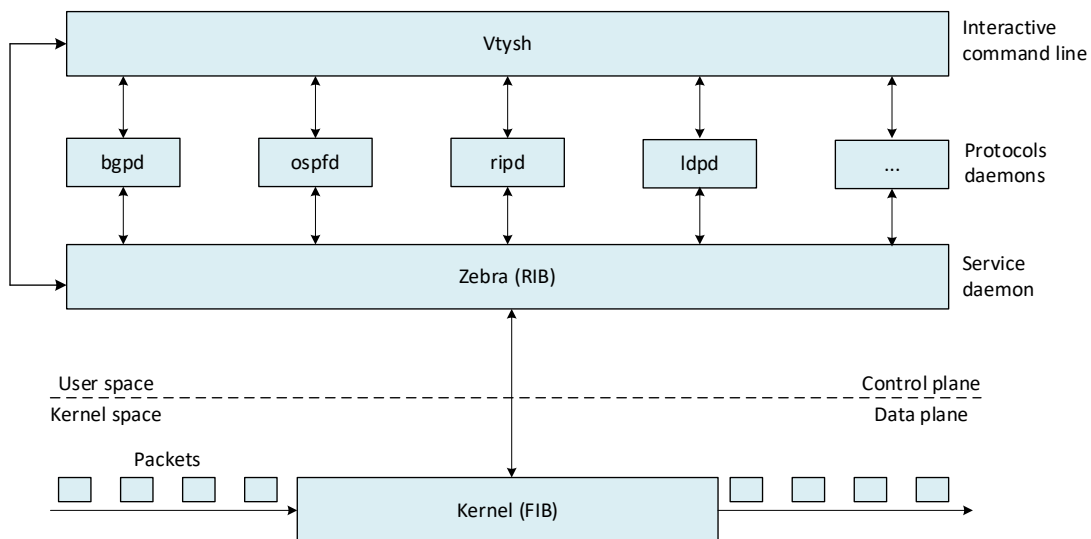


Figure 1. FRR architecture.

## 1.2 FRR and Mininet integration

Mininet is a network emulator which runs a collection of end-hosts, switches, routers, and links on a single Linux kernel<sup>5</sup>. Mininet provides network emulation, allowing all network software at any layer to be run as-is, i.e., nodes run the native network software of the physical machine. Hence, the set of commands provided by FRR are inherited and

can be executed using Mininet’s command-line interface. This feature allows the user to run and configure FRR in the emulated routers. FRR is production-ready, but we are using it in a simulated environment.

## 2 Lab topology

Consider Figure 2. The lab topology is organized in two networks, *Network 1*, and *Network 2*. Both networks have the following elements: a router to connect the networks together, a switch that defines a Local Area Network (LAN) and lastly, a host aimed to test end-to-end connectivity

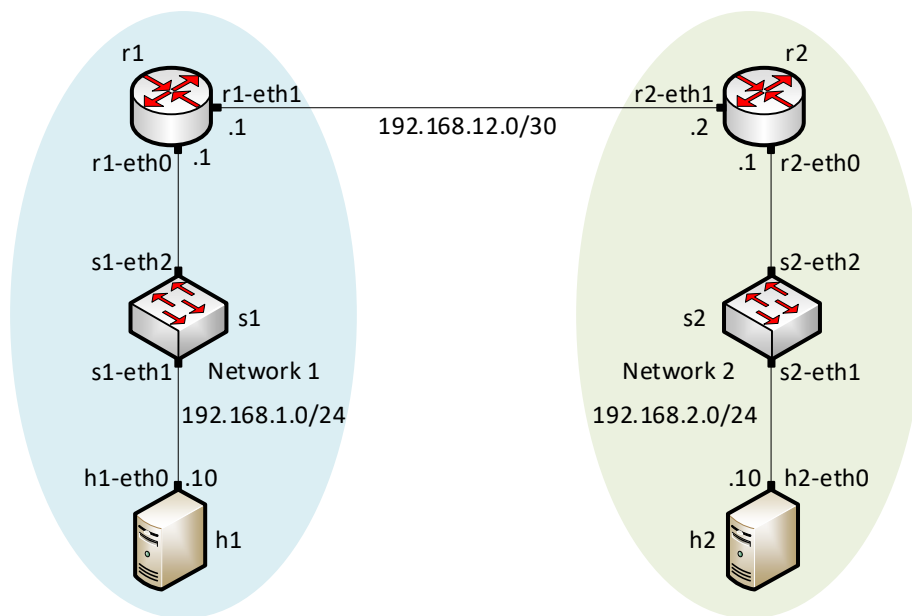


Figure 2. Lab topology.

### 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	IP Address	Subnet	Default gateway
r1	r1-eth0	192.168.1.1	/24	N/A
	r1-eth1	192.168.12.1	/30	N/A
r2	r2-eth0	192.168.2.1	/24	N/A
	r2-eth1	192.168.12.2	/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 Open the topology

In this section, you will open MiniEdit<sup>9</sup> and load the lab topology. MiniEdit provides a Graphical User Interface (GUI) that facilitates the creation and simulation of network topologies in Mininet. This tool has additional capabilities: configuring network elements (IP addresses, default gateway), saving the topology, and exporting a layer 2 model.

**Step 1.** A shortcut to MiniEdit is located on the Client machine’s desktop. Start MiniEdit by clicking on MiniEdit’s shortcut. When prompted for a password, type `password`.



Figure 3. MiniEdit shortcut.

**Step 2.** On MiniEdit’s menu bar, click on File, then open to load the lab’s topology. Open the Lab2.mn topology file stored in the default directory, `/home/frr/OSPF_Labs/lab2` and click on Open.

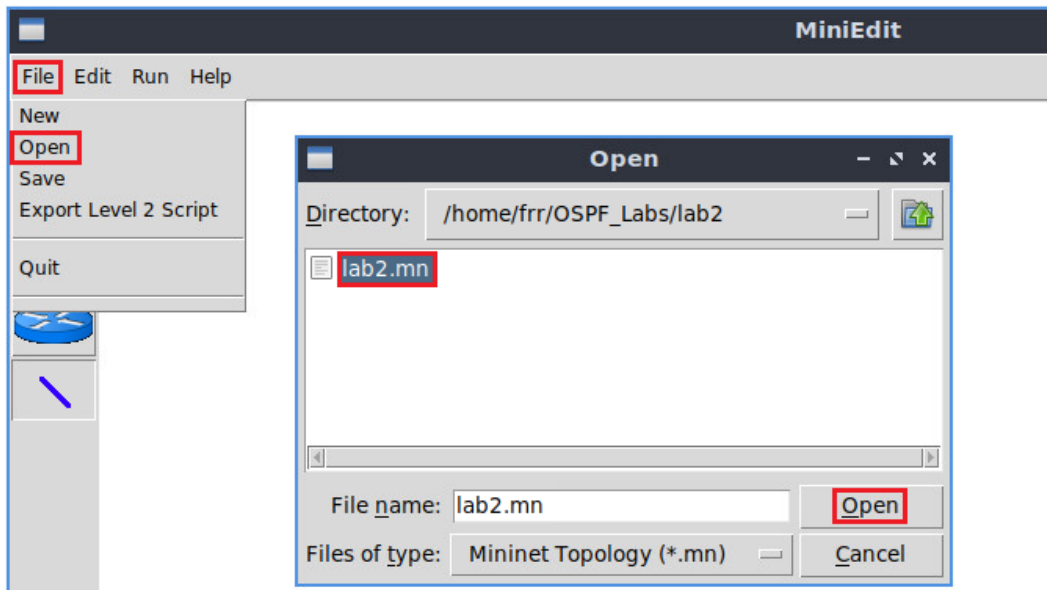


Figure 4. MiniEdit’s open dialog.

Figure 5 shows the topology used in this lab. To configure the interfaces, you will execute a script that will load the configuration on the routers.

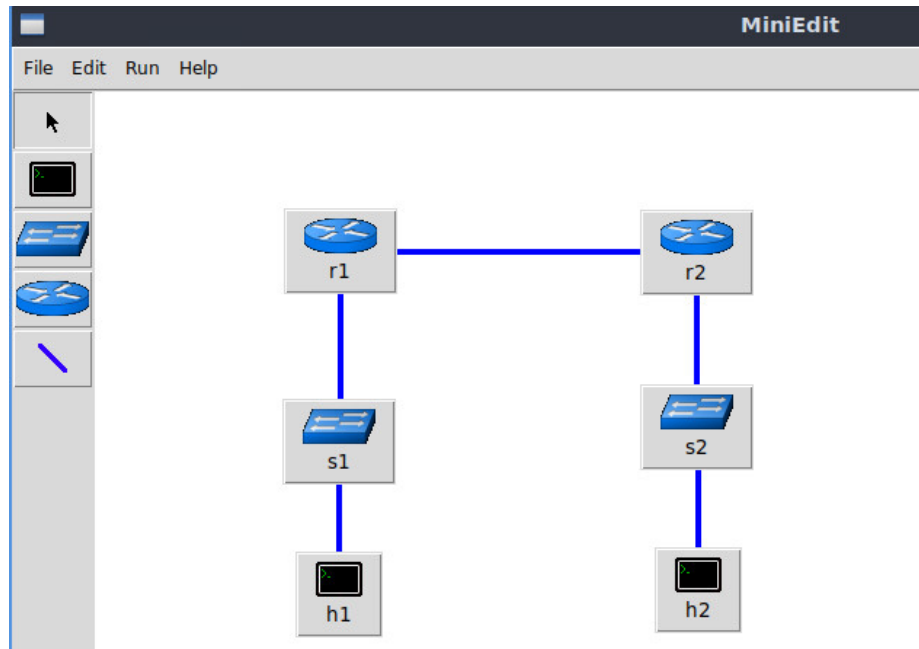


Figure 5. Mininet's topology.

### 2.3 Load the configuration file

At this point, the topology is loaded. However, the interfaces are not configured. 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 the Linux terminal.

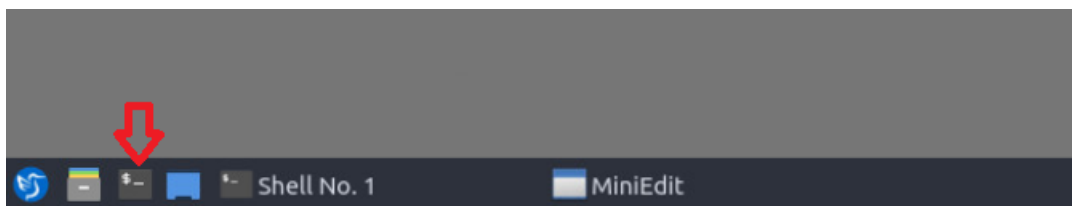


Figure 6. Opening Linux terminal.

**Step 2.** Click on the Linux terminal and navigate into *OSPF\_Labs/lab2* 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 OSPF_Labs/lab2
```

```
frr@frr-pc: ~/OSPF_Labs/lab2
File Actions Edit View Help
frr@frr-pc: ~/OSPF_Labs/lab2
frr@frr-pc:~$ cd OSPF_Labs/lab2/
frr@frr-pc:~/OSPF_Labs/lab2$
```

Figure 7. Entering the *OSPF\_Labs/lab2* directory.

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

```
./config_loader.sh lab2_conf.zip
```

```
frr@frr-pc: ~/OSPF_Labs/lab2
File Actions Edit View Help
frr@frr-pc: ~/OSPF_Labs/lab2
frr@frr-pc:~$ cd OSPF_Labs/lab2/
frr@frr-pc:~/OSPF_Labs/lab2$ ./config_loader.sh lab2_conf.zip
frr@frr-pc:~/OSPF_Labs/lab2$
```

Figure 8. Executing the shell script to load the configuration.

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

```
exit
```

```
frr@frr-pc: ~/OSPF_Labs/lab2
File Actions Edit View Help
frr@frr-pc: ~/OSPF_Labs/lab2
frr@frr-pc:~$ cd OSPF_Labs/lab2/
frr@frr-pc:~/OSPF_Labs/lab2$ ./config_loader.sh lab2_conf.zip
frr@frr-pc:~/OSPF_Labs/lab2$ exit
```

Figure 9. Exiting from the terminal.

## 2.4 Run 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 host h1 and host h2 interfaces are configured. To proceed with the emulation, click on the *Run* button located in the lower left-hand side.



Figure 10. Starting the emulation.

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

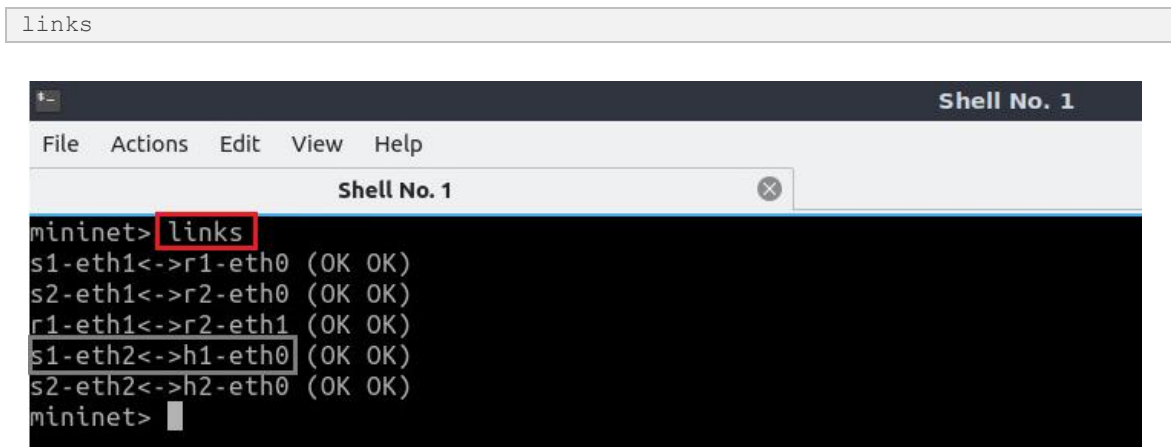


Figure 11. Displaying network interfaces.

In Figure 11, the link displayed within the gray box indicates that interface *eth2* of switch *s1* connects to interface *eth0* of host *h1* (i.e., *s1-eth2<->h1-eth0*).

## 2.5 Verify the configuration

In the following steps, you will verify the IP address to the hosts following Table 2 as the IP addresses are already configured for you. You can verify each host's IP addresses and the routing table of each router to see if the configuration is correct according to the table.

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

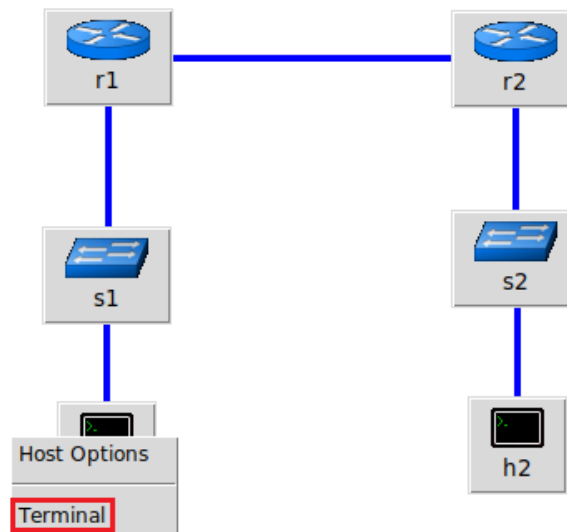


Figure 12. Opening a terminal on host h1.

**Step 2.** In the host h1 terminal, type the command shown below to verify that the IP address was assigned successfully. You will corroborate that host h1 has two interfaces, *h1-eth0* configured with the IP address 192.168.1.10 and the subnet mask 255.255.255.0, and *lo* with 127.0.0.1.

```
ifconfig
```

```

Host: h1
root@frr-pc:~# ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.10 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::7c11:30ff:fea5:d022 prefixlen 64 scopeid 0x20<link>
    ether 7e:11:30:a5:d0:22 txqueuelen 1000 (Ethernet)
    RX packets 32 bytes 3781 (3.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 12 bytes 936 (936.0 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@frr-pc:~#

```

Figure 13. Output of `ifconfig` command.

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

```
route
```



```

"Host: h1"
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.1.10 netmask 255.255.255.0 broadcast 192.168.1.255
inet6 fe80::7c11:30ff:fea5:d022 prefixlen 64 scopeid 0x20<link>
ether 7e:11:30:a5:d0:22 txqueuelen 1000 (Ethernet)
RX packets 32 bytes 3781 (3.7 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 12 bytes 936 (936.0 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@frr-pc:~# 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@frr-pc:~#

```

Figure 14. Output of `route` command.

**Step 4.** To verify host 2 default route, proceed similarly by repeating from step 1 to step 3 in the host h2 terminal. Similar results should be observed.

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

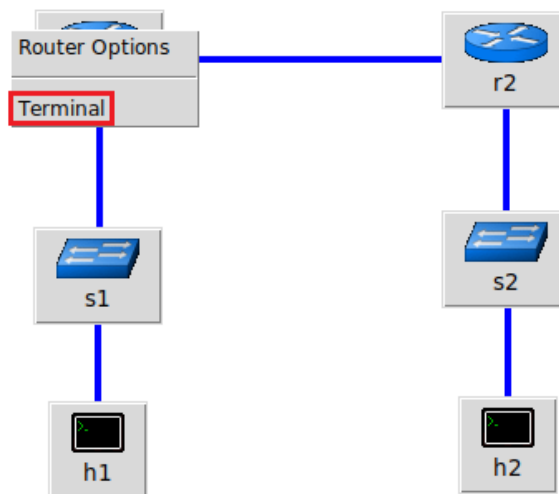
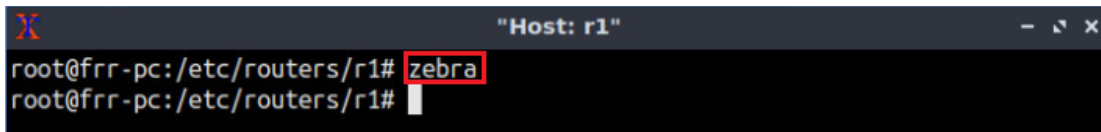


Figure 15. Opening a terminal on router r1.

**Step 6.** In this step, you will start the zebra daemon, a multi-server routing software that provides TCP/IP-based routing protocols. Further details about the zebra daemon are provided in Section 1. To start zebra, type the following command:

```
zebra
```



```

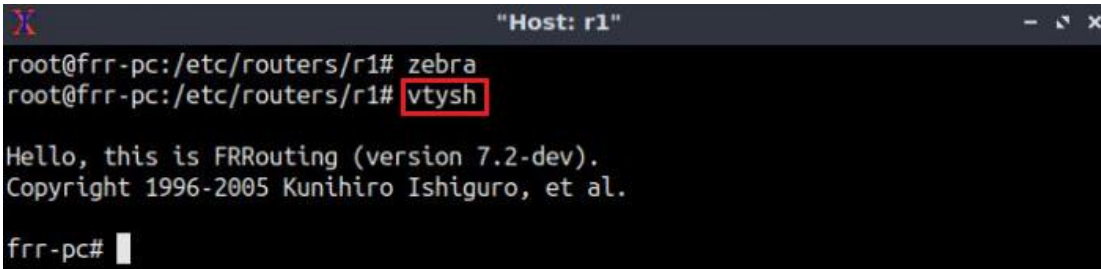
Host: r1
root@frr-pc:/etc/routers/r1# zebra
root@frr-pc:/etc/routers/r1#

```

Figure 16. Starting zebra daemon.

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

```
vtysh
```



```

Host: r1
root@frr-pc:/etc/routers/r1# zebra
root@frr-pc:/etc/routers/r1# vtysh

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

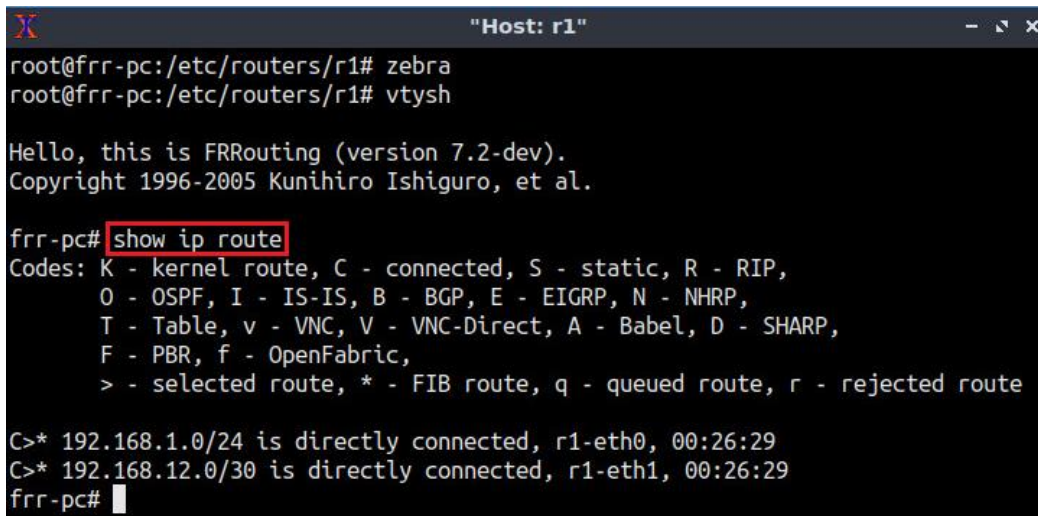
frr-pc#

```

Figure 17. Starting vtysh on router r1.

**Step 8.** Type the following command on 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@frr-pc:/etc/routers/r1# zebra
root@frr-pc:/etc/routers/r1# vtysh

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

frr-pc# 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, 00:26:29
C>* 192.168.12.0/30 is directly connected, r1-eth1, 00:26:29
frr-pc#

```

Figure 18. 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 192.168.12.0/30 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

terminal issue the commands depicted below. At the end, you will verify all the directly connected networks of router r2.

```

Host: r2
root@frr-pc:/etc/routers/r2# zebra
root@frr-pc:/etc/routers/r2# vtysh

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

frr-pc# 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, 00:00:08
C>* 192.168.12.0/30 is directly connected, r2-eth1, 00:00:08
frr-pc#

```

Figure 19. Displaying routing table of router r2.

## 2.6 Test connectivity between end-hosts

In this section, you will run a connectivity test between host 1 and host 2. You will notice that there is no connectivity because there is no routing protocol configured in the routers.

**Step 1.** In the host h1 terminal, type the command shown below. Notice that according to Table 1, the IP address 192.168.2.10 is assigned to host h2. To stop the test press `ctrl+c`.

```
ping 192.168.2.10
```

```

Host: h1
root@frr-pc:~# ping 192.168.2.10
PING 192.168.2.10 (192.168.2.10) 56(84) bytes of data:
From 192.168.1.10 icmp_seq=1 Destination Host Unreachable
From 192.168.1.10 icmp_seq=2 Destination Host Unreachable
From 192.168.1.10 icmp_seq=3 Destination Host Unreachable
From 192.168.1.10 icmp_seq=4 Destination Host Unreachable
From 192.168.1.10 icmp_seq=5 Destination Host Unreachable
From 192.168.1.10 icmp_seq=6 Destination Host Unreachable
^C
--- 192.168.2.10 ping statistics ---
7 packets transmitted, 0 received, +6 errors, 100% packet loss, time 151ms
pipe 4
root@frr-pc:~#

```

Figure 20. Connectivity test between host h1 and host h2.

## 3 Configure a routing protocol

In the previous section, you used a script to assign the IP addresses to all devices' interfaces. Then you performed an unsuccessful connectivity test. In this section, you will configure a routing protocol to establish a connection between the two networks. You

will configure static routing in router r1 and router r2 such that host h1 can reach host h2 and vice versa. First, you will initialize the daemon that enables static route configuration. Then you will configure static routes in router r1 and router r2. Specifically, static routes are configured by setting the destination network and the next hop's IP address. Finally, you will verify the configuration.

The syntax to configure static routes in the FRR router is as follows:

```
ip route <NETWORK> <GATEWAY>
```

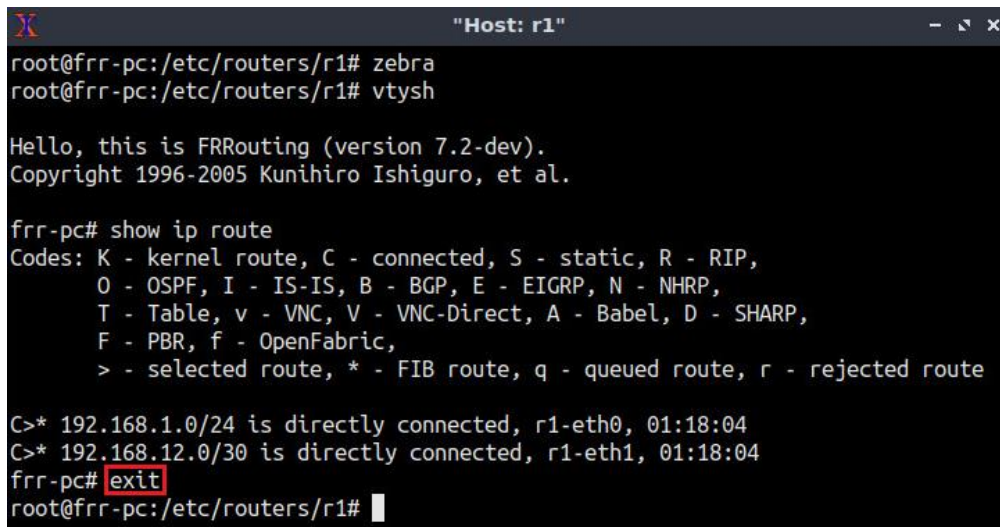
- `ip route`: is used to create or modify routing tables.
- `NETWORK`: specifies the destination network.
- `GATEWAY`: determines the next hop IP address.

### 3.1 Enable a routing daemon

In this section you will run the daemon that enables static routing configuration.

**Step 1.** In router r1 terminal, type the following command to exit from FRR terminal.

```
exit
```



```

X "Host: r1"
root@frr-pc:/etc/routers/r1# zebra
root@frr-pc:/etc/routers/r1# vtysh

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

frr-pc# 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

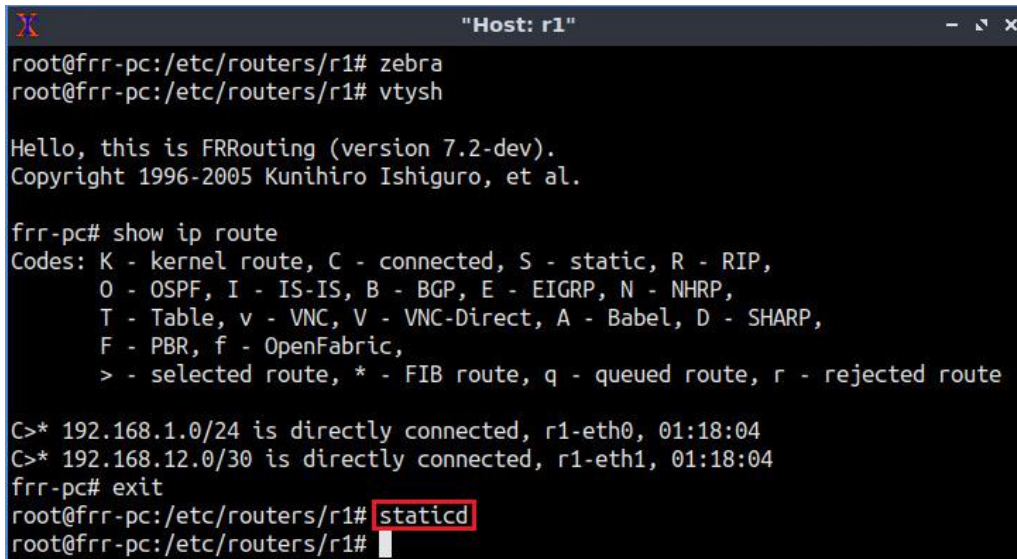
C>* 192.168.1.0/24 is directly connected, r1-eth0, 01:18:04
C>* 192.168.12.0/30 is directly connected, r1-eth1, 01:18:04
frr-pc# exit
root@frr-pc:/etc/routers/r1#

```

Figure 21. Exiting the vtysh session.

**Step 2.** Now issue the following command on router r1 terminal to enable the static routing daemon.

```
staticd
```



```

Host: r1
root@frr-pc:/etc/routers/r1# zebra
root@frr-pc:/etc/routers/r1# vtysh

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

frr-pc# 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, 01:18:04
C>* 192.168.12.0/30 is directly connected, r1-eth1, 01:18:04
frr-pc# exit
root@frr-pc:/etc/routers/r1# staticd
root@frr-pc:/etc/routers/r1#

```

Figure 22. Starting static routing daemon.

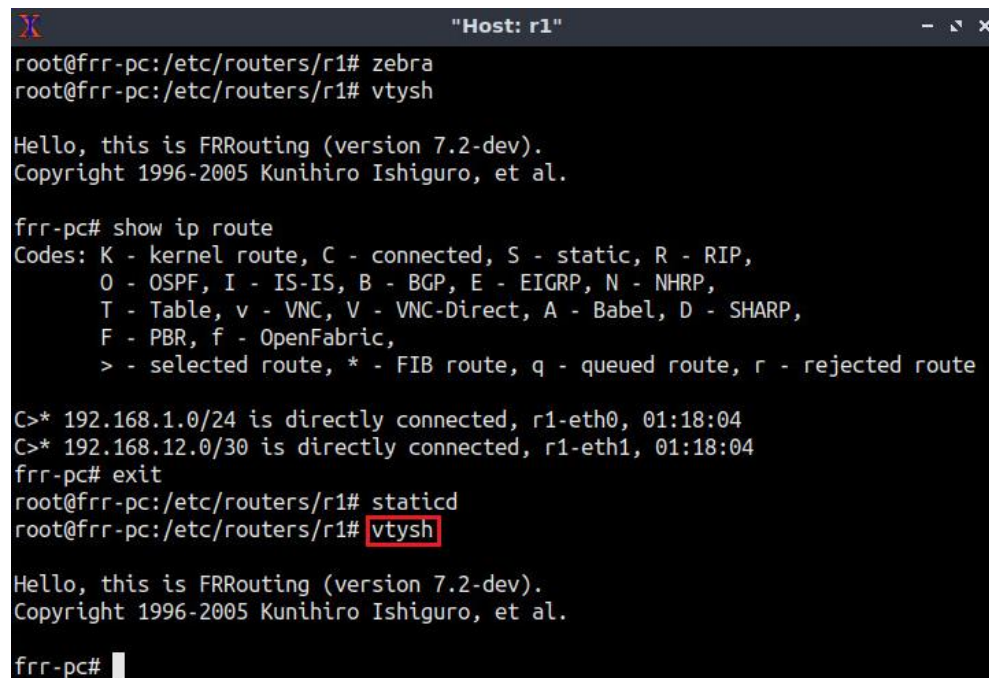
Now, the static routing daemon is running and is ready to set up a configuration.

### 3.2 Configure static route

In this section, you will configure the static routes on router r1 and router r2. This configuration will establish connectivity between the networks 192.168.1.0/24 and 192.168.2.0/24.

**Step 1.** In order to enter router r1 terminal, issue the following command:

```
vtysh
```



```

Host: r1
root@frr-pc:/etc/routers/r1# zebra
root@frr-pc:/etc/routers/r1# vtysh

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

frr-pc# 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, 01:18:04
C>* 192.168.12.0/30 is directly connected, r1-eth1, 01:18:04
frr-pc# exit
root@frr-pc:/etc/routers/r1# staticd
root@frr-pc:/etc/routers/r1# vtysh

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

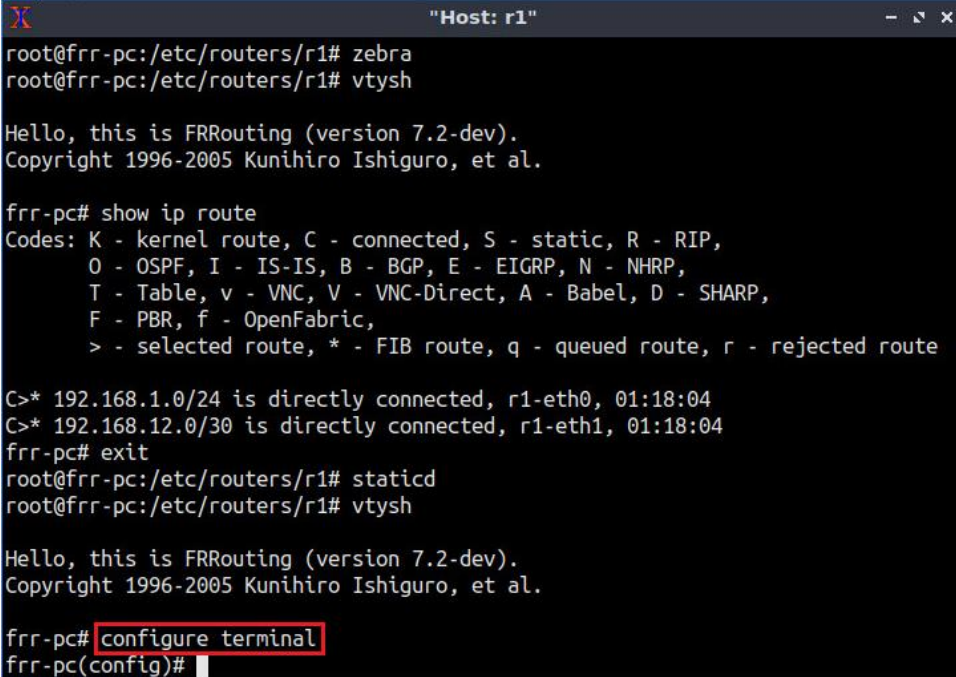
frr-pc#

```

Figure 23. Starting vtysh on router r1.

**Step 2.** To enable router r1 configuration mode, issue the following command:

```
configure terminal
```



```
Host: r1
root@frr-pc:/etc/routers/r1# zebra
root@frr-pc:/etc/routers/r1# vtysh

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

frr-pc# 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, 01:18:04
C>* 192.168.12.0/30 is directly connected, r1-eth1, 01:18:04
frr-pc# exit
root@frr-pc:/etc/routers/r1# staticd
root@frr-pc:/etc/routers/r1# vtysh

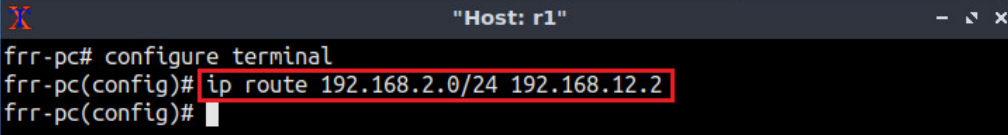
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)#
```

Figure 24. Enabling configuration mode on router r1.

**Step 3.** In order to configure a static route to reach out the network 192.168.2.0/24 thru the IP address 192.168.12.2, type the following command:

```
ip route 192.168.2.0/24 192.168.12.2
```



```
Host: r1
frr-pc# configure terminal
frr-pc(config)# ip route 192.168.2.0/24 192.168.12.2
frr-pc(config)#
```

Figure 25. Configuring a static route on router r1.

**Step 4.** To exit from configuration mode, issue the following command:

```
exit
```



```
Host: r1
frr-pc# configure terminal
frr-pc(config)# ip route 192.168.2.0/24 192.168.12.2
frr-pc(config)# exit
frr-pc#
```

Figure 26. Exiting from configuration mode.

**Step 5.** The figure below summarizes the steps that must be followed in router r2 terminal in order to configure static route. From the perspective of router r2 the network 192.168.1.0 is reachable via the IP address 192.168.12.1.

```

Host: r2

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

frr-pc# 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, 00:00:38
C>* 192.168.12.0/30 is directly connected, r2-eth1, 00:00:38
frr-pc# exit
root@frr-pc:/etc/routers/r2# staticd
root@frr-pc:/etc/routers/r2# vtysh

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

frr-pc# configure terminal
frr-pc(config)# ip route 192.168.1.0/24 192.168.12.1
frr-pc(config)# exit
frr-pc#

```

Figure 27. Configuring static routing on router r2.

### 3.3 Verify the configuration

In this section, you will verify the configuration on router r1 and router r2.

**Step 1.** In router r1 terminal, type the following command to show the routing table entries. Notice that the network 192.168.2.0/24 shown with the `S>*` is reachable via the IP address 192.168.12.2. 'S' indicates a static route. The egress interface is `r1-eth1`:

```
show ip route
```

```

Host: r1

frr-pc# configure terminal
frr-pc(config)# ip route 192.168.2.0/24 192.168.12.2
frr-pc(config)# exit
frr-pc# 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

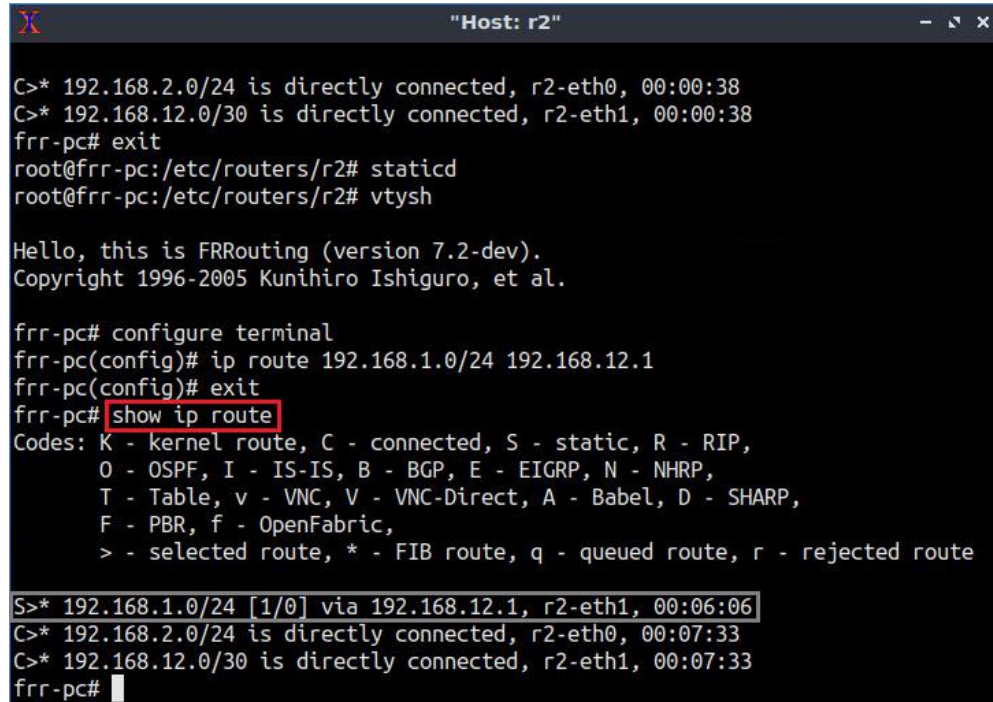
C>* 192.168.1.0/24 is directly connected, r1-eth0, 00:08:55
S>* 192.168.2.0/24 [1/0] via 192.168.12.2, r1-eth1, 00:08:28
C>* 192.168.12.0/30 is directly connected, r1-eth1, 00:08:55
frr-pc#

```

Figure 28. Verifying the routing table of router r1.

**Step 2.** Similarly, in router r2 terminal, type the following command to show the routing table entries. Notice that the network 192.168.1.0/24 (static route) is reachable via the IP address 192.168.12.1. The egress interface is `r2-eth1`:

```
show ip route
```



```

Host: r2
C>* 192.168.2.0/24 is directly connected, r2-eth0, 00:00:38
C>* 192.168.12.0/30 is directly connected, r2-eth1, 00:00:38
frr-pc# exit
root@frr-pc:/etc/routers/r2# staticd
root@frr-pc:/etc/routers/r2# vtysh

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

frr-pc# configure terminal
frr-pc(config)# ip route 192.168.1.0/24 192.168.12.1
frr-pc(config)# exit
frr-pc# 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

S>* 192.168.1.0/24 [1/0] via 192.168.12.1, r2-eth1, 00:06:06
C>* 192.168.2.0/24 is directly connected, r2-eth0, 00:07:33
C>* 192.168.12.0/30 is directly connected, r2-eth1, 00:07:33
frr-pc#

```

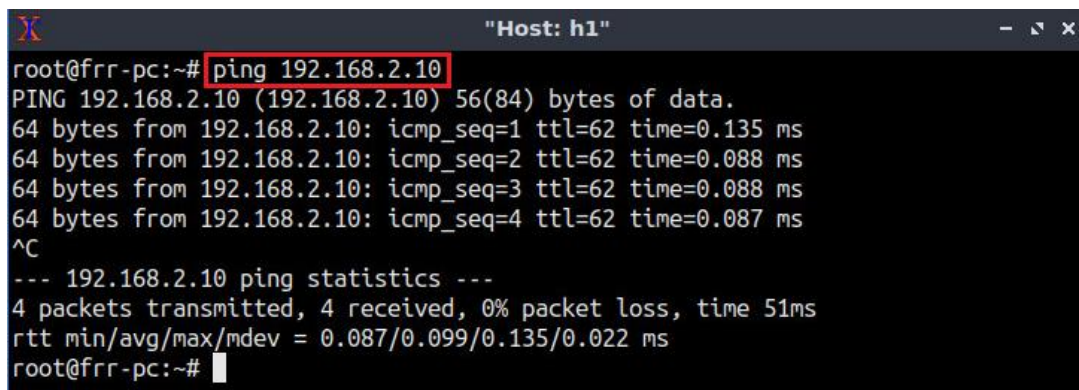
Figure 29. Verifying the routing table of router r2.

#### 4 Test connectivity and verify routes between end-hosts

In this section you will perform a connectivity test from host h1 to host h2. Additionally, you will check the details about the path that a packet takes from host h1, the source, to host h2, the destination.

**Step 1.** On host h1 terminal type the following command. The IP address 192.168.2.10 corresponds to host h2:

```
ping 192.168.2.10
```



```

Host: h1
root@frr-pc:~# 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=62 time=0.135 ms
64 bytes from 192.168.2.10: icmp_seq=2 ttl=62 time=0.088 ms
64 bytes from 192.168.2.10: icmp_seq=3 ttl=62 time=0.088 ms
64 bytes from 192.168.2.10: icmp_seq=4 ttl=62 time=0.087 ms
^C
--- 192.168.2.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 51ms
rtt min/avg/max/mdev = 0.087/0.099/0.135/0.022 ms
root@frr-pc:~#

```

Figure 30. Output of `ping` command on host h1.

**Step 2.** On host h1 terminal type the following command. Notice that it takes three hops to reach out the destination which, in this case is host h2.



```
tracert 192.168.2.10
```

```

root@frr-pc:~# 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=62 time=0.135 ms
64 bytes from 192.168.2.10: icmp_seq=2 ttl=62 time=0.088 ms
64 bytes from 192.168.2.10: icmp_seq=3 ttl=62 time=0.088 ms
64 bytes from 192.168.2.10: icmp_seq=4 ttl=62 time=0.087 ms
^C
--- 192.168.2.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 51ms
rtt min/avg/max/mdev = 0.087/0.099/0.135/0.022 ms
root@frr-pc:~# traceroute 192.168.2.10
traceroute to 192.168.2.10 (192.168.2.10), 30 hops max, 60 byte packets
 1 192.168.1.1 (192.168.1.1) 1.414 ms 1.414 ms 1.410 ms
 2 192.168.12.2 (192.168.12.2) 1.408 ms 1.410 ms 1.424 ms
 3 192.168.2.10 (192.168.2.10) 1.609 ms 1.600 ms 1.584 ms
root@frr-pc:~#

```

Figure 31. Verifying the path details using `tracert` command.

This concludes lab 2. Stop the emulation and then exit out of MiniEdit and Linux terminal.

## References

1. Linux foundation collaborative projects, "FRR routing documentation", 2017. [Online]. Available: <http://docs.frrouting.org/en/latest/>
2. Jakma P., Lamparter D. "Introduction to the quagga routing suite," 2014, *IEEE Network* 28.
3. Ishiguro K., "Gnu zebra,". [Online]. Available: <http://www.zebra.org> (2002).
4. Mininet walkthrough. [Online]. Available: <http://Mininet.org>.
5. Malkin G., Minnear R., "RIPng for IPv6," RFC 2080, 1997. [Online]. Available: <http://www.ietf.org/rfc/rfc2080.txt>.
6. Rekhter Y., Li T., Hares S. "A border gateway protocol 4 (BGP-4)," RFC 4271 updated by RFCs 6286, 6608, 6793, 2006. [Online]. Available: <http://www.ietf.org/rfc/rfc4271.txt>.
7. Oran D., "OSI IS-IS intra-domain routing protocol," RFC 1142, 1990. [Online]. Available: <http://www.ietf.org/rfc/rfc1142.txt>.
8. Lantz B., Gee G. "MiniEdit: a simple network editor for Mininet," 2013. [Online]. Available: <https://github.com/Mininet/Mininet/blob/master/examples>.



UNIVERSITY OF  
**SOUTH CAROLINA**

## OPEN SHORTEST PATH FIRST

### Lab 3: Configuring Single-Area OSPFv2

Document Version: **10-22-2021**



Award 1829698

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

## Contents

Overview .....	3
Objectives.....	3
Lab settings .....	3
Lab roadmap .....	3
1 Introduction .....	3
1.1 Routing protocols.....	3
1.2 The OSPF routing protocol .....	5
1.3 OSPF packets .....	5
1.4 OSPF states, LSDB exchange, and synchronization .....	6
1.5 OSPF passive interfaces.....	7
2 Lab topology.....	8
2.1 Lab settings.....	9
2.2 Loading the topology.....	10
2.3 Load the zebra daemon and verify the configuration .....	13
3 Configuring single-area OSPF.....	17
3.1 Configuring OSPF in routers r1, r2, and r3 .....	17
3.2 Verifying OSPF neighbors, routing information, and connectivity .....	22
4 Configuring OSPF passive interfaces.....	25
References .....	27

## Overview

This lab describes the routing protocol Open Shortest Path First (OSPF) which falls into the category of Interior Gateway Protocols (IGPs). In this lab, the user will configure single-area OSPF and verify connectivity between devices in different networks. Additionally, the user will configure and verify OSPF neighbors, routing information, and passive interfaces on routers.

## Objectives

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

1. Explain the concept of OSPF.
2. Configure single-area OSPF.
3. Verify OSPF neighbors and routing information.
4. Understand and configure OSPF passive interfaces.

## Lab settings

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

Table 1. Credentials to access the Client 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 single-area OSPF.
4. Section 4: Configuring OSPF passive interfaces.

### 1 Introduction

#### 1.1 Routing protocols

Routing protocols allow the routers in a network to learn about the existing available routes to enable intercommunication. There are two types of routing protocols, static and dynamic routing protocols. Static routing is the process of manually inserting the routing information in each router in order for the router to have a full view of the network. This approach is feasible on small networks where it does not require too much effort from the administrator. However, as the number of connected routers and components increase, static routing becomes less favorable<sup>1</sup>.

Consider Figure 1. Dynamic routing is the method of using routing protocols so that routers are able to learn the routes by themselves instead of having them manually inserted. This approach is divided into two types: Interior Gateway Protocol (IGP) which is used within a routing domain, i.e., within networks of a single organization, and Exterior Gateway Protocol (EGP) which is used between two different Autonomous Systems (ASes) controlled by different administrations<sup>1</sup>.

IGPs are divided into two different algorithms, distance vector and link-state. A router running a distance vector protocol will share its knowledge about the whole network to its neighbors and accordingly updates its own table based on its neighbors. However, a router running a link-state protocol will share its knowledge regarding its direct neighbors to all the routers through a technique called flooding<sup>1</sup>.

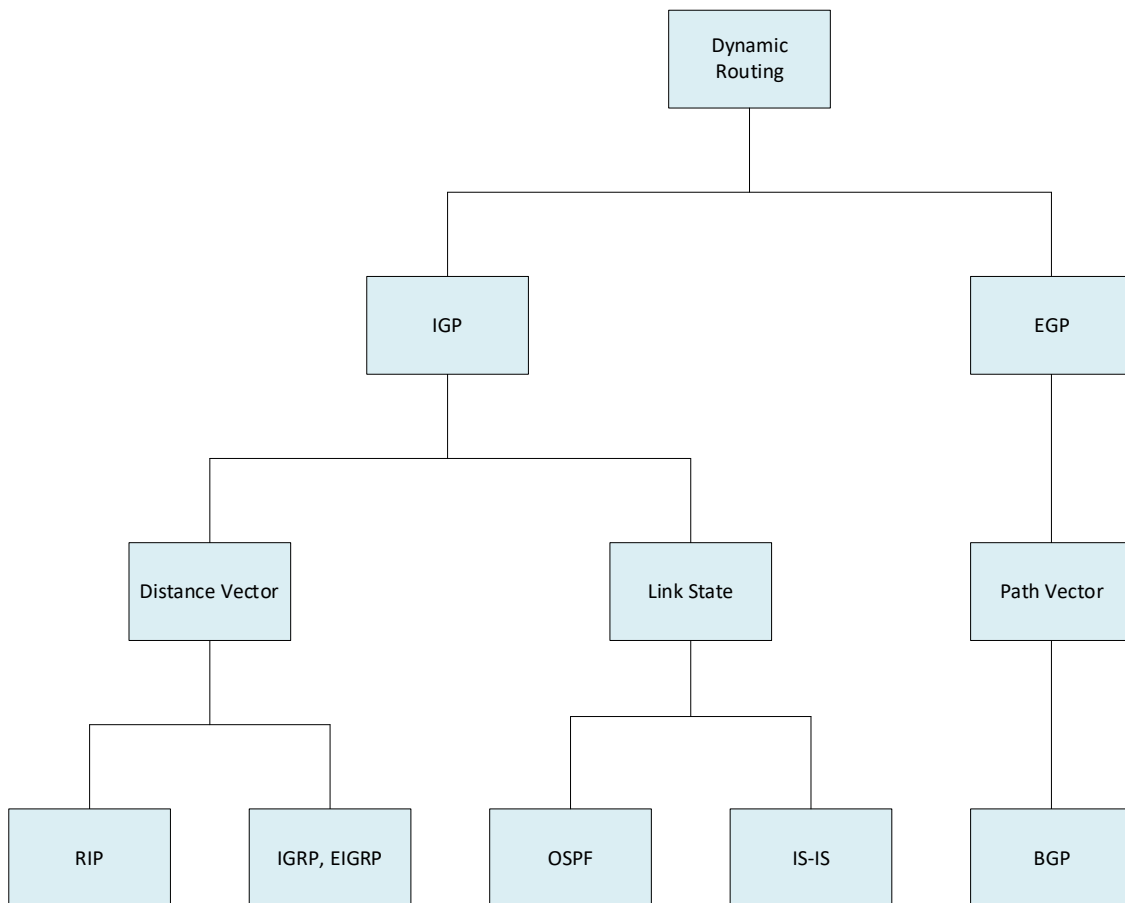


Figure 1. Dynamic routing protocols.

## 1.2 The OSPF routing protocol

OSPF is a widely supported IGP in the family of routing protocols, meaning that it is designed to be used within a single AS. It is a link-state protocol which sends information about directly connected links to all routers in the network, instead of sending the entire routing table as in the case of distance vector protocol<sup>2</sup>.

An OSPF area is a logical collection of OSPF networks, routers, and links that has the same area-ID. A router within an area must maintain a topological database for the area to which it belongs<sup>2</sup>.

The OSPF routing protocol uses Link-State Advertisements (LSAs) to communicate the router's local routing topology with all other local routers in the same OSPF area<sup>2</sup>.

LSA is the standard way of communication in the OSPF routing protocol. OSPF is designed to be scalable, thus there are multiple types of LSA packets that accommodate OSPF features. For example, type 1 LSA packets are exchanged between the same area of origin and do not leave the area<sup>2</sup>.

Consider Figure 2. Routers share LSA packets on OSPF-enabled interfaces to form a view of the current topology. Thus, when routers converge, they will have full visibility of all the network's components, i.e., the routers and the links.

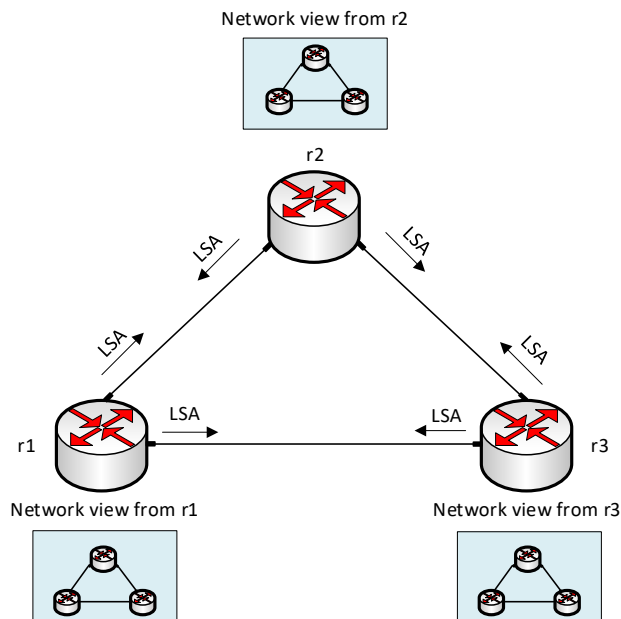


Figure 2. A network running OSPF.

## 1.3 OSPF packets

The OSPF version 2<sup>1,2</sup> (OSPFv2) is the one used to handle IPv4 traffic. The protocol number is 89. OSPF packets are classified into the following types:

- Hello packets: They are used to discover, build, and keep OSPF neighbor adjacencies.
- Database Description Packets (DBD): It is used to describe the Link-State Data Base (LSDB) of a router. The LSDB is used by routers participating in the OSPF process to verify that their databases are synchronized.
- Link-State Request (LSR): A router uses it to request the latest Link State Advertisement (LSA).
- Link-State Update (LSU): They are used to flood LSAs and sending LSA responses to LSR packets.
- Link-State Acknowledgement (LSAck): It is used to acknowledge LSA packets.

#### 1.4 OSPF states, LSDB exchange, and synchronization

The routers participating in the OSPF process pass through multiple states before forming full adjacency, as shown in Figure 3. Routers r1 and r2 will start exchanging OSPF packets following the states summarized below.

- Down: No information is sent or received by routers r1 and r2.
- Init: Router r2 interface has detected a Hello packet broadcasted by router r1. The broadcast IP address used by OSPFv2 is 225.0.0.5. The hello packet also contains the Router ID. However, bidirectional communication has not been established.
- Two-way: Router r2 responded by sending a hello packet to the unicast address of router r1 (i.e., 20.0.0.1). At the end of this state, the Designated Router (DR) and the Backup Designated Router (BDR) are chosen. The DR and the BDR are used to reduce the amount of packet exchanged in the OSPF process, meaning that non-DR and non-BDR will only exchange routes with the former and the latter. The router with the highest router ID will be selected as the DR (i.e., router r2). The router with the second router ID will become the BDR (i.e., router r1).
- ExStart: In this state, routers r1 and r2 establish the initial sequence number used to ensure that routers get the latest update when they exchange OSPF packets.
- Exchange: During this state, routers exchange LSDB by sending DBD packets. DBD packet contain information about the LSA entry header included in the router's LSDB. Additionally, the router uses a sequence number to determine the freshness of the OSPF packet. Each DBD packet is acknowledged with LSAck packet.
- Loading: In this state, router information exchange finalizes. Routers build two lists that comprise the link-state request and link-state retransmission. The link-state request list includes any information that might be incomplete or outdated, whereas any update that is sent will be registered in the retransmission list until it gets acknowledged.
- Full: At this point, neighbor adjacencies are fully established, meaning that adjacent routers will have the same LSDB.

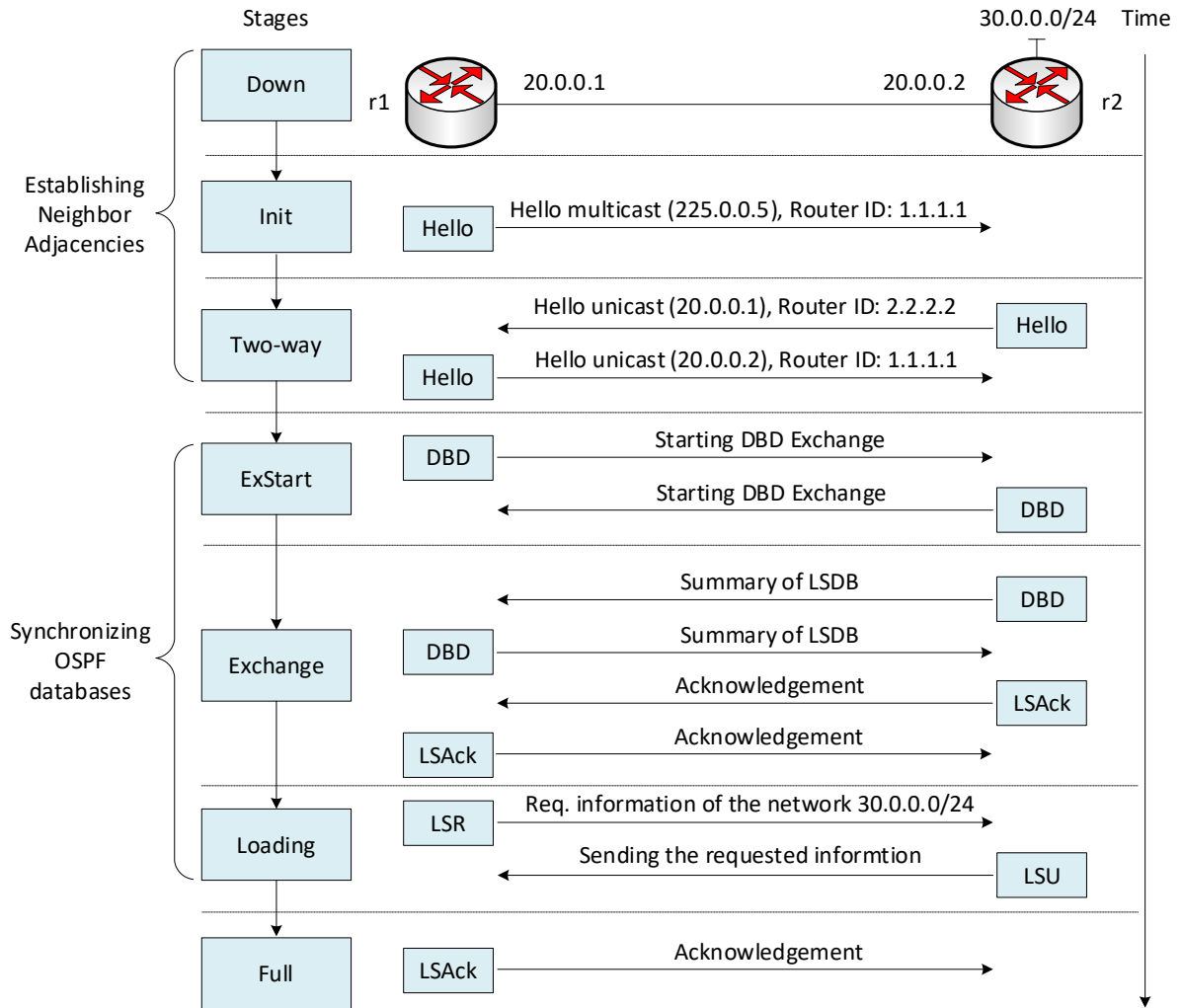


Figure 3. OSPF states and packet exchange timeline.

### 1.5 OSPF passive interfaces

When an interface is enabled to participate in the OSPF routing process, the router will start sending routing updates via this interface. This is useful when the interface is connected to another router since it ensures the consistency of the network topology view among all routers running OSPF<sup>2</sup>.

A passive interface is an interface that does not participate in sending LSA packets. It is usually configured with interfaces that are connected to a Local Area Network (LAN) to avoid overwhelming the LAN with unnecessary routing updates<sup>2</sup>.

Consider Figure 4. Router r1 configures the interface facing host h1 as passive. Thus, host h1 connected to this interface will no longer receive LSA updates. However, router r1 will still send updates about the passive interface to other OSPF-enabled interfaces, i.e., the one facing router r2.



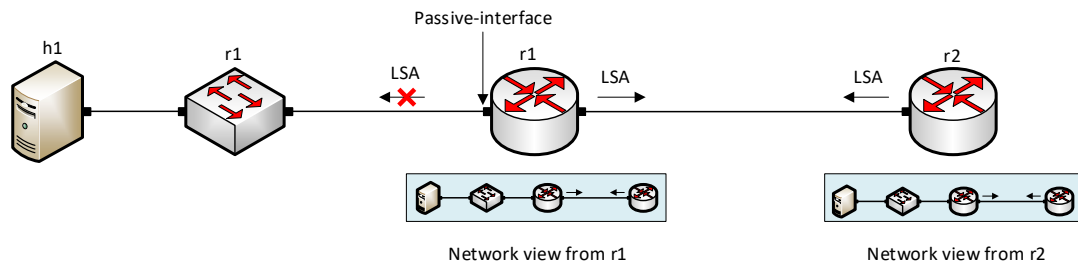


Figure 4. Passive interface in OSPF.

## 2 Lab topology

Consider Figure 5. The topology consists of three networks connected to their respective routers. The communication between networks is established via single-area OSPF configured in all routers. The routers participating in the OSPF process are in the backbone area (i.e., area 0).

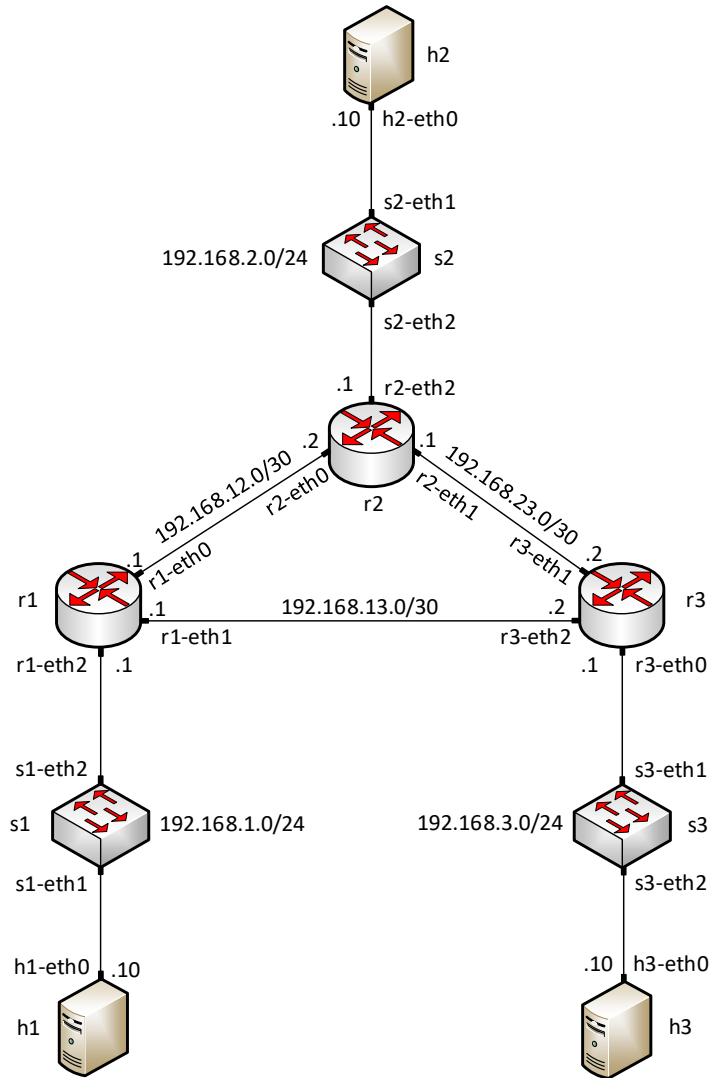


Figure 5. Lab topology.

## 2.1 Lab settings

Table 2 contains information about the devices and their corresponding interfaces, IPv4 addresses, subnet, and default gateway.

Table 2. Topology Information.

Device	Interface	IPv4 Address	Subnet	Default Gateway
Router 1 (r1)	r1-eth0	192.168.12.1	/30	N/A
	r1-eth1	192.168.13.1	/30	N/A
	r1-eth2	192.168.1.1	/24	N/A

Router 2 (r2)	r2-eth0	192.168.12.2	/30	N/A
	r2-eth1	192.168.23.1	/30	N/A
	r2-eth2	192.168.2.1	/24	N/A
Router 3 (r3)	r3-eth0	192.168.3.1	/24	N/A
	r3-eth1	192.168.23.2	/30	N/A
	r3-eth2	192.168.13.2	/30	N/A
Host 1 (h1)	h1-eth0	192.168.1.10	/24	192.168.1.1
Host 2 (h2)	h2-eth0	192.168.2.10	/24	192.168.2.1
Host 3 (h3)	h3-eth0	192.168.3.10	/24	192.168.3.1

## 2.2 Loading the topology

**Step 1.** Launch MiniEdit by double clicking on the shortcut located on the desktop. When prompted for a password, type `password`.



Figure 6. MiniEdit shortcut.

**Step 2.** In MiniEdit's menu bar, click on *File* then *Open* to load the lab topology. Locate the `lab3.mn` topology file in the default directory, `/home/frr/OSPF_Labs/lab3` and click on *Open*.

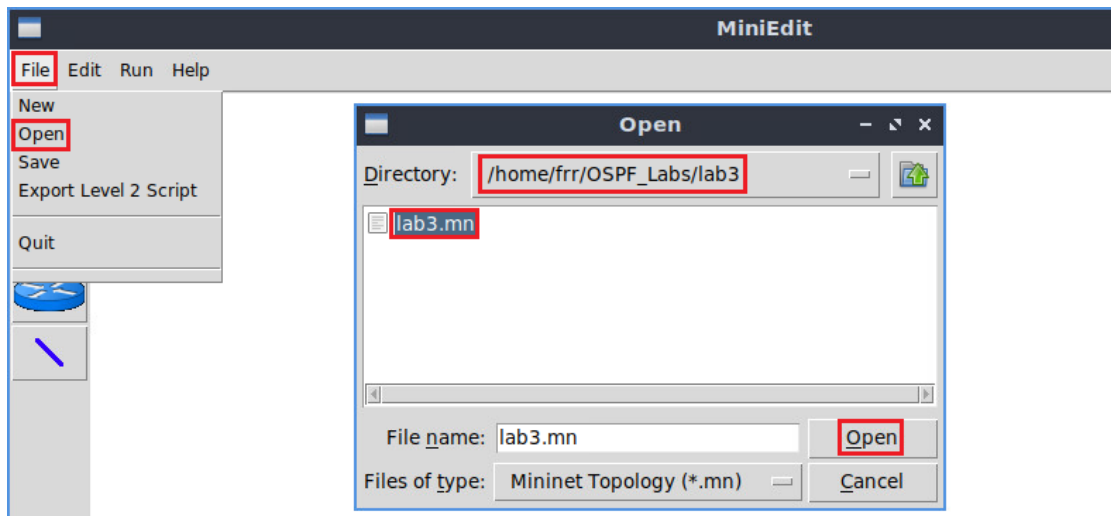


Figure 7. MiniEdit's open dialog.

**Step 3.** Click on the *Run* button located on the lower left-hand side to run the emulation.



Figure 8. Starting the emulation.

At this point, the topology is loaded with all the required network components. Next, you will execute a script that will configure the interfaces of the routers.

Automating the configuration of interfaces with the script is intended to facilitate the lab experience and to focus on the OSPF experiment.

**Step 4.** Open the Linux terminal by clicking on the icon located in the taskbar.

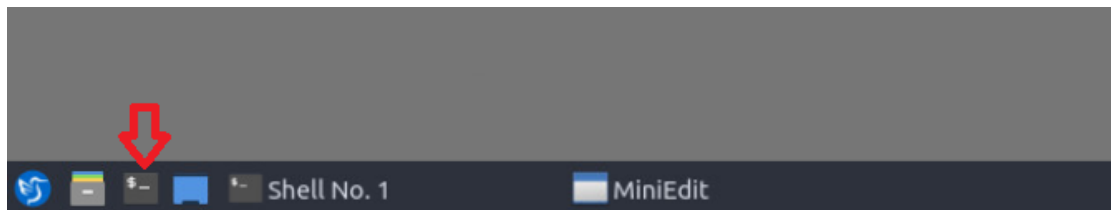
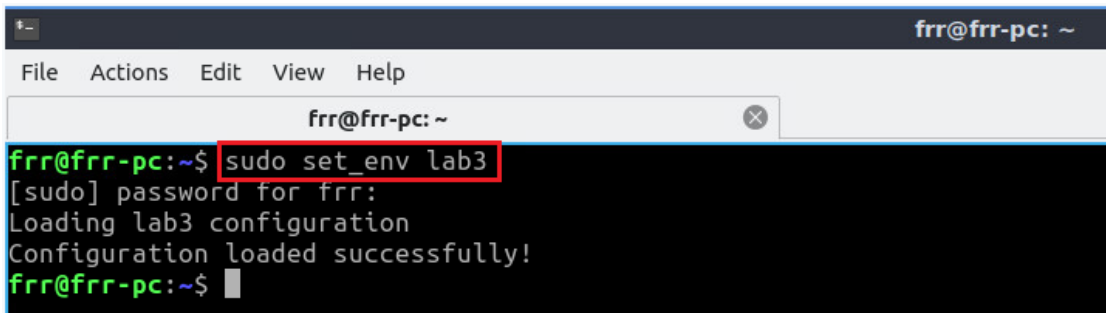


Figure 9. Opening Linux terminal.

**Step 5.** Issue the command shown below to set the lab environment. This command configures the IP addresses of the routers' interface according to Table 2. If a password is required, type `password`.

```
sudo set_env lab3
```

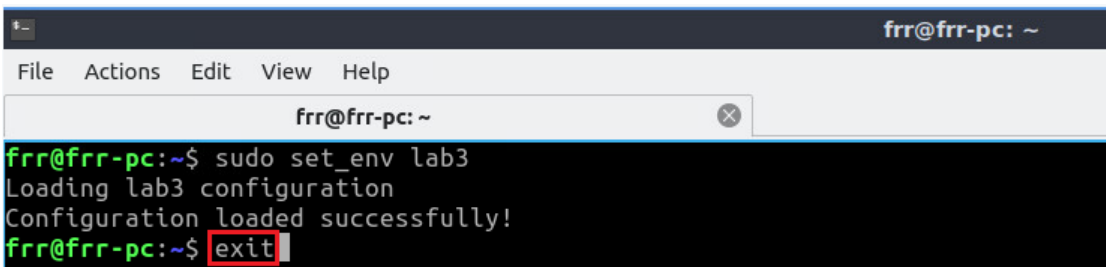
A terminal window titled 'frr@frr-pc: ~' with a menu bar (File, Actions, Edit, View, Help) and a title bar (frr@frr-pc: ~). The terminal shows the command 'sudo set\_env lab3' being entered and executed. The output is: '[sudo] password for frr:', 'Loading lab3 configuration', and 'Configuration loaded successfully!'. The prompt returns to 'frr@frr-pc:~\$'.

```
frr@frr-pc:~$ sudo set_env lab3
[sudo] password for frr:
Loading lab3 configuration
Configuration loaded successfully!
frr@frr-pc:~$
```

Figure 10. Setting the lab environment.

**Step 6.** Type the following command to exit the Linux terminal.

```
exit
```

A terminal window titled 'frr@frr-pc: ~' with a menu bar (File, Actions, Edit, View, Help) and a title bar (frr@frr-pc: ~). The terminal shows the command 'sudo set\_env lab3' being entered and executed, followed by 'exit'. The output is: 'Loading lab3 configuration', 'Configuration loaded successfully!', and the prompt returns to 'frr@frr-pc:~\$'.

```
frr@frr-pc:~$ sudo set_env lab3
Loading lab3 configuration
Configuration loaded successfully!
frr@frr-pc:~$ exit
```

Figure 11. Exiting the Linux terminal.

**Step 7.** Click on Mininet's terminal, i.e., the one launched when MiniEdit was started.

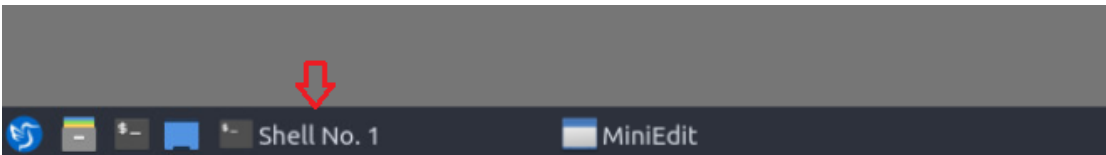
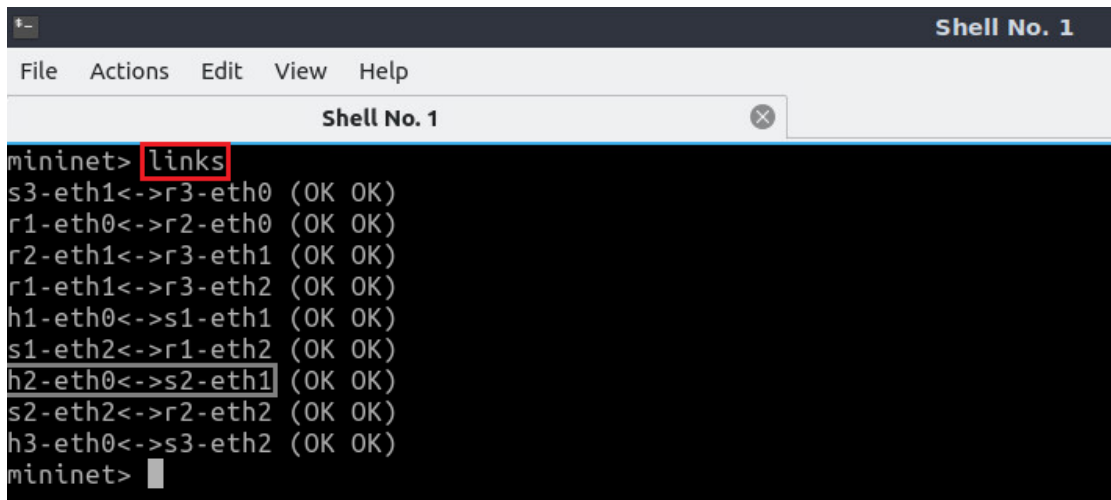


Figure 12. Opening Mininet's terminal.

**Step 8.** Issue the following command to display the interface names and connections.

```
links
```

A screenshot of a terminal window titled "Shell No. 1". The terminal shows the command "links" being executed in a "mininet" environment. The output lists several network links between hosts and switches, each followed by "(OK OK)". The link "h2-eth0<->s2-eth1" is highlighted with a gray box. The terminal text is as follows:

```
mininet> links
s3-eth1<->r3-eth0 (OK OK)
r1-eth0<->r2-eth0 (OK OK)
r2-eth1<->r3-eth1 (OK OK)
r1-eth1<->r3-eth2 (OK OK)
h1-eth0<->s1-eth1 (OK OK)
s1-eth2<->r1-eth2 (OK OK)
h2-eth0<->s2-eth1 (OK OK)
s2-eth2<->r2-eth2 (OK OK)
h3-eth0<->s3-eth2 (OK OK)
mininet>
```

Figure 13. Displaying network interfaces.

In the figure above, the link displayed within the gray box indicates that interface *eth2* of host *h2* connects to interface *eth1* of switch *s2* (i.e., *h2-eth0<->s2-eth1*).

### 2.3 Load the zebra daemon and verify the configuration

In this section, you will verify that the IP addresses listed in Table 2 and inspect the routing table of the routers.

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

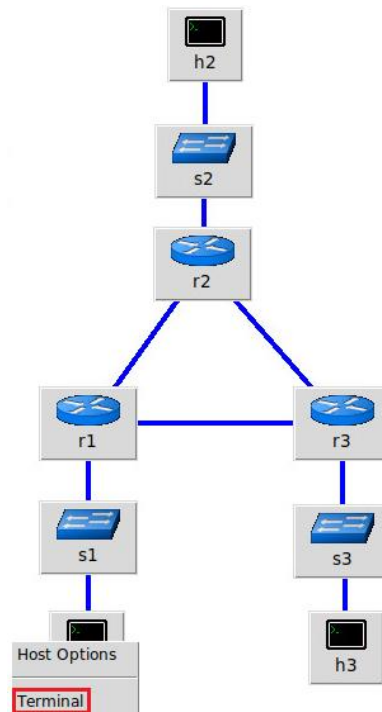
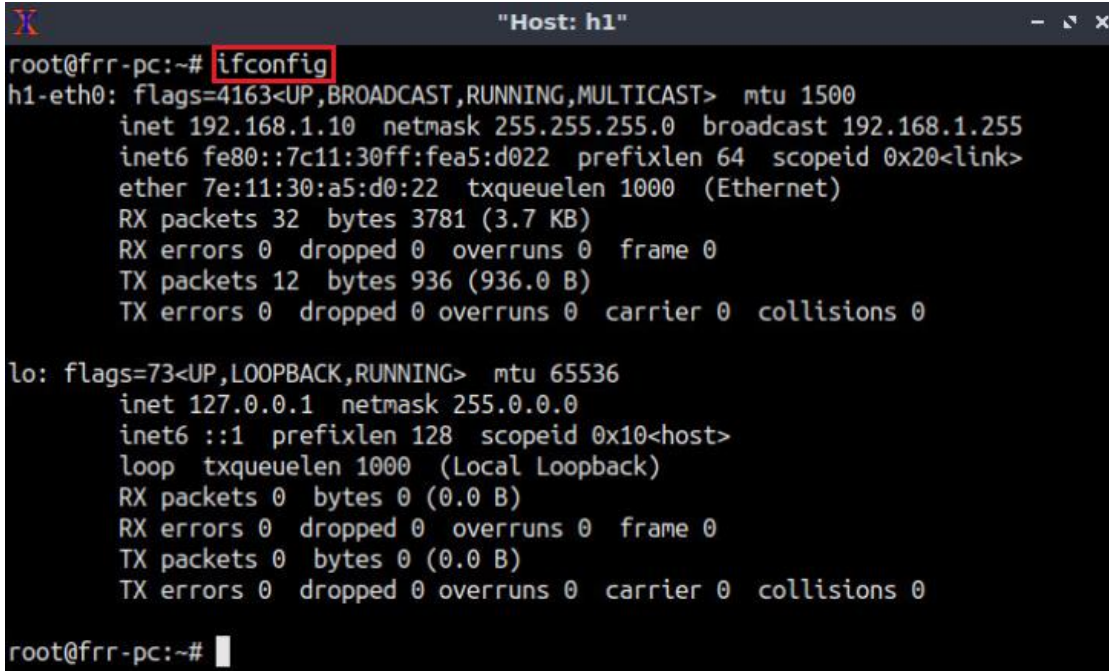


Figure 14. Opening a terminal on host h1.

**Step 2.** In the host h1 terminal, type the command shown below to verify that the IP address was assigned successfully. You will verify that host h1 has two interfaces. Interface *h1-eth0* is configured with the IP address 192.168.1.10 and the subnet mask 255.255.255.0 and a loopback *lo* interface.

```
ifconfig
```



```
root@frr-pc:~# ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.10 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::7c11:30ff:fea5:d022 prefixlen 64 scopeid 0x20<link>
    ether 7e:11:30:a5:d0:22 txqueuelen 1000 (Ethernet)
    RX packets 32 bytes 3781 (3.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 12 bytes 936 (936.0 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@frr-pc:~#
```

Figure 15. Displaying host h1 interfaces.

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

```
route
```

```

Host: h1
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
  inet 192.168.1.10 netmask 255.255.255.0 broadcast 192.168.1.255
  inet6 fe80::7c11:30ff:fea5:d022 prefixlen 64 scopeid 0x20<link>
  ether 7e:11:30:a5:d0:22 txqueuelen 1000 (Ethernet)
  RX packets 32 bytes 3781 (3.7 KB)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 12 bytes 936 (936.0 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@frr-pc:~# 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@frr-pc:~#

```

Figure 16. Displaying host h1 default gateway.

**Step 4.** To verify the hosts h2 and h3 IP addresses and default routes, proceed by repeating from step 1 to step 3. Similar results should be observed.

**Step 5.** Verify the IP addresses assigned to the interfaces of router r1 are according to Table 2. To proceed, hold right-click on router r1 and select *Terminal*.



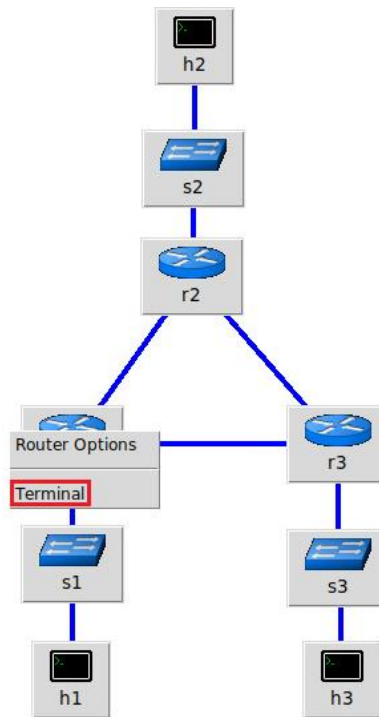


Figure 17. Opening a terminal on router r1.

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

```
zebra
```

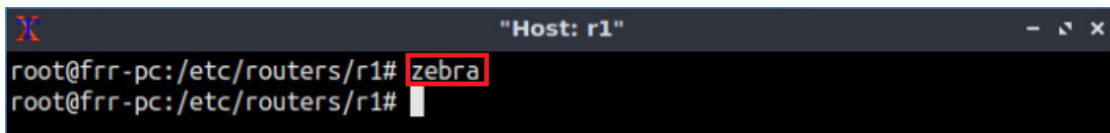


Figure 18. Starting zebra daemon.

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

```
vtysh
```

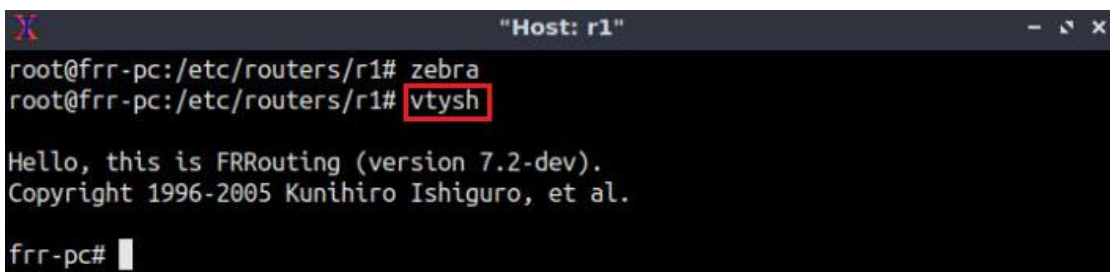


Figure 19. Starting vtysh on router r1.

**Step 8.** Type the following command on router r1 terminal to verify the routing table. It will list all the directly connected networks. Notice that 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
```

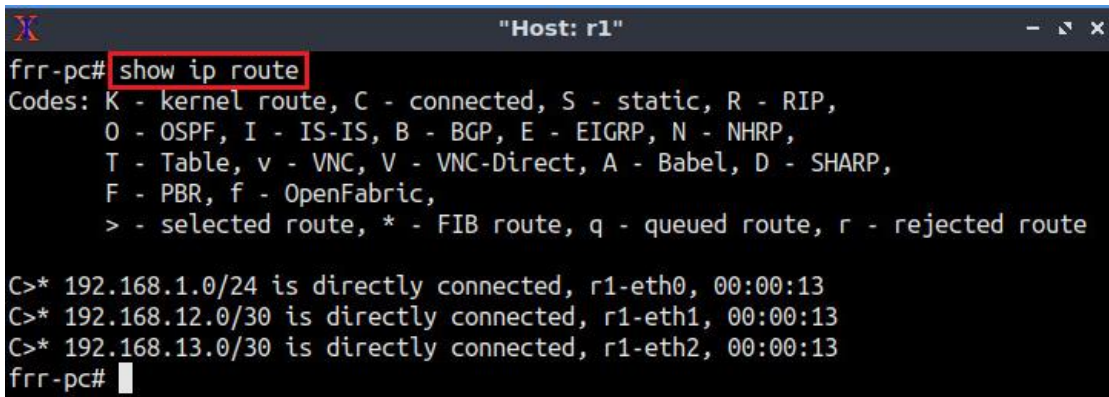


Figure 20. Displaying routing table of router r1.

**Step 9.** To verify the routing table of the other routers, proceed similarly by repeating from step 5 to step 8 on the router terminal.

### 3 Configuring single-area OSPF

In this section, you will configure the OSPF routing protocol in routers r1, r2, and r3. OSPF will be handling the internal routing in this configuration. To configure OSPF, you will be using single area which is area 0 (also known as backbone area).

#### 3.1 Configuring OSPF in routers r1, r2, and r3

**Step 1.** Navigate to router r1’s terminal by selecting the router’s r1 terminal from the taskbar.

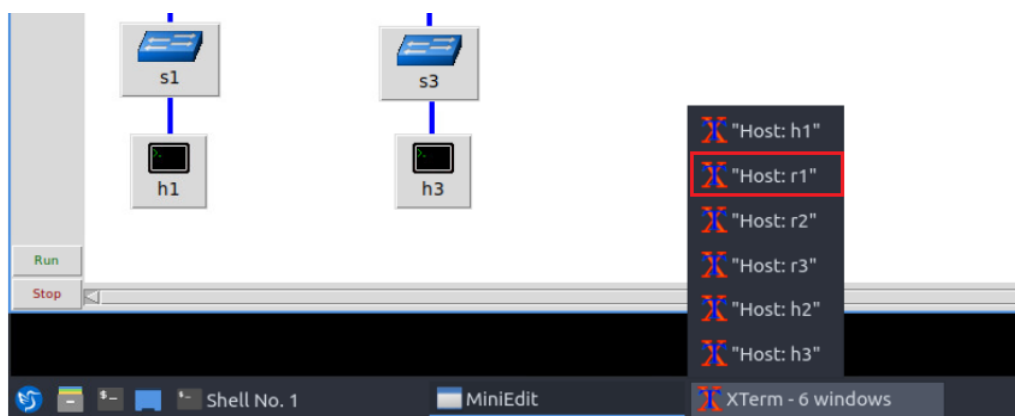
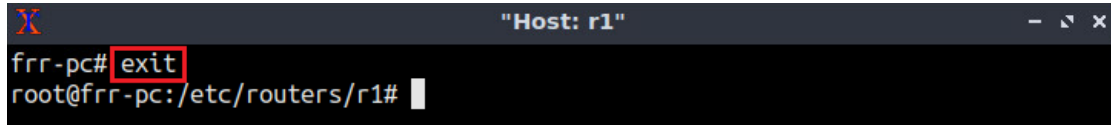


Figure 21. Opening router r1’s terminal.

**Step 2.** In router r1’s terminal, type the command below to exit the vtysh session.

```
exit
```



```

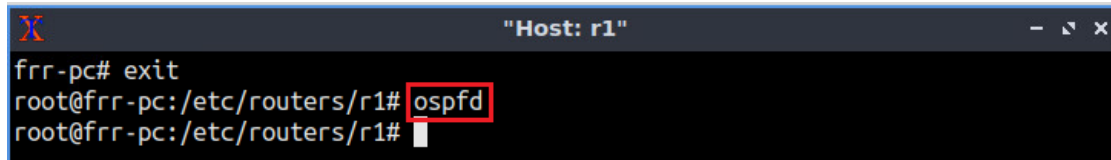
X "Host: r1"
frr-pc# exit
root@frr-pc:/etc/routers/r1#

```

Figure 22. Exiting the vtysh session.

**Step 3.** Enable and start the OSPF routing protocol by typing the following command:

```
ospfd
```



```

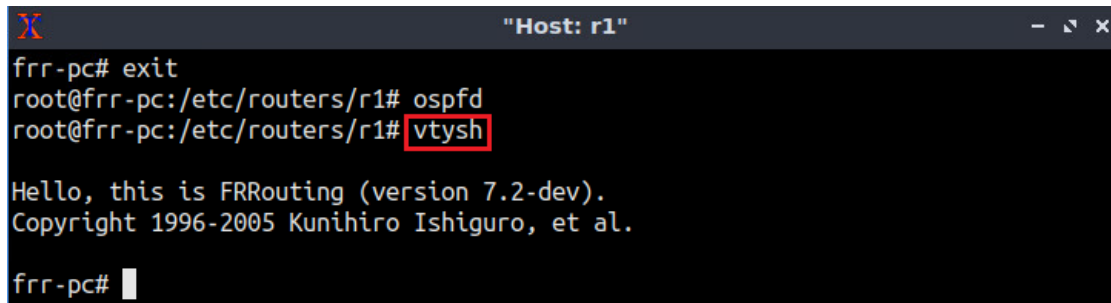
X "Host: r1"
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1#

```

Figure 23. Starting the OSPF daemon.

**Step 4.** In order to enter router r1's shell for FRR daemons, issue the following command:

```
vttysh
```



```

X "Host: r1"
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vttysh

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

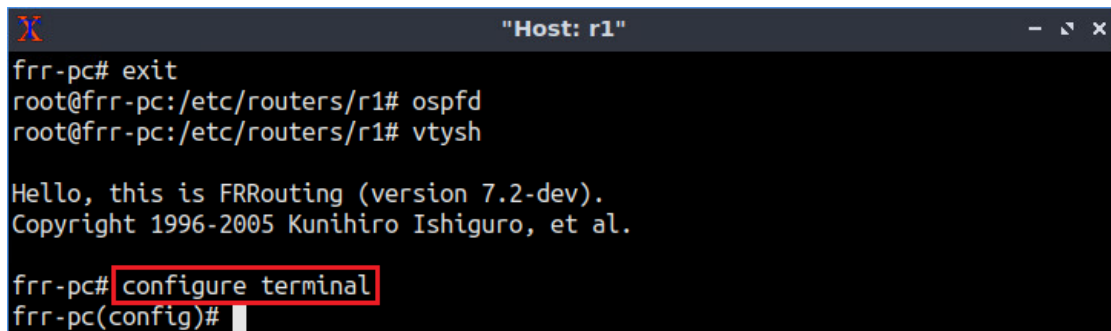
frr-pc#

```

Figure 24. Starting vtysh in router r1.

**Step 5.** At this point, you have entered router r1's shell for FRR daemons, to enter configuration mode in router r1, issue the following command:

```
configure terminal
```



```

X "Host: r1"
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vttysh

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

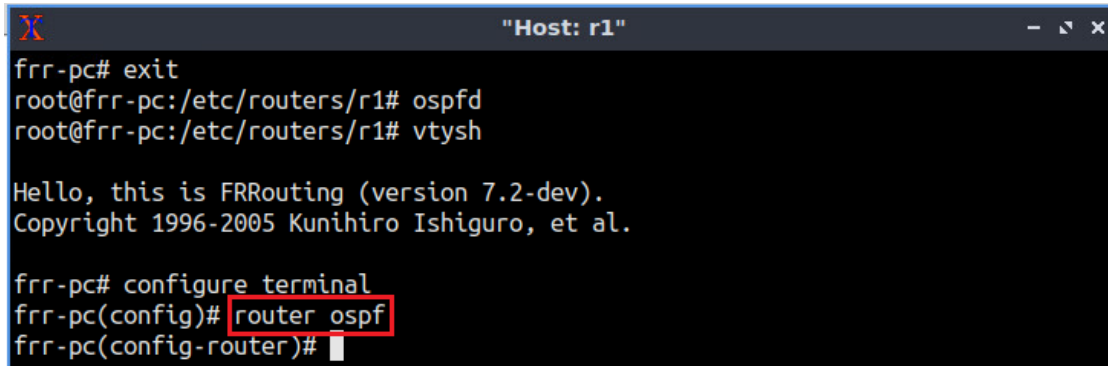
frr-pc# configure terminal
frr-pc(config)#

```

Figure 25. Enabling configuration mode in router r1.

**Step 6.** In order to configure OSPF type the command shown below. This command will enable OSPF configuration mode where you can assign the networks directly connected to router r1.

```
router ospf
```

A terminal window titled "Host: r1" showing the configuration process. The user starts at the frr-pc prompt, exits to root, runs 'ospfd' and 'vtysh', then enters 'configure terminal'. The prompt changes to 'frr-pc(config)#', and the command 'router ospf' is entered, resulting in the prompt 'frr-pc(config-router)#'. The command 'router ospf' is highlighted with a red box.

```
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

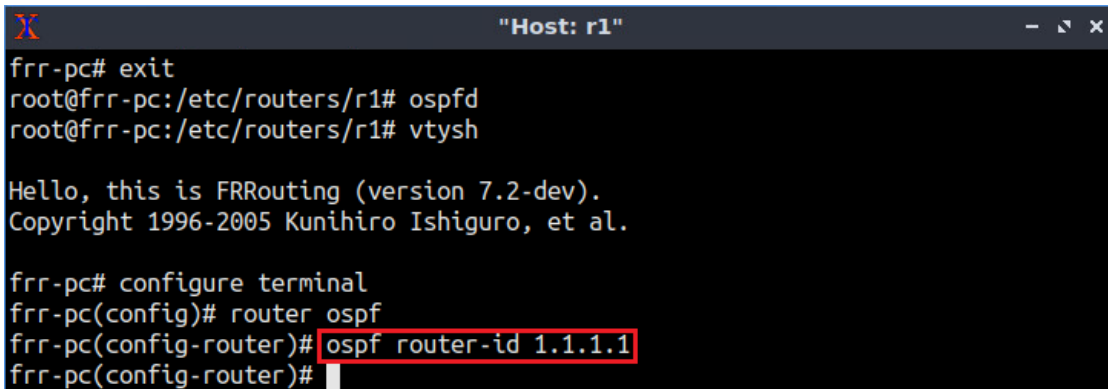
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)#
```

Figure 26. Configuring OSPF in router r1.

**Step 7.** Assign a router ID to router r1 by issuing the following command.

```
ospf router-id 1.1.1.1
```

A terminal window titled "Host: r1" showing the configuration process. The user starts at the frr-pc prompt, exits to root, runs 'ospfd' and 'vtysh', then enters 'configure terminal'. The prompt changes to 'frr-pc(config)#', and the command 'router ospf' is entered, resulting in the prompt 'frr-pc(config-router)#'. The command 'ospf router-id 1.1.1.1' is entered, resulting in the prompt 'frr-pc(config-router)#'. The command 'ospf router-id 1.1.1.1' is highlighted with a red box.

```
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 1.1.1.1
frr-pc(config-router)#
```

Figure 27. Assigning a router ID in router r1.

**Step 8.** Type the following command to enable the interface *r1-eth2* to participate in the OSPF process in area 0.

```
network 192.168.1.0/24 area 0
```

```
Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 1.1.1.1
frr-pc(config-router)# network 192.168.1.0/24 area 0
frr-pc(config-router)#
```

Figure 28. Enabling interface *r1-eth2* to participate in the OSPF process of area 0.

**Step 9.** Type the following command to enable the interface *r1-eth0* to participate in the OSPF process in area 0.

```
network 192.168.12.0/30 area 0
```

```
Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

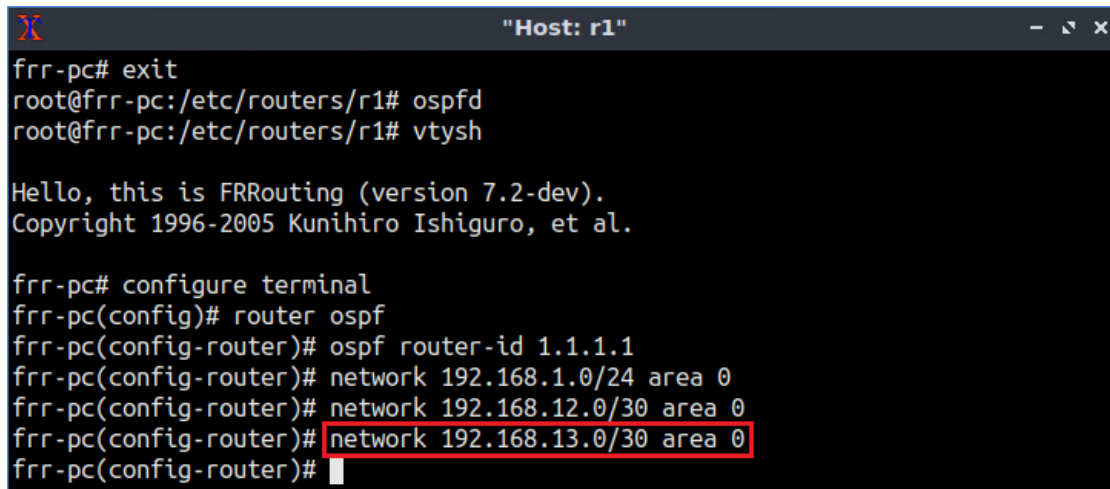
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 1.1.1.1
frr-pc(config-router)# network 192.168.1.0/24 area 0
frr-pc(config-router)# network 192.168.12.0/30 area 0
frr-pc(config-router)#
```

Figure 29. Enabling interface *r1-eth0* to participate in the OSPF process of area 0.

**Step 10.** Type the following command to enable the interface *r1-eth1* to participate in the OSPF process in area 0.

```
network 192.168.13.0/30 area 0
```

A terminal window titled "Host: r1" showing the configuration of OSPF on a router. The user starts in the root directory of the router's configuration files, enters 'ospfd' to start the daemon, and then 'vtysh' to enter the configuration mode. The user then enters 'configure terminal' to enter configuration mode. In configuration mode, the user enters 'router ospf', followed by 'ospf router-id 1.1.1.1'. Then, three 'network' commands are entered: 'network 192.168.1.0/24 area 0', 'network 192.168.12.0/30 area 0', and 'network 192.168.13.0/30 area 0'. The last command is highlighted with a red box. The terminal output shows the version of FRRouting (7.2-dev) and the copyright information.

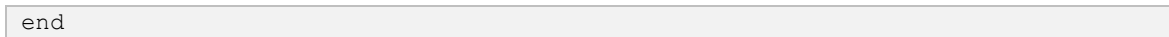
```
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

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

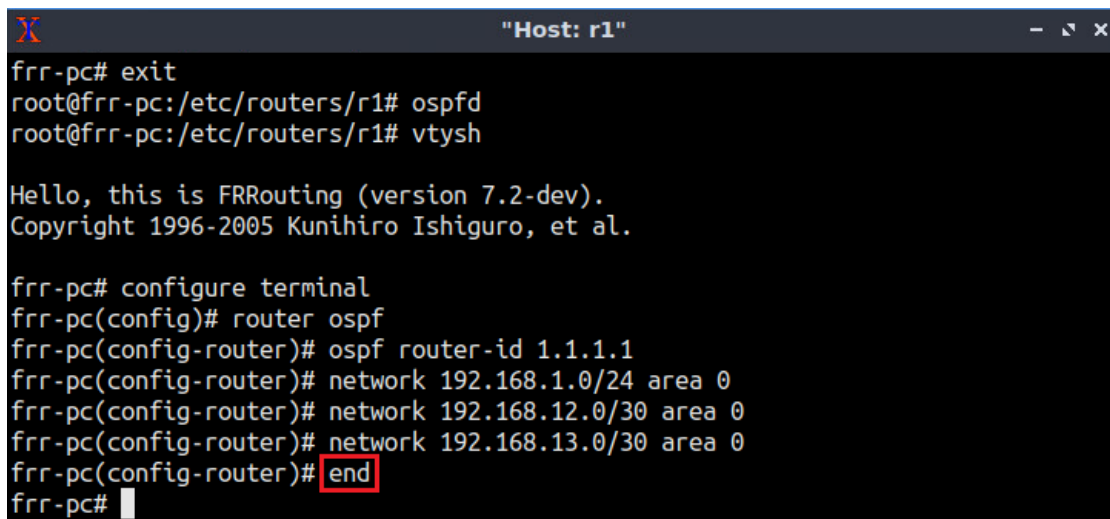
frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 1.1.1.1
frr-pc(config-router)# network 192.168.1.0/24 area 0
frr-pc(config-router)# network 192.168.12.0/30 area 0
frr-pc(config-router)# network 192.168.13.0/30 area 0
frr-pc(config-router)#
```

Figure 30. Enabling interface *r1-eth1* to participate in the OSPF process of area 0.

**Step 11.** Type the following command to exit configuration mode.

A terminal window showing the 'end' command being entered to exit configuration mode.

```
end
```

A terminal window titled "Host: r1" showing the completion of the OSPF configuration. The user enters 'end' to exit configuration mode, and the terminal returns to the root directory prompt. The 'end' command is highlighted with a red box.

```
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 1.1.1.1
frr-pc(config-router)# network 192.168.1.0/24 area 0
frr-pc(config-router)# network 192.168.12.0/30 area 0
frr-pc(config-router)# network 192.168.13.0/30 area 0
frr-pc(config-router)# end
frr-pc#
```

Figure 31. Exiting configuration mode.

**Step 12.** 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* terminal, issue the commands depicted below.

```

Host: r2
frr-pc# exit
root@frr-pc:/etc/routers/r2# ospfd
root@frr-pc:/etc/routers/r2# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 2.2.2.2
frr-pc(config-router)# network 192.168.2.0/24 area 0
frr-pc(config-router)# network 192.168.12.0/30 area 0
frr-pc(config-router)# network 192.168.23.0/30 area 0
frr-pc(config-router)# end
frr-pc#

```

Figure 32. Router r2 configuration summary.

**Step 13.** Router r3 is configured similarly to routers r1 and r2 but with different IP addresses (see Table 2). Those steps are summarized in the following figure. To proceed, in router r3 terminal, issue the commands depicted below.

```

Host: r3
frr-pc# exit
root@frr-pc:/etc/routers/r3# ospfd
root@frr-pc:/etc/routers/r3# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 3.3.3.3
frr-pc(config-router)# network 192.168.3.0/24 area 0
frr-pc(config-router)# network 192.168.13.0/30 area 0
frr-pc(config-router)# network 192.168.23.0/30 area 0
frr-pc(config-router)# end
frr-pc#

```

Figure 33. Router r3 configuration summary.

### 3.2 Verifying OSPF neighbors, routing information, and connectivity

**Step 1.** In router's r1 terminal, issue the following command to verify that router r1 lists the other routers in the network as neighbors.

```
show ip ospf neighbor
```

```

Host: r1
frr-pc# show ip ospf neighbor
Neighbor ID      Pri State           Dead Time Address           Interface           RXmtL RqstL DBsmL
2.2.2.2          1 Full/Backup     33.574s 192.168.12.2       r1-eth0:192.168.12.1 0      0      0
3.3.3.3          1 Full/Backup     38.753s 192.168.13.2       r1-eth1:192.168.13.1 0      0      0
frr-pc#

```

Figure 34. Displaying router r1's OSPF neighbors.

The output of the figure above shows that the neighbor ID is a unique ID for the router which, if not explicitly specified, will be the IP address of the highest interface of the router. The address field is the IP address of the corresponding neighbor. The interface field refers to the interface of the router through which the network is connected, with its corresponding IP address.

**Step 2.** To display all the routing information in the routing table of router r1, issue the following command:

```
show ip route
```

```

Host: r1
frr-pc# 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

O 192.168.2.0/24 [110/20] via 192.168.12.2, r1-eth1, 00:16:18
O>* 192.168.3.0/24 [110/20] via 192.168.13.2, r1-eth2, 00:07:48
O 192.168.12.0/30 [110/10] is directly connected, r1-eth1, 00:28:38
C>* 192.168.12.0/30 is directly connected, r1-eth1, 00:52:53
O 192.168.13.0/30 [110/10] is directly connected, r1-eth2, 00:27:31
C>* 192.168.13.0/30 is directly connected, r1-eth2, 00:52:53
O>* 192.168.23.0/30 [110/20] via 192.168.12.2, r1-eth1, 00:07:48
*
   via 192.168.13.2, r1-eth2, 00:07:48
frr-pc#

```

Figure 35. Displaying router r1's routing table.

Consider the above figure. The output displays the networks learned along with additional attributes, such as the routing protocol that discovered the network. For example, networks with the symbol `O` on the left are learned from OSPF, whereas those with symbol `C` are directly connected to the device. Furthermore, the symbol `>` on the left of the displayed network indicates that this entry is selected to reach the network. The symbol `*` indicates that the route is inserted in Forwarding Information Base (FIB) of the router. In this network, the routes 192.168.2.0/24, 192.168.3.0/24, and 192.168.23.0/30 are discovered and reached by the OSPF routing protocol (`O>*`).

**Step 3.** Issue the following command to show the FIB table.



```
show ip fib
```

```

Host: r1
frr-pc# show ip fib
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-eth2, 00:12:44
O>* 192.168.2.0/24 [110/20] via 192.168.12.2, r1-eth0, 00:09:20
O>* 192.168.3.0/24 [110/20] via 192.168.13.2, r1-eth1, 00:08:25
C>* 192.168.12.0/30 is directly connected, r1-eth0, 00:12:44
C>* 192.168.13.0/30 is directly connected, r1-eth1, 00:12:44
O>* 192.168.23.0/30 [110/20] via 192.168.12.2, r1-eth0, 00:08:25
   *                               via 192.168.13.2, r1-eth1, 00:08:25
frr-pc#

```

Figure 36. Displaying router r1's FIB table.

**Step 5.** Follow from step 1 through 3 in routers r2 and r3 to verify their OSPF neighbors and the existence of all the networks in their routing table.

**Step 6.** In the host h1 terminal, issue the following command to test the connectivity between host h1 and host h2.

```
ping 192.168.2.10 -c 4
```

```

Host: h1
root@frr-pc:~# ping 192.168.2.10 -c 4
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=62 time=0.800 ms
64 bytes from 192.168.2.10: icmp_seq=2 ttl=62 time=0.096 ms
64 bytes from 192.168.2.10: icmp_seq=3 ttl=62 time=0.095 ms
64 bytes from 192.168.2.10: icmp_seq=4 ttl=62 time=0.092 ms

--- 192.168.2.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 80ms
rtt min/avg/max/mdev = 0.092/0.270/0.800/0.306 ms
root@frr-pc:~#

```

Figure 37. Performing a connectivity test from host h2 from h1.

**Step 7.** Test the connectivity between hosts h1 and h3 by issuing the following command:

```
ping 192.168.3.10 -c 4
```

```

Host: h1
root@frr-pc:~# ping 192.168.3.10 -c 4
PING 192.168.3.10 (192.168.3.10) 56(84) bytes of data.
64 bytes from 192.168.3.10: icmp_seq=1 ttl=62 time=0.669 ms
64 bytes from 192.168.3.10: icmp_seq=2 ttl=62 time=0.096 ms
64 bytes from 192.168.3.10: icmp_seq=3 ttl=62 time=0.088 ms
64 bytes from 192.168.3.10: icmp_seq=4 ttl=62 time=0.096 ms

--- 192.168.3.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 78ms
rtt min/avg/max/mdev = 0.088/0.237/0.669/0.249 ms
root@frr-pc:~#

```

Figure 38. Performing a connectivity test from host h3 from h1.

**Step 6.** Similarly, repeat steps 5 and 6 to test the connectivity between any two components in the network.

#### 4 Configuring OSPF passive interfaces

In this section, you will configure the passive interface OSPF feature on the interfaces of the routers connecting them to their hosts, i.e., the interfaces *r1-eth1*, *r3-eth0* and *r2-eth2* on each router. Thus, reducing the traffic on the LANs as they do not need to receive dynamic routing protocol communication.

**Step 1.** Re-enter router *r1*'s terminal. Interface *r1-eth2* is activated to send and receive OSPF routing updates. To verify the OSPF status of this interface, type the following command.

```
show ip ospf interface r1-eth2
```

```

Host: r1
frr-pc# show ip ospf interface r1-eth2
r1-eth2 is up
  ifindex 4, MTU 1500 bytes, BW 10000 Mbit <UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 192.168.1.1/24, Broadcast 192.168.1.255, Area 0.0.0.0
  MTU mismatch detection: enabled
  Router ID 1.1.1.1, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State DR, Priority 1
  No backup designated router on this network
  Multicast group memberships: OSPFALLRouters OSPFDesignatedRouters
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
  Hello due in 7.489s
  Neighbor Count is 0, Adjacent neighbor count is 0

frr-pc#

```

Figure 39. OSPF information on interface *r1-eth2*.

Consider the figure above. The interface *r1-eth2* exchanges hello packets. These messages indicate that this interface is alive and participating in the OSPF routing updates. The counter indicates how long it will take to send the next hello packet, in this case 7.489 seconds.

**Step 2.** To enter configuration mode in router *r1*, issue the following command:

```
configure terminal
```

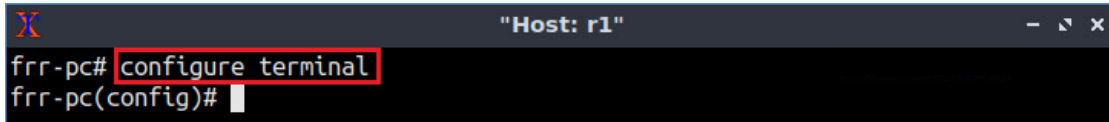
A terminal window titled "Host: r1" showing the command sequence: frr-pc# configure terminal (highlighted in red), frr-pc(config)#, and frr-pc(config)#. The cursor is at the end of the last line.

Figure 40. Starting vtysh in router *r1*.

**Step 3.** In order to configure OSPF issue the following command.

```
router ospf
```

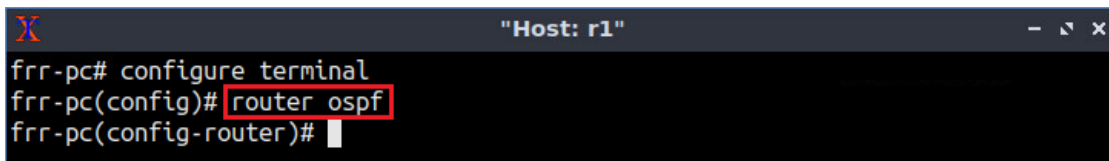
A terminal window titled "Host: r1" showing the command sequence: frr-pc# configure terminal, frr-pc(config)# router ospf (highlighted in red), and frr-pc(config-router)#. The cursor is at the end of the last line.

Figure 41. Configuring OSPF in router *r1*.

**Step 4.** To prevent routing updates from being sent through the interface *r1-eth0*, issue the following command:

```
passive-interface r1-eth2
```

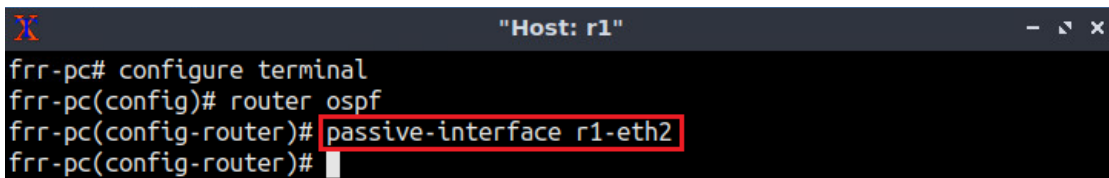
A terminal window titled "Host: r1" showing the command sequence: frr-pc# configure terminal, frr-pc(config)# router ospf, frr-pc(config-router)# passive-interface r1-eth2 (highlighted in red), and frr-pc(config-router)#. The cursor is at the end of the last line.

Figure 42. Configuring interface *r1-eth2* as passive.

**Step 5.** To exit configuration mode, issue the following command:

```
end
```

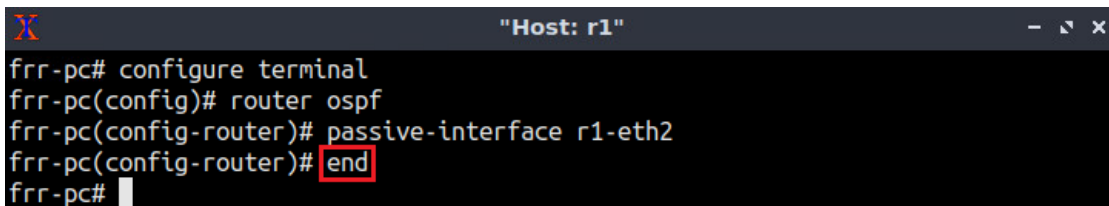
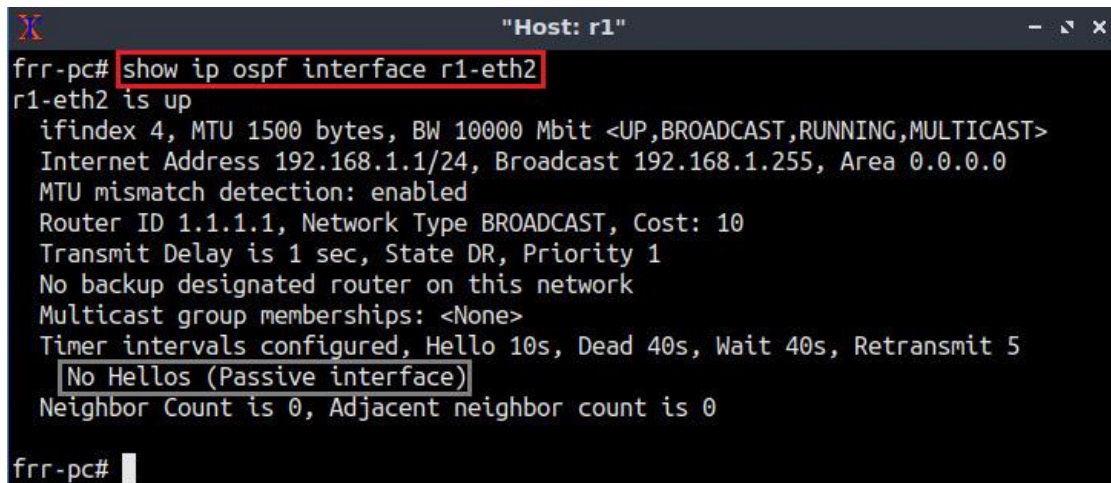
A terminal window titled "Host: r1" showing the command sequence: frr-pc# configure terminal, frr-pc(config)# router ospf, frr-pc(config-router)# passive-interface r1-eth2, frr-pc(config-router)# end (highlighted in red), and frr-pc#. The cursor is at the end of the last line.

Figure 43. Exiting configuration mode.

**Step 6.** To verify that the interface *r1-eth2* is now passive, issue the following command.

```
show ip ospf interface r1-eth2
```



```

Host: r1
frr-pc# show ip ospf interface r1-eth2
r1-eth2 is up
  ifindex 4, MTU 1500 bytes, BW 10000 Mbit <UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 192.168.1.1/24, Broadcast 192.168.1.255, Area 0.0.0.0
  MTU mismatch detection: enabled
  Router ID 1.1.1.1, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State DR, Priority 1
  No backup designated router on this network
  Multicast group memberships: <None>
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
  No Hellos (Passive interface)
  Neighbor Count is 0, Adjacent neighbor count is 0

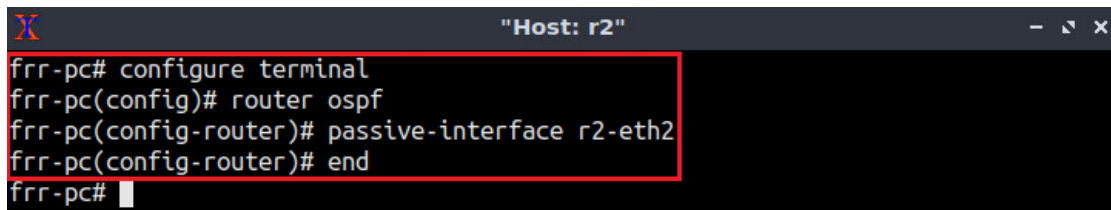
frr-pc#

```

Figure 44. OSPF information in interface *r1-eth2*.

Consider the figure above. Notice that the interface *r1-eth2* no longer exchanges hello packets and is configured as passive, which means it is not receiving the OSPF routing updates.

**Step 7.** Similarly, in router *r2* configure *r2-eth2* as a passive interface by repeating steps 2 through 6 with the appropriate addresses. The commands are summarized below.



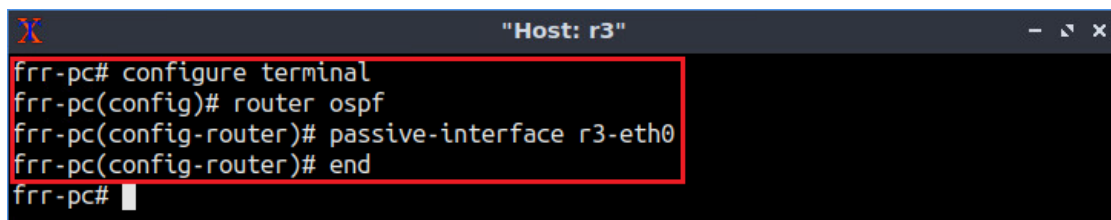
```

Host: r2
frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# passive-interface r2-eth2
frr-pc(config-router)# end
frr-pc#

```

Figure 45. Configuring *r2-eth2* as a passive interface.

**Step 8.** Similarly, in router *r3* configure *r3-eth0* as a passive interface by repeating steps 2 through 6 with the appropriate addresses. The necessary commands are summarized below.



```

Host: r3
frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# passive-interface r3-eth0
frr-pc(config-router)# end
frr-pc#

```

Figure 46. Configuring *r3-eth0* as a passive interface.

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

## References

1. Moy, J. "RFC 1247: OSPF version 2." 1991.
2. Moy, J. "RFC 2328: OSPF version 2." 1998.

3. Coltun, R., Ferguson, D., Moy, J., Lindem, A. "RFC 5340, OSPF for IPv6." 2008.
4. Peterson, L. "Computer networks: a systems approach." 6th Edition, 2021.
5. Tanenbaum, D. Wetherall. "Computer networks." 5th Edition. 2012.
6. J. Kurose, K. Ross. "Computer networking, a top-down approach." 7th Edition. 2017.
7. Teare, D., Vachon, B., Graziani, R. "Implementing Cisco IP routing (ROUTE) foundation learning guide: (CCNP ROUTE 300-101)." 2014.
8. Linux foundation collaborative projects. "FRR routing documentation." 2017 [Online] Available: <http://docs.frrouting.org/en/latest/>.



UNIVERSITY OF  
**SOUTH CAROLINA**

## OPEN SHORTEST PATH FIRST

### Lab 4: Configuring Multi-Area OSPFv2

Document Version: **10-22-2021**



Award 1829698

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

## Contents

Overview .....	3
Objectives.....	3
Lab settings .....	3
Lab roadmap .....	3
1 Introduction .....	3
1.1 Single-area OSPF .....	3
1.2 Multi-area OSPF .....	4
1.3 OSPF router ID.....	5
2 Lab topology.....	6
2.1 Lab settings.....	7
2.2 Loading the topology.....	7
2.3 Load the zebra daemon and verify the configuration .....	10
3 Configuring multi-area OSPF.....	14
3.1 Configuring multi-area OSPF in routers r1, r2, and r3 .....	14
3.2 Verifying OSPF neighbors, routing information, and connectivity .....	19
4 Configuring OSPF passive interfaces.....	21
4.1 Configuring passive interfaces in routers r1, r2, and r3.....	21
References .....	23

## Overview

This lab explains the concept of multi-area Open Shortest Path First (OSPF) and provides a scenario to show its usage and effectiveness. Moreover, the user will configure multi-area OSPF on a simple topology, assign each router a unique ID, configure passive interfaces, and test the connectivity between hosts in different networks.

## Objectives

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

1. Explain the concept of multi-area OSPF.
2. Understand when and why to use multi-area OSPF.
3. Configure multi-area OSPF.
4. Understand and configure OSPF router IDs.
5. Understand and configure OSPF passive interfaces.

## Lab settings

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

Table 1. Credentials to access the Client 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 multi-area OSPF.
4. Section 4: Configuring OSPF passive interfaces.

### 1 Introduction

#### 1.1 Single-area OSPF



OSPF is a widely supported Interior Gateway Protocol (IGP) in the family of routing protocols, meaning that it is designed to be used within a single Autonomous System (AS). It is a link-state protocol which sends information about directly connected links to all routers in the network, instead of sending the entire routing table as in the case of distance vector protocol<sup>1</sup>.

An OSPF area is a logical collection of OSPF networks, routers, and links that has the same Area-ID. A router within an OSPF area maintains a topological database for the area to which it belongs<sup>2</sup>. Configuring only one OSPF area (single-area OSPF) within networks, becomes infeasible in large networks due to several reasons listed below<sup>1</sup>:

- The link state database of every router can become overwhelmingly large since it will be filled with information about each router and link within the entire network.
- The routing table of every router will be large as well, which can delay routing decisions and slow down the network.
- The entire network will be flooded by updates, causing congestion, and requiring the OSPF algorithm to be run in each router after every update.

Consider Figure 1. The figure displays single-area OSPF within a network of 5 routers. Each router has a full view of the whole topology in the network. Additionally, the Link-State Advertisements (LSAs) used by OSPF are flooded over the entire network.

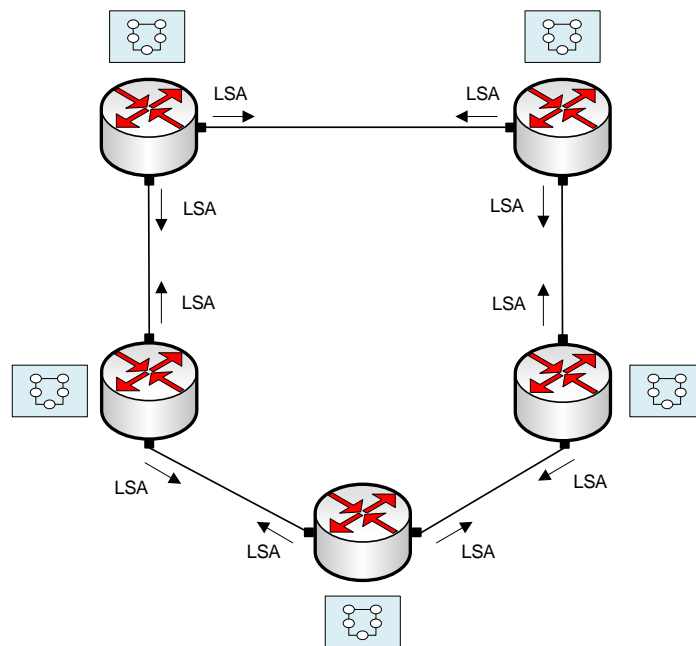


Figure 1. Single-area OSPF within a large network.

## 1.2 Multi-area OSPF

To make OSPF more efficient and scalable, OSPF supports hierarchical routing using the concept of area. An OSPF area is a group of routers that share the same link-state information in their link-state databases. When a large OSPF area is divided into smaller areas, it is called multi-area OSPF. Multi-area OSPF is useful in larger network deployments since it reduces the size of the routing table, minimizes the link-state update overhead, and shrinks the frequency of OSPF algorithm calculations. When multi-area OSPF is running, routers only need to know about the links in their own area<sup>2</sup>.

Consider Figure 2. The routers run OSPF, and the topology is divided into three areas, area 0, area 1 and area 2. Area 0 is also called the backbone area since all other areas are connected to it directly, and it is the zone where the communication between the areas occurs<sup>3</sup>.

Essentially, routers within an area receive OSPF advertisements from routers within their area only. For instance, a link failure in area 1 won't be propagated to area 2. Routers that connect to multiple areas are referred to as Area Border Routers (ABR), i.e., routers r4 and r5 are ABRs in Figure 2. ABRs establish a connection between the backbone area and their area.

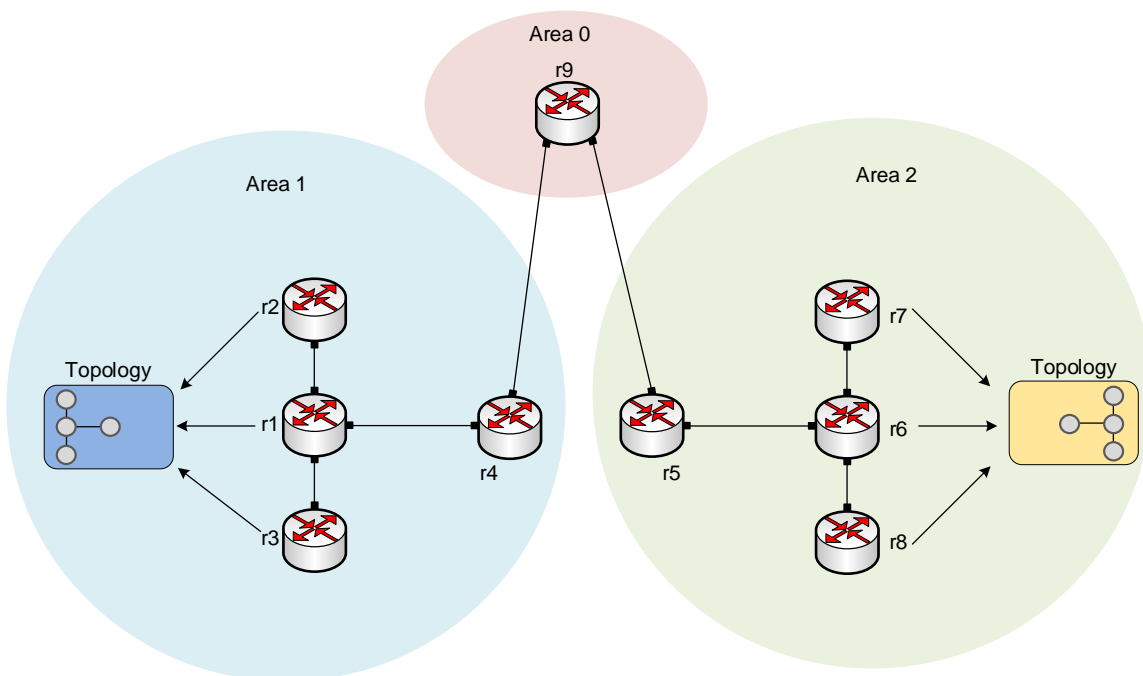


Figure 2. Multi-area OSPF topology.

### 1.3 OSPF router ID

Each OSPF router chooses a router ID that is unique in the network. This ID identifies each router in the topology. If there are duplicate router IDs, then the network will have reachability issues.

To assign a router ID, OSPF uses the following criteria:

1. Manually assign the router ID.
2. Highest IP address on a loopback interface.
3. Highest IP address on a non-loopback interface.

## 2 Lab topology

Consider Figure 3. The topology consists of five networks connected to its respective routers. These networks are distributed among three OSPF areas. For example, area 2 contains networks 192.168.2.0/24, 192.168.23.0/30 and 192.168.3.0/24. The communication between networks is established via multi-area OSPF configured in all routers.

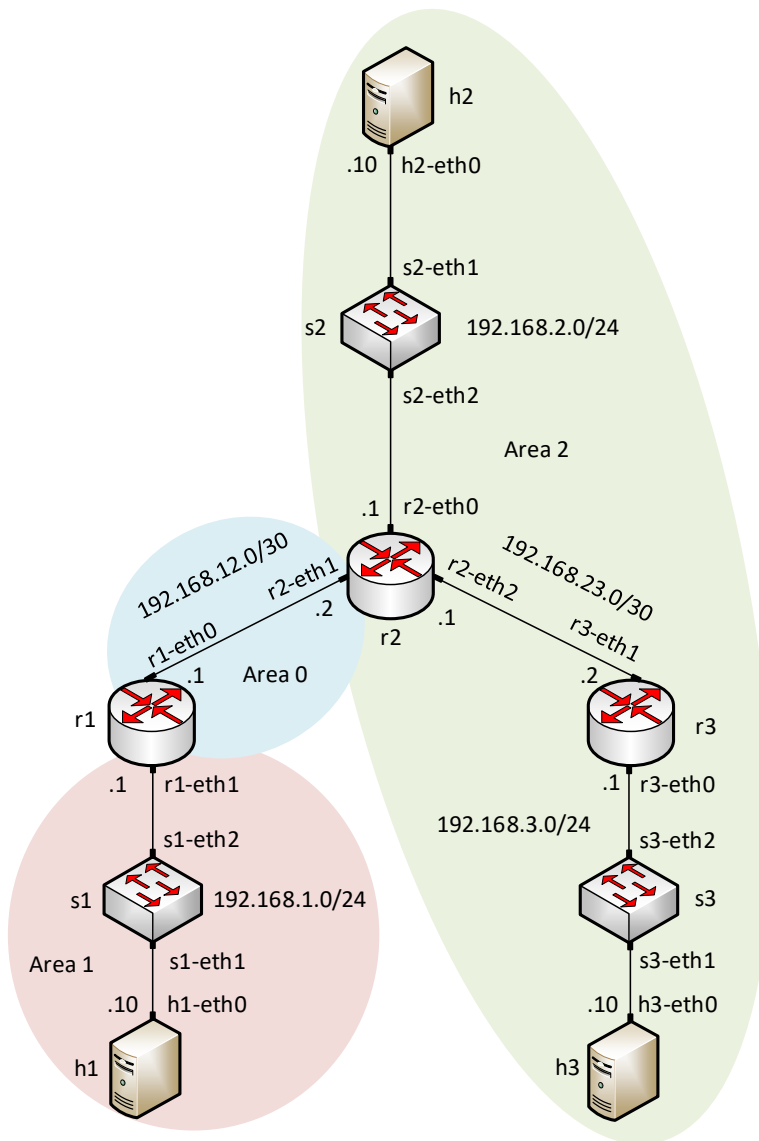


Figure 3. Topology.

## 2.1 Lab settings

Table 2 contains information about the devices and their corresponding interfaces, IPv4 addresses, subnet, and default gateway.

Table 2. Topology Information.

Device	Interface	IPv4 Address	Subnet	Default Gateway
Router 1 (r1)	r1-eth0	192.168.12.1	/30	N/A
	r1-eth1	192.168.1.1	/24	N/A
Router 2 (r2)	r2-eth0	192.168.2.1	/24	N/A
	r2-eth1	192.168.12.2	/30	N/A
	r2-eth2	192.168.23.1	/30	N/A
Router 3 (r3)	r3-eth0	192.168.3.1	/24	N/A
	r3-eth1	192.168.23.2	/30	N/A
Host 1 (h1)	h1-eth0	192.168.1.10	/24	192.168.1.1
Host 2 (h2)	h2-eth0	192.168.2.10	/24	192.168.2.1
Host 3 (h3)	h3-eth0	192.168.3.10	/24	192.168.3.1

## 2.2 Loading the topology

**Step 1.** Launch MiniEdit by double clicking on the shortcut located on the desktop. When prompted for a password, type `password`.



Figure 4. MiniEdit shortcut.

**Step 2.** In MiniEdit's menu bar, click on *File* then *open* to load the lab topology. Locate the *lab4.mn* topology file in the default directory, */home/frr/OSPF\_Labs/lab4* and click on *Open*.

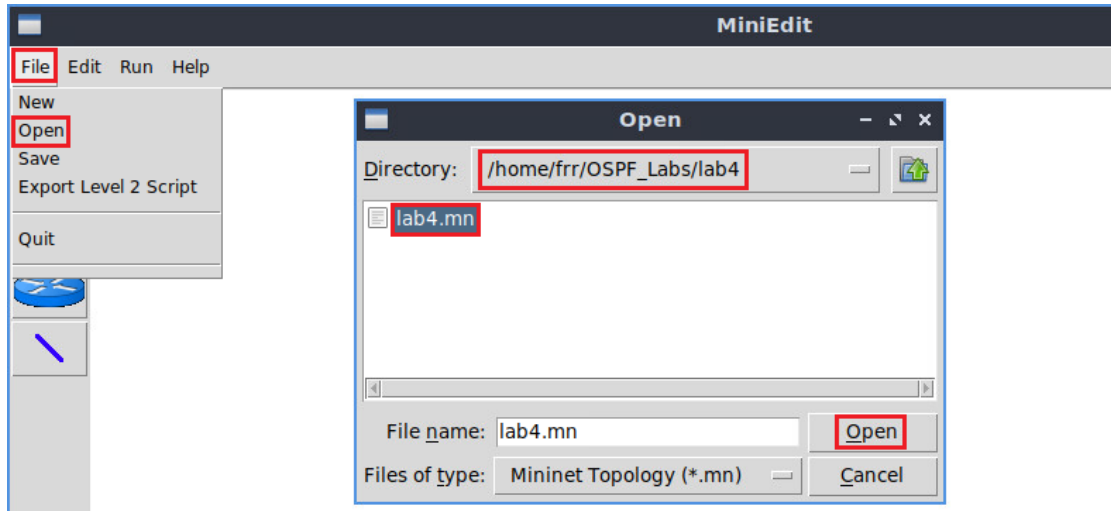


Figure 5. MiniEdit's open dialog.

**Step 3.** Click on the *Run* button located on the lower left-hand side to run the emulation.

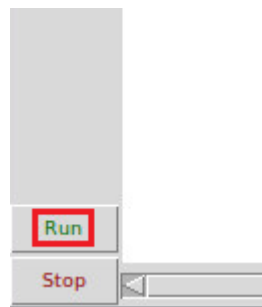


Figure 6. Starting the emulation.

At this point, the topology is loaded with all the required network components. Next, you will execute a script that will load the configure the interfaces of the routers.

Automating the configuration of interfaces with the script is intended to facilitate the lab experience and to focus on the OSPF experiment.

**Step 4.** Open the Linux terminal by clicking on the icon located in the taskbar.

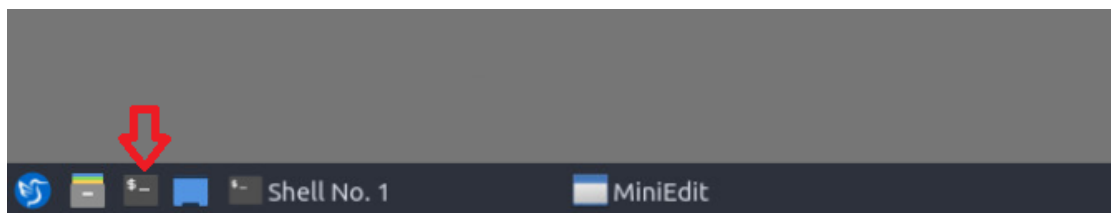


Figure 7. Opening Linux terminal.

**Step 5.** Issue the command shown below to set the lab environment. This command configures the IP addresses of the routers' interface according to Table 2. If a password is required, type `password`.

```
sudo set_env lab4
```

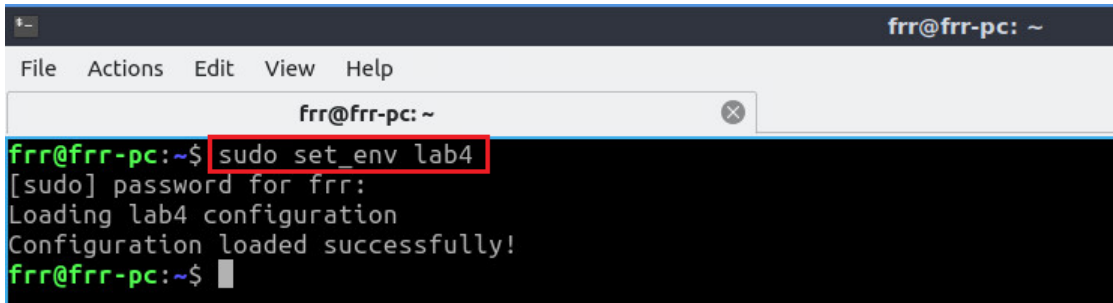


Figure 8. Setting the lab environment.

**Step 6.** Type the following command to exit the Linux terminal.

```
exit
```

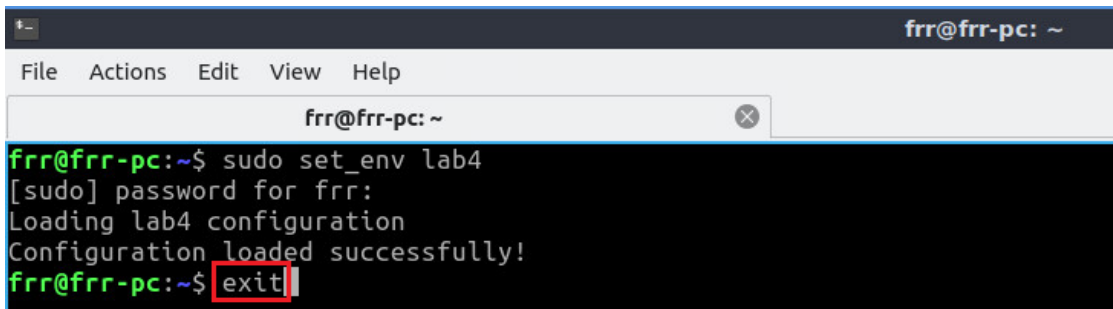


Figure 9. Exiting the Linux terminal.

**Step 7.** Click on Mininet's terminal, i.e., the one launched when MiniEdit was started.

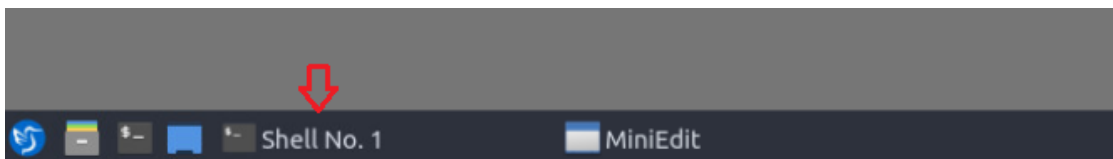


Figure 10. Opening Mininet's terminal.

**Step 8.** Issue the following command to display the interface names and connections.

```
links
```

```

mininet> links
h3-eth0<->s3-eth1 (OK OK)
s3-eth2<->r3-eth0 (OK OK)
h2-eth0<->s2-eth1 (OK OK)
s2-eth2<->r2-eth0 (OK OK)
r1-eth0<->r2-eth1 (OK OK)
r2-eth2<->r3-eth1 (OK OK)
h1-eth0<->s1-eth1 (OK OK)
s1-eth2<->r1-eth1 (OK OK)
mininet>
    
```

Figure 11. Displaying network interfaces.

In the figure above, the link displayed within the gray box indicates that interface *eth2* of switch *s2* connects to interface *eth0* of router *r2* (i.e., *s2-eth2<->r2-eth0*).

### 2.3 Load the zebra daemon and verify the configuration

In this section, you will verify that the IP addresses listed in Table 2 and inspect the routing table of the routers.

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

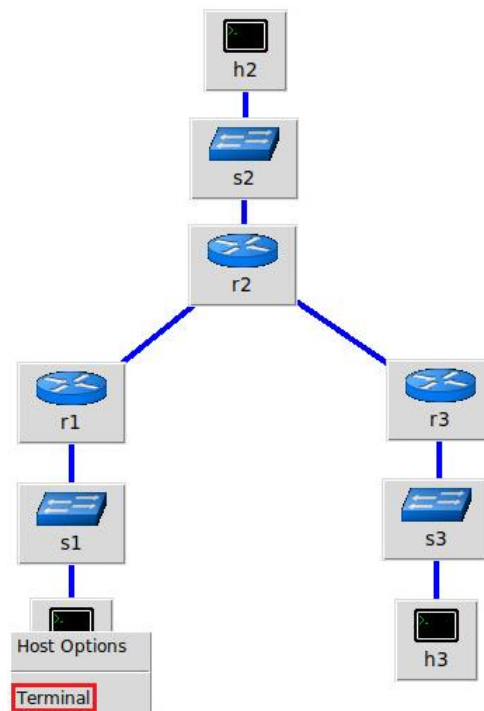
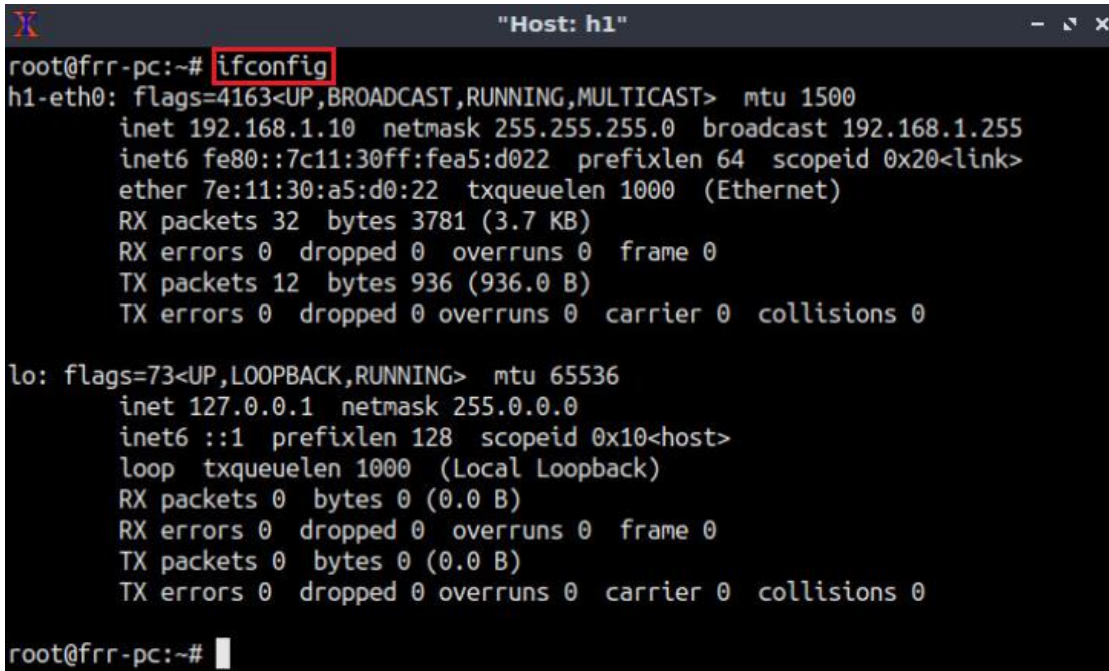


Figure 12. Opening a terminal on host *h1*.

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

```
ifconfig
```



```
root@frr-pc:~# ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.10 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::7c11:30ff:fea5:d022 prefixlen 64 scopeid 0x20<link>
    ether 7e:11:30:a5:d0:22 txqueuelen 1000 (Ethernet)
    RX packets 32 bytes 3781 (3.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 12 bytes 936 (936.0 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@frr-pc:~#
```

Figure 13. Displaying host h1 interfaces' configuration.

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

```
route
```



```

Host: h1
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
  inet 192.168.1.10 netmask 255.255.255.0 broadcast 192.168.1.255
  inet6 fe80::7c11:30ff:fea5:d022 prefixlen 64 scopeid 0x20<link>
  ether 7e:11:30:a5:d0:22 txqueuelen 1000 (Ethernet)
  RX packets 32 bytes 3781 (3.7 KB)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 12 bytes 936 (936.0 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@frr-pc:~# 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@frr-pc:~#

```

Figure 14. Displaying the default gateway of host h1.

**Step 4.** To verify the hosts h2 and h3 default route, proceed by repeating from step 1 to step 3. Similar results should be observed.

**Step 5.** You will validate that the router interfaces are configured according to Table 2. To proceed, hold right-click on router r1 and select *Terminal*.

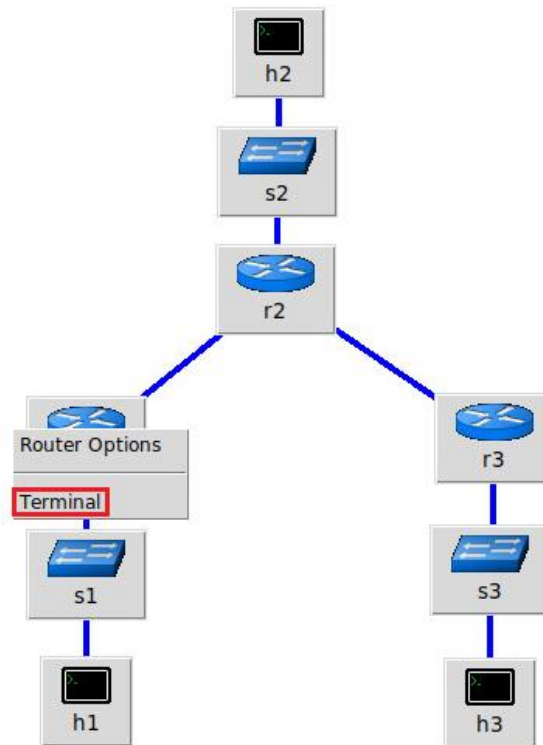


Figure 15. Opening a terminal on router r1.

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

```
zebra
```

A terminal window titled "Host: r1" shows the command prompt root@frr-pc:/etc/routers/r1#. The word "zebra" is entered and highlighted with a red box. The prompt returns to root@frr-pc:/etc/routers/r1#.

Figure 16. Starting zebra daemon.

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

```
vtysh
```

A terminal window titled "Host: r1" shows the command prompt root@frr-pc:/etc/routers/r1#. The word "zebra" is entered, followed by "vtysh", which is highlighted with a red box. The prompt returns to root@frr-pc:/etc/routers/r1#. Below this, the text "Hello, this is FRRouting (version 7.2-dev). Copyright 1996-2005 Kunihiro Ishiguro, et al." is displayed. The prompt then changes to frr-pc#.

Figure 17. Starting vtysh on router r1.

**Step 8.** Type the following command on router r1 terminal to verify the routing table. It will list all the directly connected networks. Notice that 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
```

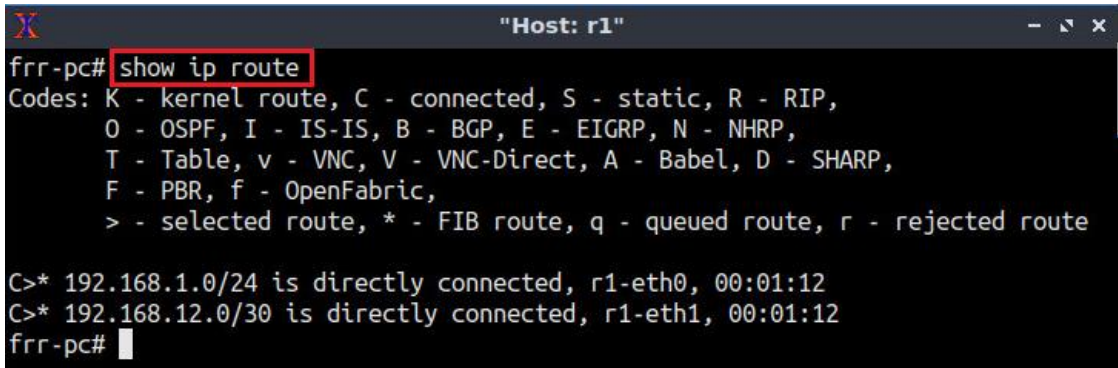


Figure 18. Displaying routing table of router r1.

**Step 9.** To verify the routing table of routers r2 and r3, proceed similarly by repeating from step 5 to step 8 on the router terminal.

### 3 Configuring multi-area OSPF

In this section, you will configure routers r1, r2, and r3 to participate in the OSPFv2 process. The topology in this lab is divided into three areas as shown in Figure 3. By dissecting a network into several areas, routers will not have a detailed information about the network topology outside of their area. This allows for greater scalability and control over the network.

#### 3.1 Configuring multi-area OSPF in routers r1, r2, and r3

**Step 1.** Navigate to router r1’s terminal by hovering over the open windows and selecting router r1.

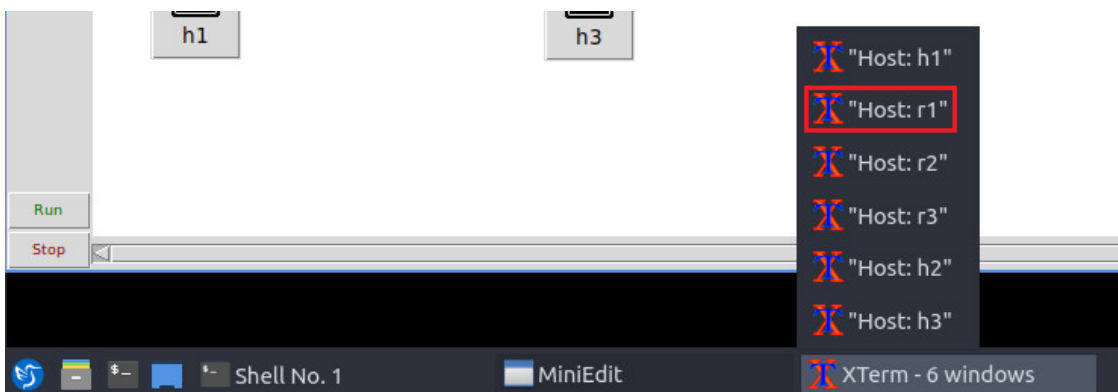


Figure 19. Opening router r1’s terminal.

**Step 2.** In router r1's terminal, type the command below to exit the vtysh session.

```
exit
```

```

X "Host: r1"
frr-pc# exit
root@frr-pc:/etc/routers/r1#
    
```

Figure 20. Exiting the vtysh session.

**Step 3.** Enable and start the OSPF routing protocol by typing the following command:

```
ospfd
```

```

X "Host: r1"
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1#
    
```

Figure 21. Starting the OSPF daemon.

**Step 4.** In order to enter router r1's shell for FRR daemons, issue the following command:

```
vtysh
```

```

X "Host: r1"
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

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

frr-pc#
    
```

Figure 22. Starting vtysh in router r1.

**Step 5.** At this point, you have entered router r1's shell for FRR daemons, to enter configuration mode in router r1, issue the following command:

```
configure terminal
```

```

X "Host: r1"
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

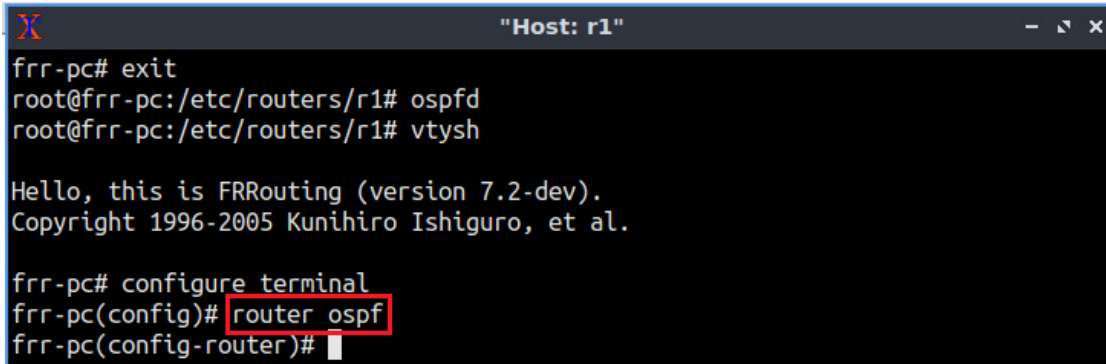
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)#
    
```

Figure 23. Enabling configuration mode in router r1.

**Step 6.** In order to configure OSPF type the command shown below. This command will enable OSPF configuration mode where you can assign the networks directly connected to router r1.

```
router ospf
```



```
Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

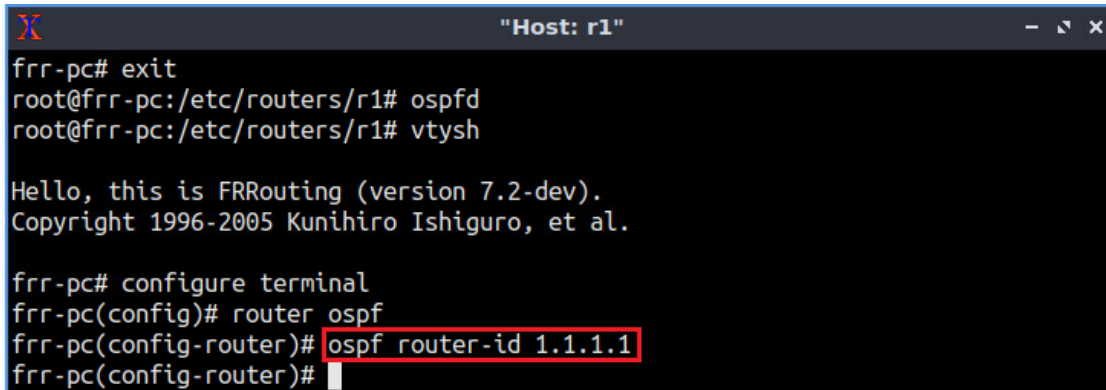
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)#
```

Figure 24. Configuring OSPF in router r1.

**Step 7.** To establish router r1 ID, issue the following command:

```
ospf router-id 1.1.1.1
```



```
Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 1.1.1.1
frr-pc(config-router)#
```

Figure 25. Assigning the OSPF router ID of 1.1.1.1 to router r1.

**Step 8.** Type the following command to enable the interface *r1-eth1* to participate in the OSPF process in area 1.

```
network 192.168.1.0/24 area 1
```

```
Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 1.1.1.1
frr-pc(config-router)# network 192.168.1.0/24 area 1
frr-pc(config-router)#
```

Figure 26. Enabling interface *r1-eth1* to participate in the OSPF process of area 1.

Consider the above figure. The command enables the interface *r1-eth0* to participate in the OSPF routing process of area 1 since this interface matches 192.168.1.0/24.

**Step 9.** Similarly, issue the following command, so that the interface *r1-eth0* can participate in the routing process of area 0.

```
network 192.168.12.0/30 area 0
```

```
Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 1.1.1.1
frr-pc(config-router)# network 192.168.1.0/24 area 1
frr-pc(config-router)# network 192.168.12.0/30 area 0
frr-pc(config-router)#
```

Figure 27. Enabling interface *r1-eth0* to participate in the OSPF process of area 0.

**Step 10.** Type the following command to exit from configuration mode.

```
end
```

```

Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 1.1.1.1
frr-pc(config-router)# network 192.168.1.0/24 area 1
frr-pc(config-router)# network 192.168.12.0/30 area 0
frr-pc(config-router)# end
frr-pc#

```

Figure 28. Exiting configuration mode.

**Step 11.** Follow steps 2 through 10 to configure OSPF in router r2. The necessary steps are summarized in the figure below.

```

Host: r2
frr-pc# exit
root@frr-pc:/etc/routers/r2# ospfd
root@frr-pc:/etc/routers/r2# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 2.2.2.2
frr-pc(config-router)# network 192.168.2.0/24 area 2
frr-pc(config-router)# network 192.168.12.0/30 area 0
frr-pc(config-router)# network 192.168.23.0/30 area 2
frr-pc(config-router)# end
frr-pc#

```

Figure 29. Configuring OSPF in router r2.

Consider the figure above. The commands entered will allow interfaces *r2-eth0* and *r2-eth2* to participate in the OSPF routing process of area 2 since their IPs match 192.168.2.0/24 and 192.168.23.0/30, respectively. Similarly, interface *r2-eth1* will participate in the OSPF routing process of area 0 since its IP matches 192.168.12.0/30.

**Step 12.** Follow steps 2 through 10 to configure OSPF in router r3. The necessary steps are summarized in the following figure.

```

Host: r3
frr-pc# exit
root@frr-pc:/etc/routers/r3# ospfd
root@frr-pc:/etc/routers/r3# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 3.3.3.3
frr-pc(config-router)# network 192.168.3.0/24 area 2
frr-pc(config-router)# network 192.168.23.0/24 area 2
frr-pc(config-router)# end
frr-pc#

```

Figure 30. Configuring OSPF in router r3.

Consider the figure above. The commands entered will allow interfaces *r3-eth0* and *r3-eth1* to participate in the OSPF routing process of area 2 since their IPs match 192.168.2.0/24 and 192.168.23.0/30, respectively.

### 3.2 Verifying OSPF neighbors, routing information, and connectivity

**Step 1.** In router's r1 terminal, issue the following command to verify that router r1 shows that router r2 is its neighbor.

```
show ip ospf neighbor
```

```

Host: r1
frr-pc# show ip ospf neighbor

Neighbor ID    Pri  State           Dead Time  Address          Interface          RXmtL  RqstL  DBsmL
2.2.2.2        1    Full/Backup     33.646s   192.168.12.2    r1-eth0:192.168.12.1  0      0      0
frr-pc#

```

Figure 31. Displaying router r1's OSPF neighbors

**Step 2.** To display all the routing information in the routing table of router r1, issue the following command:

```
show ip route
```



```

Host: r1
frr-pc# 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

O 192.168.1.0/24 [110/10] is directly connected, r1-eth1, 00:03:26
C>* 192.168.1.0/24 is directly connected, r1-eth1, 00:03:48
O>* 192.168.2.0/24 [110/20] via 192.168.12.2, r1-eth0, 00:02:21
O>* 192.168.3.0/24 [110/30] via 192.168.12.2, r1-eth0, 00:01:31
O 192.168.12.0/30 [110/10] is directly connected, r1-eth0, 00:03:21
C>* 192.168.12.0/30 is directly connected, r1-eth0, 00:03:48
O>* 192.168.23.0/30 [110/20] via 192.168.12.2, r1-eth0, 00:02:21
frr-pc#

```

Figure 32. Displaying router r1's routing table.

Consider the above figure. The output displays the networks learned along with additional attributes, such as the routing protocol that discovered the network. For example, networks with the symbol `O` on the left are learned from OSPF, whereas those with symbol `C` are directly connected to the device. Furthermore, the symbol `>` on the left of the displayed network indicates that this entry is selected to reach the network. The symbol `*`, this indicates that the route is inserted in Forwarding Information Base (FIB) of the router. In this network, the routes 192.168.2.0/24, 192.168.3.0/24, and 192.168.23.0/30 are discovered and reached by the OSPF routing protocol (`O>*`).

**Step 3.** Test the connectivity between host h1 and host h2 by typing the command below.

```
ping 192.168.2.10 -c 4
```

```

Host: h1
root@frr-pc:~# ping 192.168.2.10 -c 4
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=64 time=0.207 ms
64 bytes from 192.168.2.10: icmp_seq=2 ttl=64 time=0.060 ms
64 bytes from 192.168.2.10: icmp_seq=3 ttl=64 time=0.046 ms
64 bytes from 192.168.2.10: icmp_seq=4 ttl=64 time=0.051 ms

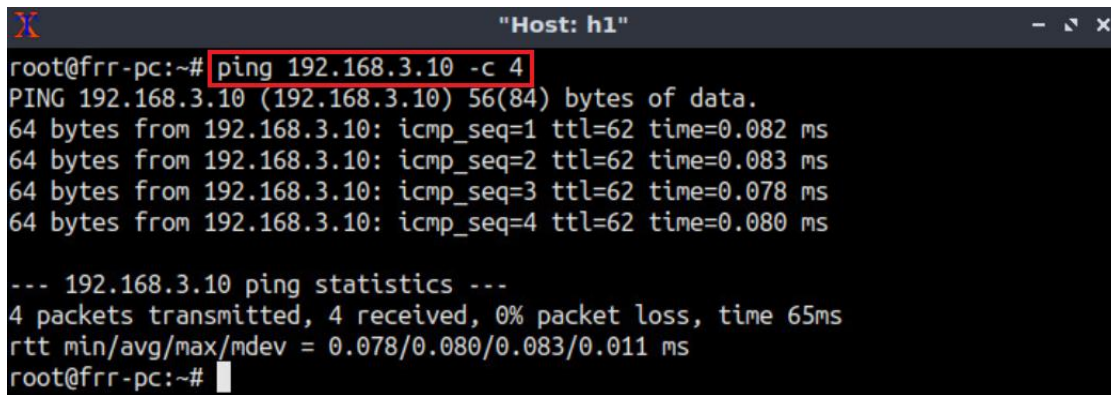
--- 192.168.2.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 70ms
rtt min/avg/max/mdev = 0.046/0.091/0.207/0.067 ms
root@frr-pc:~#

```

Figure 33. Performing a connectivity test between host h1 and host h2.

**Step 4.** Test the connectivity between host h1 and host h3 by issuing the command below.

```
ping 192.168.3.10 -c 4
```



```

Host: h1
root@frr-pc:~# ping 192.168.3.10 -c 4
PING 192.168.3.10 (192.168.3.10) 56(84) bytes of data:
64 bytes from 192.168.3.10: icmp_seq=1 ttl=62 time=0.082 ms
64 bytes from 192.168.3.10: icmp_seq=2 ttl=62 time=0.083 ms
64 bytes from 192.168.3.10: icmp_seq=3 ttl=62 time=0.078 ms
64 bytes from 192.168.3.10: icmp_seq=4 ttl=62 time=0.080 ms

--- 192.168.3.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 65ms
rtt min/avg/max/mdev = 0.078/0.080/0.083/0.011 ms
root@frr-pc:~#

```

Figure 34. Performing a connectivity test between host h1 and host h3.

**Step 5.** Similarly, you can test the connectivity on hosts h2 and h3 by following steps 1 and 2.

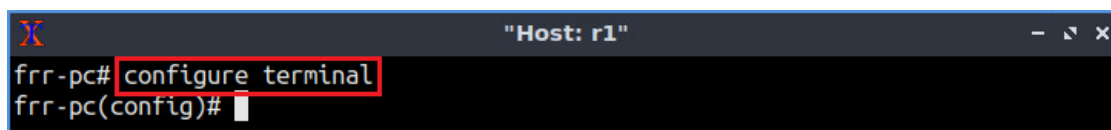
## 4 Configuring OSPF passive interfaces

In this section, you will configure OSPF passive interfaces in routers r1, r2, and r3. These configurations will disable the forwarding of OSPF packets out of the interfaces. The use of passive interfaces increases security and reduces the network traffic within the network it is implemented on. The passive interfaces belong to the link connecting the routers to their hosts, i.e., the interface eth0 of each router.

### 4.1 Configuring passive interfaces in routers r1, r2, and r3

**Step 1.** Issue the following command to enter configuration mode.

```
configure terminal
```



```

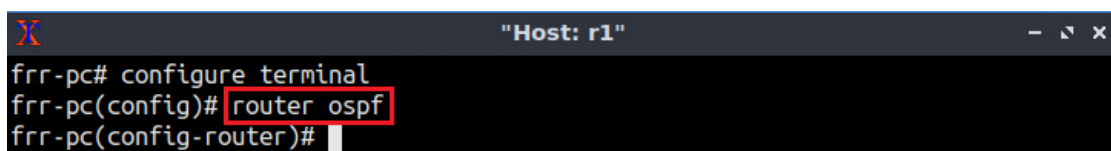
Host: r1
frr-pc# configure terminal
frr-pc(config)#

```

Figure 35. Entering configuration mode in router r1.

**Step 2.** To configure OSPF in the router type the command shown below. This command will enable OSPF configuration mode.

```
router ospf
```



```

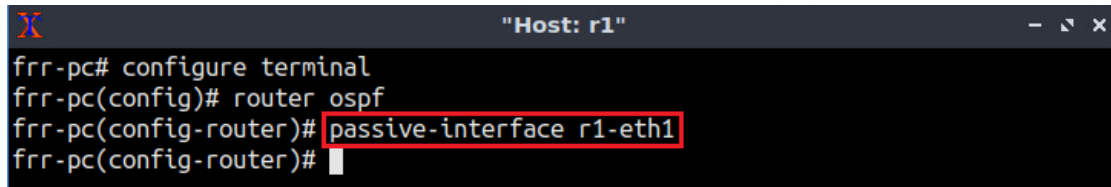
Host: r1
frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)#

```

Figure 36. Configuring OSPF in router r1.

**Step 3.** In this step, you will configure *r1-eth1* to be passive interface. To do so, type the command shown below:

```
passive-interface r1-eth1
```



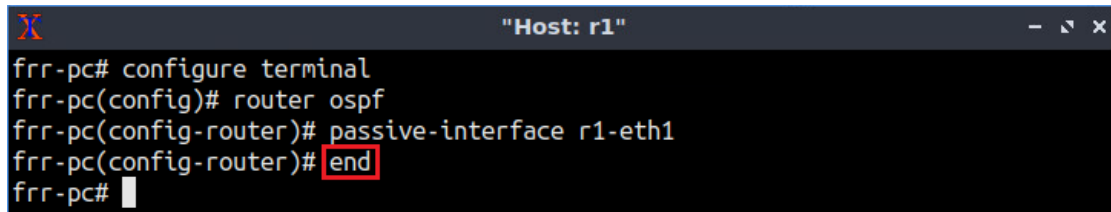
```

"Host: r1"
frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# passive-interface r1-eth1
frr-pc(config-router)#
  
```

Figure 37. Configuring interface *r1-eth1* as passive.

**Step 4.** Type the following command to exit configuration mode.

```
end
```



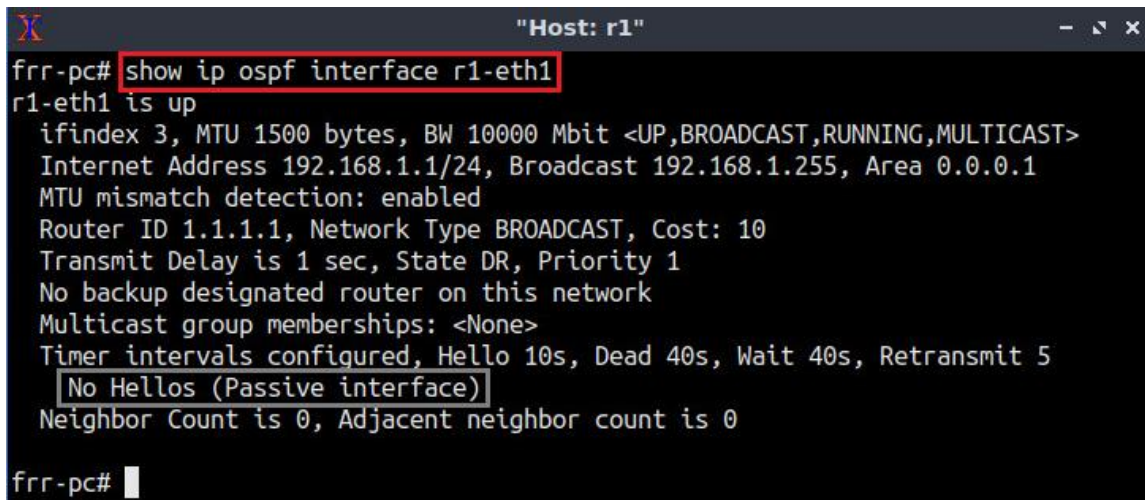
```

"Host: r1"
frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# passive-interface r1-eth1
frr-pc(config-router)# end
frr-pc#
  
```

Figure 38. Exiting configuration mode.

**Step 5.** Issue the following command to verify that the configuration was applied correctly

```
show ip ospf interface r1-eth1
```



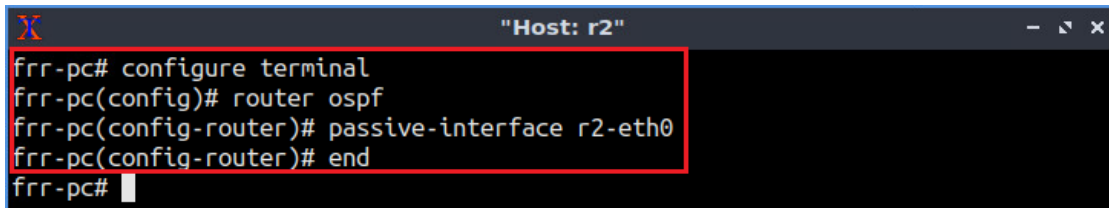
```

"Host: r1"
frr-pc# show ip ospf interface r1-eth1
r1-eth1 is up
  ifindex 3, MTU 1500 bytes, BW 10000 Mbit <UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 192.168.1.1/24, Broadcast 192.168.1.255, Area 0.0.0.1
  MTU mismatch detection: enabled
  Router ID 1.1.1.1, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State DR, Priority 1
  No backup designated router on this network
  Multicast group memberships: <None>
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
  No Hellos (Passive interface)
  Neighbor Count is 0, Adjacent neighbor count is 0

frr-pc#
  
```

Figure 39. Verifying passive interface configuration.

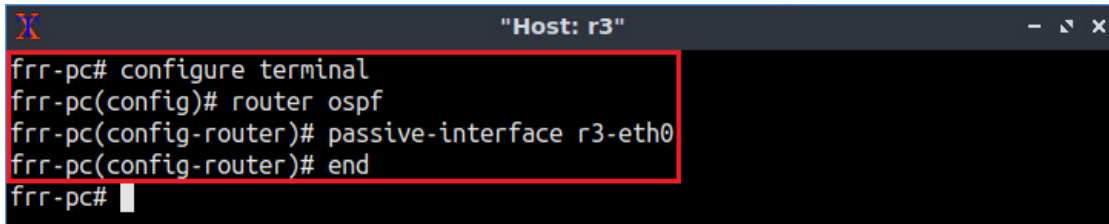
**Step 6.** Repeat steps 2 through 4 in order to configure the passive interface *r2-eth0*. The necessary steps are summarized in the following figure.



```
frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# passive-interface r2-eth0
frr-pc(config-router)# end
frr-pc#
```

Figure 40. Configuring OSPF passive interface in router r2.

**Step 7.** Repeat steps 2 through 5 in order to configure the passive interface *r3-eth0*. The necessary steps are summarized in the following figure.



```
frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# passive-interface r3-eth0
frr-pc(config-router)# end
frr-pc#
```

Figure 41. Configuring OSPF passive interfaces in router r3.

This concludes lab 4. Stop the emulation and then exit out of MiniEdit.

## References

1. Moy, J. "RFC 1247: OSPF version 2." 1991.
2. Moy, J. "RFC 2328: OSPF version 2." 1998.
3. Coltun, R., Ferguson, D., Moy, J., Lindem, A. "RFC 5340, OSPF for IPv6." 2008.
4. Peterson, L. "Computer networks: a systems approach." 6th Edition, 2021.
5. Tanenbaum, D. Wetherall. "Computer networks." 5th Edition. 2012.
6. J. Kurose, K. Ross. "Computer networking, a top-down approach." 7th Edition. 2017.
7. Teare, D., Vachon, B., Graziani, R. "Implementing Cisco IP routing (ROUTE) foundation learning guide: (CCNP ROUTE 300-101)." 2014.
8. Linux foundation collaborative projects. "FRR routing documentation." 2017 [Online] Available: <http://docs.frrouting.org/en/latest/>.



UNIVERSITY OF  
**SOUTH CAROLINA**

## **OPEN SHORTEST PATH FIRST**

### **Exercise 1: Configuring Multi-Area OSPFv2**

Document Version: **10-22-2021**



Award 1829698

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

## Contents

1	Exercise description .....	3
1.1	Credentials .....	3
2	Setting the environment.....	3
3	Deliverables.....	6

## 1 Exercise description

This exercise is a practice to the Open Shortest Path First (OSPF). The exercise requires configuring multi-area OSPF to provide connectivity between two hosts located in different areas. The topology in the figure below provides the information such as the devices' names, IP addresses, interface names and the OSPF areas where each device is located.

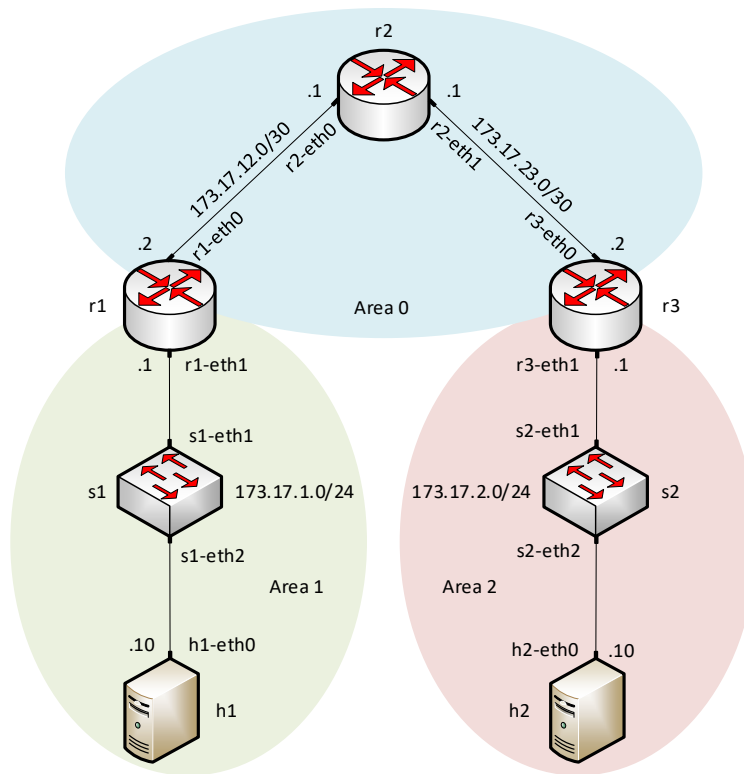


Figure 1. Exercise topology.

### 1.1 Credentials

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

Table 1. Credentials to access the Client machine.

Device	Account	Password
Client	admin	password

## 2 Setting the environment

Follow the steps below to set the environment.

**Step 1.** Open MiniEdit by double-clicking the shortcut on the desktop. If a password is required type `password`.



Figure 2. MiniEdit shortcut.

**Step 2.** Load the topology located at `/home/frr/OSPF_Labs/Exercise1/`.

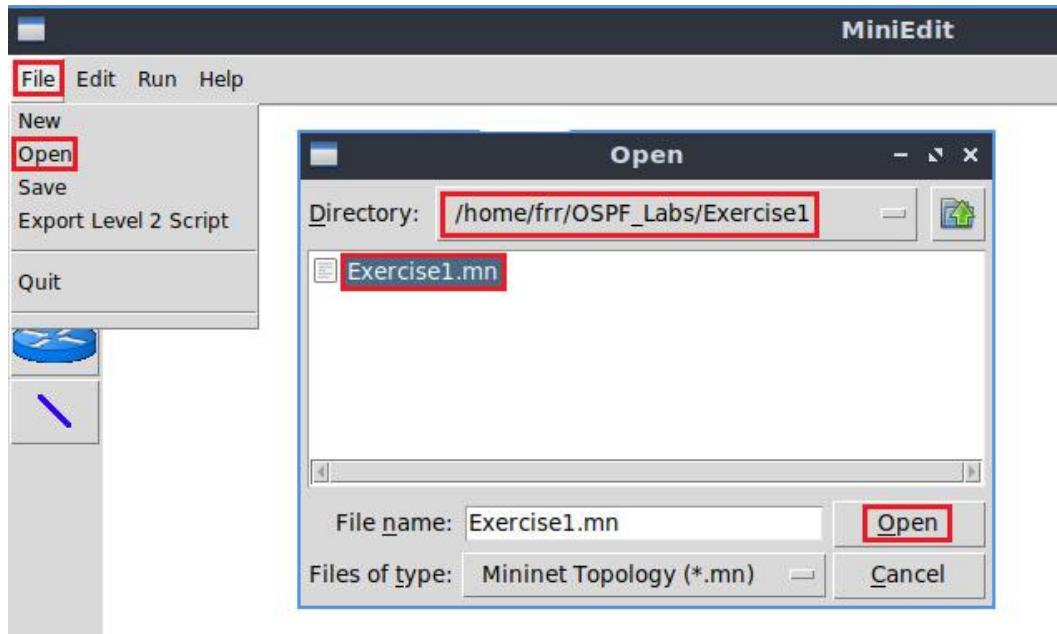


Figure 3. MiniEdit's open dialog.

**Step 3.** Run the emulation by clicking on the button located on the lower left-hand side.





Figure 4. Starting the emulation.

**Step 4.** Open the Linux terminal by clicking on the icon located in the taskbar.

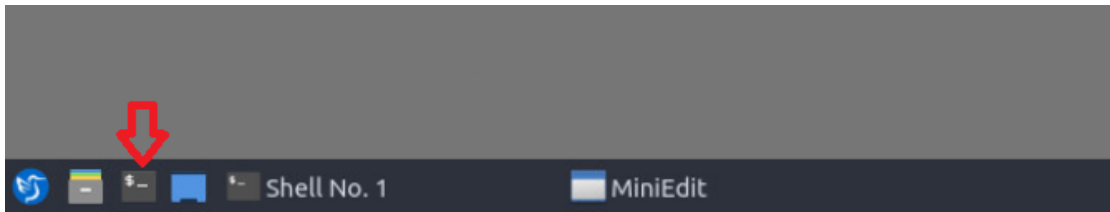


Figure 5. Opening Linux terminal.

**Step 5.** Issue the command shown below to set the lab environment. This command configures the IP addresses in all routers. If a password is required, type `password`.

```
sudo set_env exercise1
```

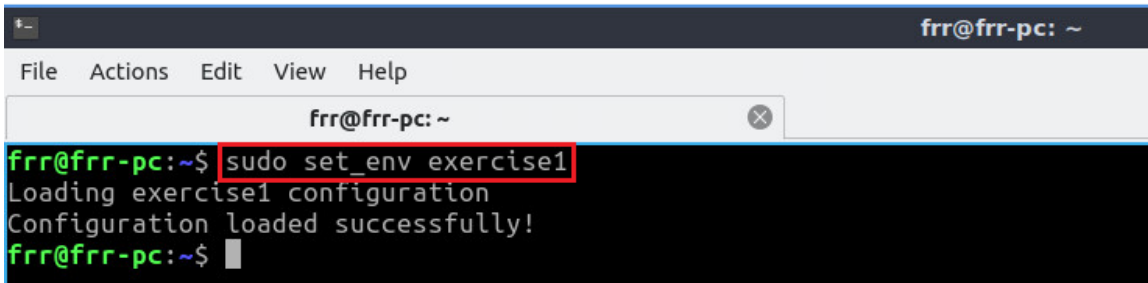


Figure 6. Setting the lab environment.

**Step 6.** Type the following command to exit the Linux terminal.

```
exit
```

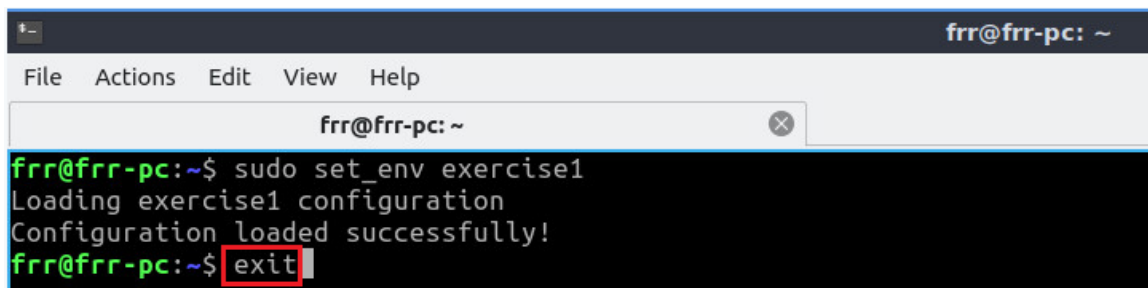


Figure 7. Exiting the Linux terminal.

### 3 Deliverables

Follow the steps below to complete the exercise.

- a) Open a terminal in each router, then, start zebra and the OSPF daemons.
- b) Access each router CLI using the `vttysh` command.
- c) Verify the directly connected routes in each router.
- d) Assign an OSPF router ID to each router.
- e) Configure router r1 to participate in the OSPF process of areas 0 and 1.
- f) Configure router r2 to participate in the OSPF process of area 0.
- g) Configure router r3 to participate in the OSPF process of areas 0 and 2.
- h) Set the passive interfaces in routers r1 and r3.
- i) Verify the routing and OSPF neighbor tables of each router.
- j) Perform a connectivity test from host h1 to host h2.



UNIVERSITY OF  
**SOUTH CAROLINA**

## OPEN SHORTEST PATH FIRST

### Lab 5: Configuring OSPFv2 with Default Route

Document Version: **10-22-2021**



Award 1829698

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

## Contents

Overview .....	3
Objectives.....	3
Lab settings .....	3
Lab roadmap .....	3
1 Introduction to advanced OSPF features.....	3
1.1 Default route .....	4
1.2 Propagating the default route in OSPF .....	4
2 Lab topology.....	5
2.1 Lab settings.....	6
2.2 Loading the topology.....	7
2.3 Load the zebra daemon and verify the configuration .....	10
3 Configuring OSPF .....	14
3.1 Configuring single-area OSPF in routers r1, r2, and r3 .....	14
3.2 Verifying Connectivity .....	18
4 Configuring a static route .....	19
5 Propagating default routes in OSPF.....	21
5.1 Configuring a static route in router r2 .....	21
5.2 Propagating the default route to routers r1 and r3 in OSPF.....	23
5.3 Verifying Connectivity .....	25
References .....	26

## Overview

This lab shows how to propagate a default route between routers running Open Shortest Path First (OSPF). A default route establishes an exit point from a router when the packet destination is not available in the routing table. In this lab, you will configure OSPF to advertise the default route to the other routers.

## Objectives

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

1. Configure single-area OSPF.
2. Establish a default static route.
3. Propagate the default route throughout the entire OSPF autonomous system.

## Lab settings

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

Table 1. Credentials to access Client machine.

Device	Account	Password
Client	admin	password

## Lab roadmap

This lab is organized as follows:

1. Section 1: Introduction to advanced OSPF features.
2. Section 2: Lab topology.
3. Section 3: Configuring OSPF.
4. Section 4: Configuring a static route.
5. Section 5: Propagating the default route in OSPF.

### 1 Introduction to advanced OSPF features

OSPF is a widely used Interior Gateway Protocol (IGP) in large enterprise networks. In deployment, there are several scenarios where network administrators have to change the default configuration inside the routing protocol and configure some advanced features. By default, OSPF does not share default routes, and the timer to share OSPF

packets is predefined. However, Free Range Routing (FRR) has the flexibility to change these values and configure advanced features.

### 1.1 Default route

The default route is an entry in the forwarding table of a router or a setting on a host that determines where the packet should be forwarded when no specific route can be determined for a given Internet Protocol (IP) destination address. The device to which the default route points is often called the default gateway. In IPv4, the default route is designated as the zero-address 0.0.0.0/0, which matches against any given IP address. Thus, it is used as the last resort for matching IP addresses.

Consider Figure 1. The campus network contains three routers, where router r2 is the border router that connects to the Internet (ISP). Since the Internet has a plethora of IP addresses that are not infeasible to fit within a routing table, router r2 will configure a default route (0.0.0.0/0 via r2-eth0) that points to the ISP and forward all unmatched traffic flows through this interface.

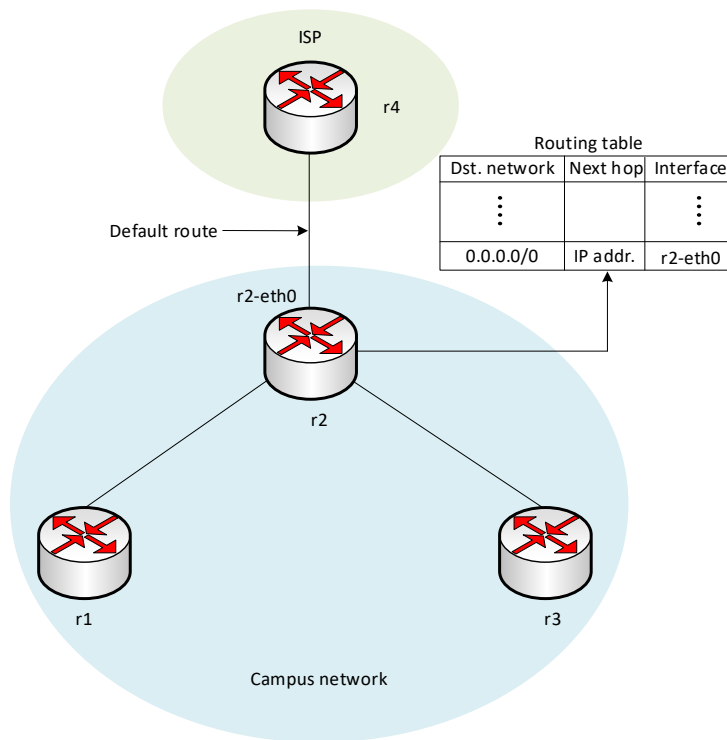


Figure 1. Default route.

The default route is an entry in the forwarding table of a router or a setting on a host that determines where the packet should be forwarded when no specific route can be determined for a given Internet Protocol (IP) destination address.

### 1.2 Propagating the default route in OSPF

OSPF shares dynamic routes between routers in the same area. However, when a router is configured with a default route, this route is not propagated to other routers. This is the default behavior of OSPF, but it can be adjusted so that also the default route is shared with other routers.

Consider Figure 2. The campus network runs OSPF as an IGP, and router r2 configures a default route to the ISP. Moreover, router r2 is configured to propagate the default route to its OSPF neighbors. Therefore, routers r1 and r3 will receive and store the default route in their forwarding table, where it will point to the interface facing router r2.

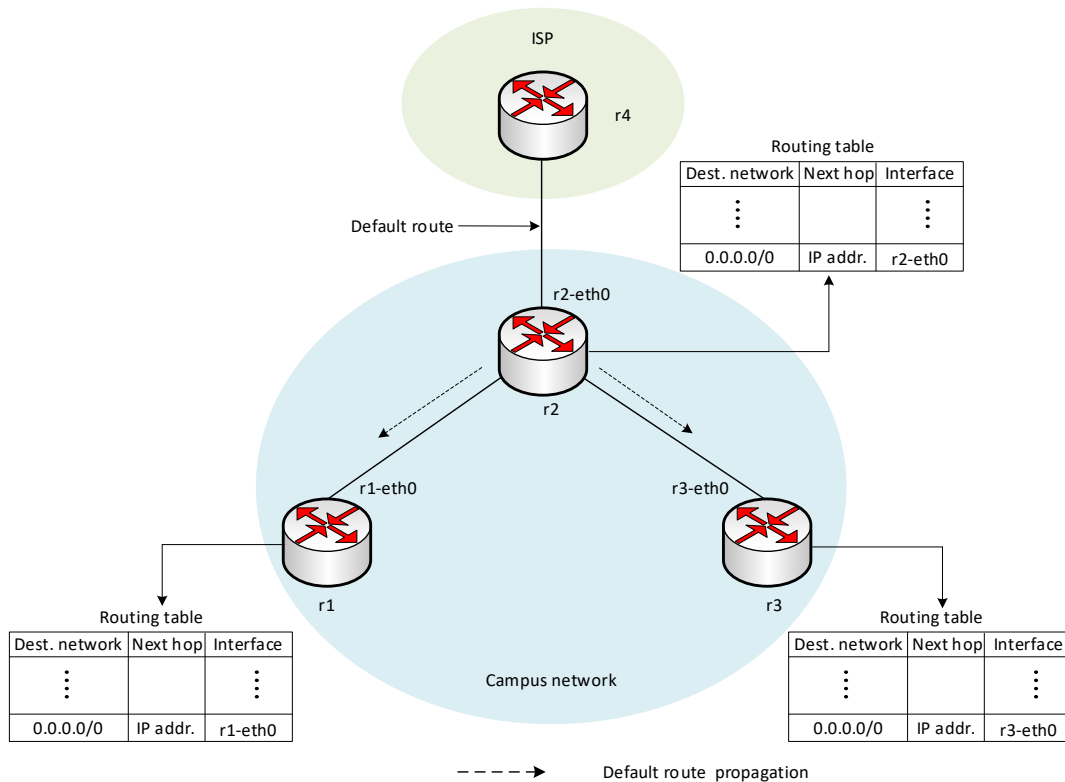


Figure 2. Propagating the default route in OSPF.

## 2 Lab topology

Consider Figure 3. The topology comprises two Autonomous Systems (ASes), an internal network, and an external network. The communication within the internal network is performed via single-area OSPF. The internal network communicates to the external network using a static route.

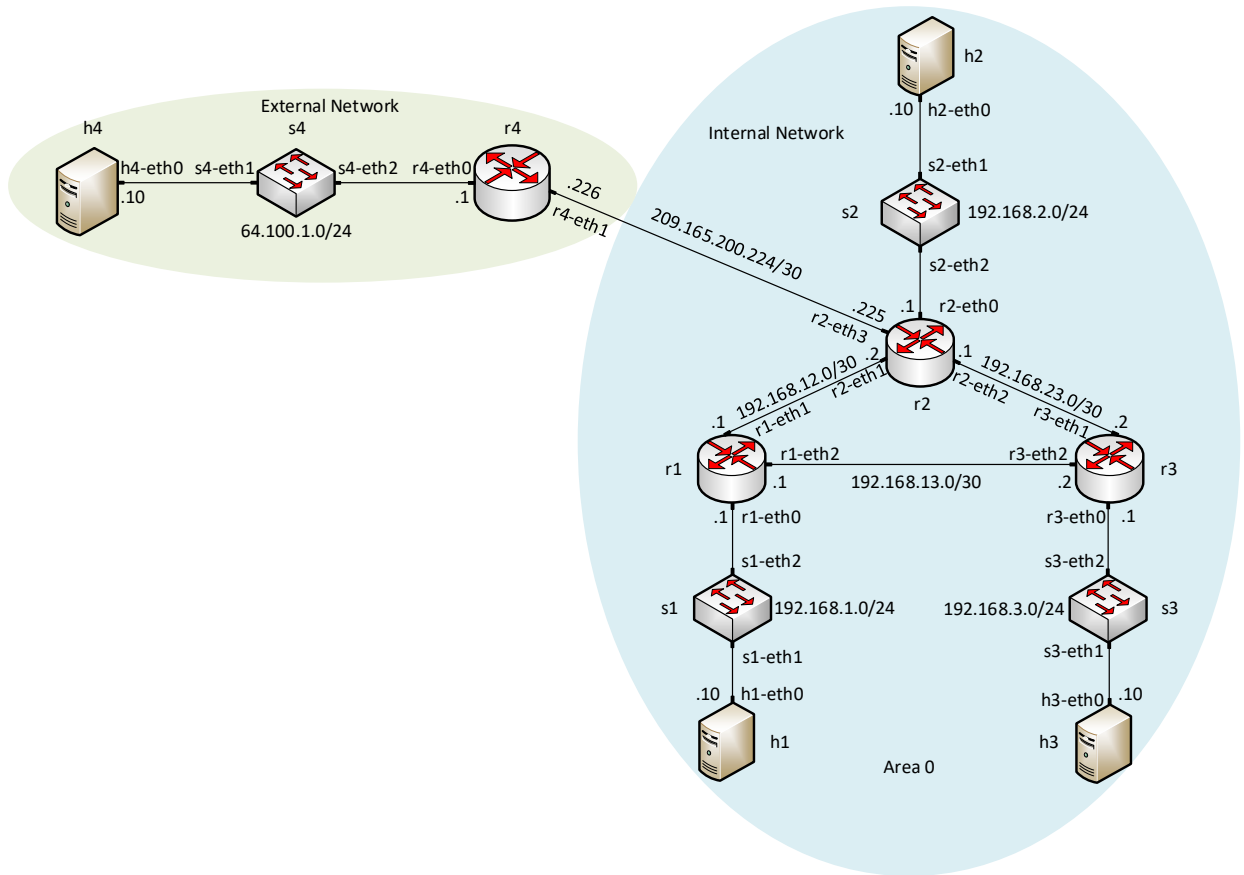


Figure 3. Topology.

## 2.1 Lab settings

The information in Table 2 will be needed in order to complete the lab. You need to configure the topology according to the IP addresses assigned for each interface in the table.

Table 2. Topology Information

Device	Interface	IPv4 Address	Subnet	Default Gateway
Router 1 (r1)	r1-eth0	192.168.1.1	/24	N/A
	r1-eth1	192.168.12.1	/30	N/A
	r1-eth2	192.168.13.1	/30	N/A
Router 2 (r2)	r2-eth0	192.168.2.1	/24	N/A
	r2-eth1	192.168.12.2	/30	N/A
	r2-eth2	192.168.23.1	/30	N/A
	r2-eth3	209.165.200.225	/30	N/A



Router 3 (r3)	r3-eth0	192.168.3.1	/24	N/A
	r3-eth1	192.168.23.2	/30	N/A
	r3-eth2	192.168.13.2	/30	N/A
Router 4 (r4)	r4-eth0	64.100.1.1	/24	N/A
	r4-eth1	209.165.200.226	/30	N/A
Host 1 (h1)	h1-eth0	192.168.1.10	/24	192.168.1.1
Host 2 (h2)	h2-eth0	192.168.2.10	/24	192.168.2.1
Host 3 (h3)	h3-eth0	192.168.3.10	/24	192.168.3.1
Host 4 (h4)	h4-eth0	64.100.1.10	/24	64.100.1.1

## 2.2 Loading the topology

**Step 1.** Launch MiniEdit by double clicking on the shortcut located on the desktop. When prompted for a password, type `password`.



Figure 4. MiniEdit shortcut.

**Step 2.** In MiniEdit's menu bar, click on *File* then *open* to load the lab topology. Locate the *lab5.mn* topology file in the default directory, */home/frr/OSPF\_Labs/lab5* and click on *Open*.

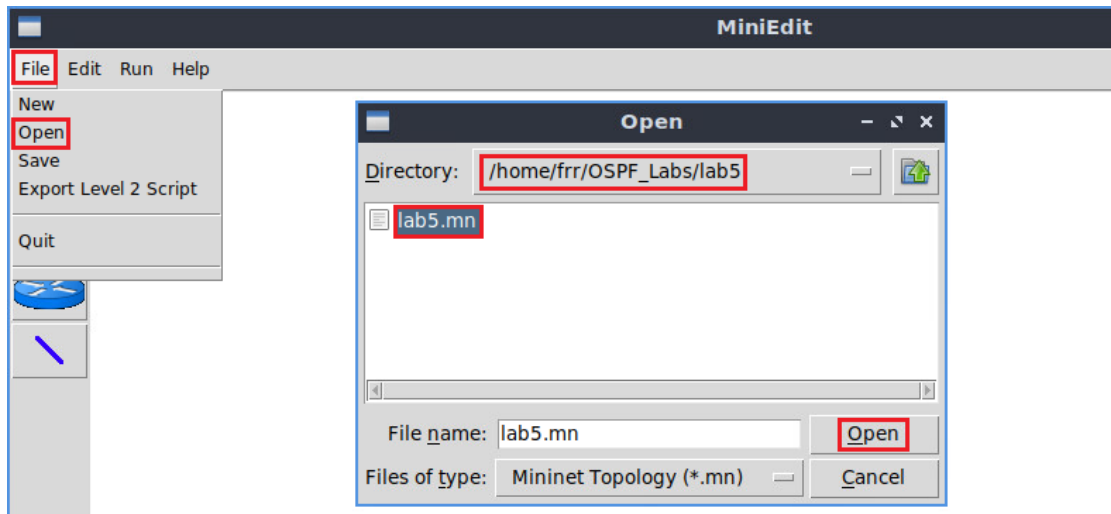


Figure 5. MiniEdit's open dialog.

**Step 3.** Click on the *Run* button located on the lower left-hand side to run the emulation.



Figure 6. Starting the emulation.

At this point, the topology is loaded with all the required network components. Next, you will execute a script that will configure the interfaces of the routers.

Automating the configuration of interfaces with the script is intended to facilitate the lab experience and to focus on the OSPF experiment.

**Step 4.** Open the Linux terminal by clicking on the icon located in the taskbar.

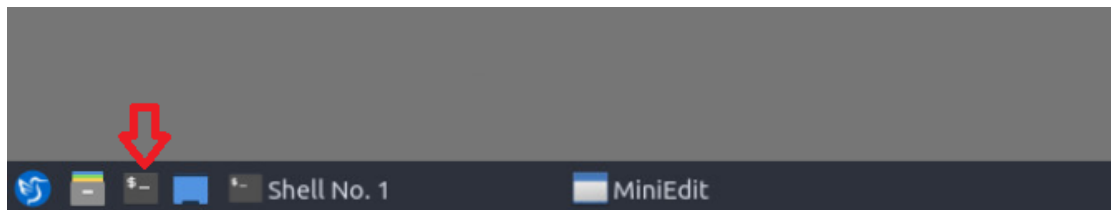
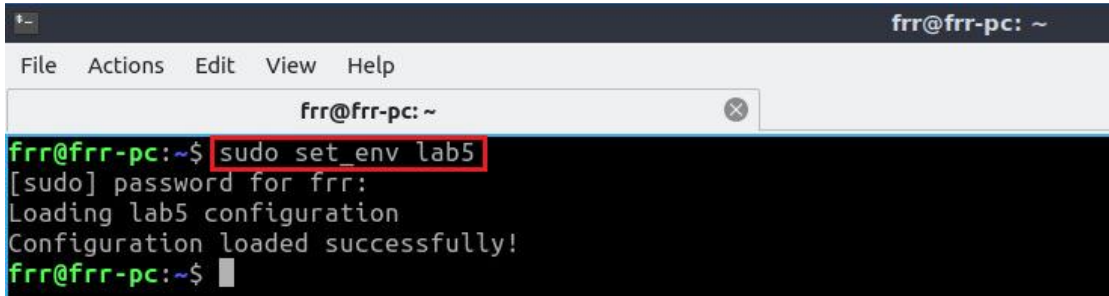


Figure 7. Opening Linux terminal.

**Step 5.** Issue the command shown below to set the lab environment. This command configures the IP addresses of the routers' interface according to Table 2. If a password is required, type `password`.

```
sudo set_env lab5
```

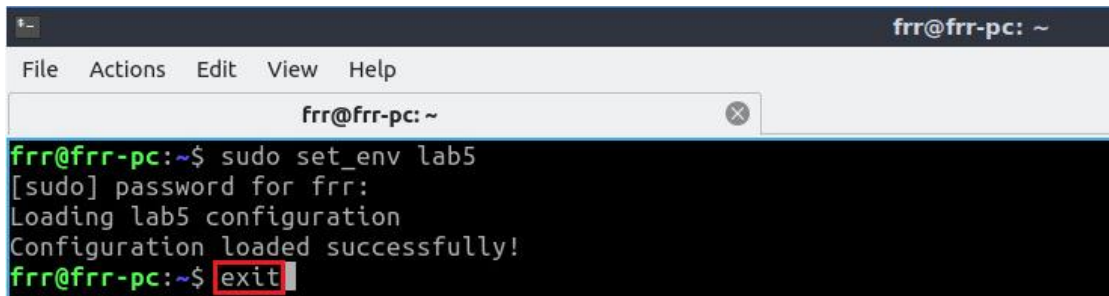


```
frr@frr-pc: ~  
File Actions Edit View Help  
frr@frr-pc: ~  
frr@frr-pc:~$ sudo set_env lab5  
[sudo] password for frr:  
Loading lab5 configuration  
Configuration loaded successfully!  
frr@frr-pc:~$
```

Figure 8. Setting the lab environment.

**Step 6.** Type the following command to exit the Linux terminal.

```
exit
```



```
frr@frr-pc: ~  
File Actions Edit View Help  
frr@frr-pc: ~  
frr@frr-pc:~$ sudo set_env lab5  
[sudo] password for frr:  
Loading lab5 configuration  
Configuration loaded successfully!  
frr@frr-pc:~$ exit
```

Figure 9. Exiting the Linux terminal.

**Step 7.** Click on Mininet's terminal, i.e., the one launched when MiniEdit was started.



Figure 10. Opening Mininet's terminal.

**Step 8.** Issue the following command to display the interface names and connections.

```
links
```

```

mininet> links
h1-eth0<->s1-eth1 (OK OK)
s1-eth2<->r1-eth0 (OK OK)
h2-eth0<->s2-eth1 (OK OK)
s2-eth2<->r2-eth0 (OK OK)
h3-eth0<->s3-eth1 (OK OK)
s3-eth2<->r3-eth0 (OK OK)
h4-eth0<->s4-eth1 (OK OK)
s4-eth2<->r4-eth0 (OK OK)
r1-eth1<->r2-eth1 (OK OK)
r2-eth2<->r3-eth1 (OK OK)
r1-eth2<->r3-eth2 (OK OK)
r2-eth3<->r4-eth1 (OK OK)
mininet>

```

Figure 11. Displaying network interfaces.

In the figure above, the link displayed within the gray box indicates that interface *eth0* of host *h2* connects to interface *eth1* of switch *s2* (i.e., *h2-eth0<->s2-eth1*).

### 2.3 Load the zebra daemon and verify the configuration

In this section, you will verify that the IP addresses listed in Table 2 and inspect the routing table of the routers.

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

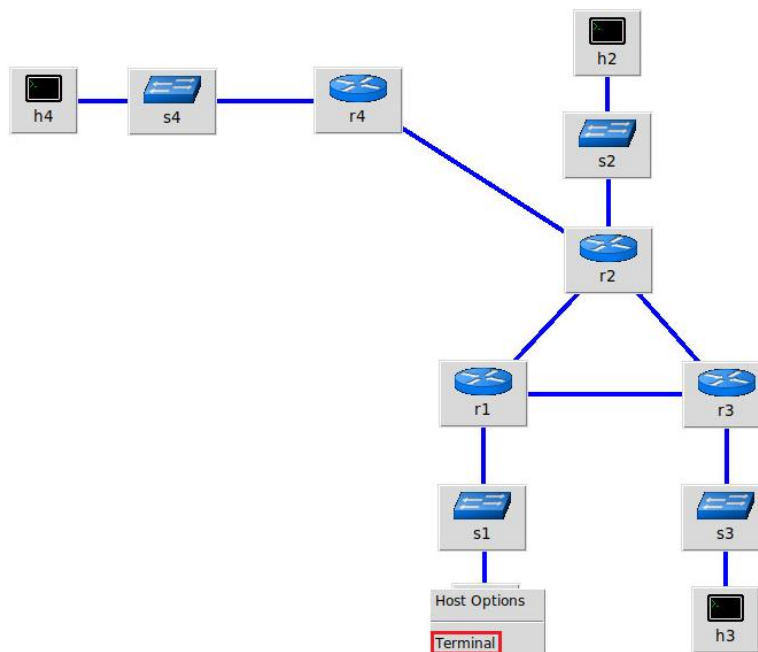
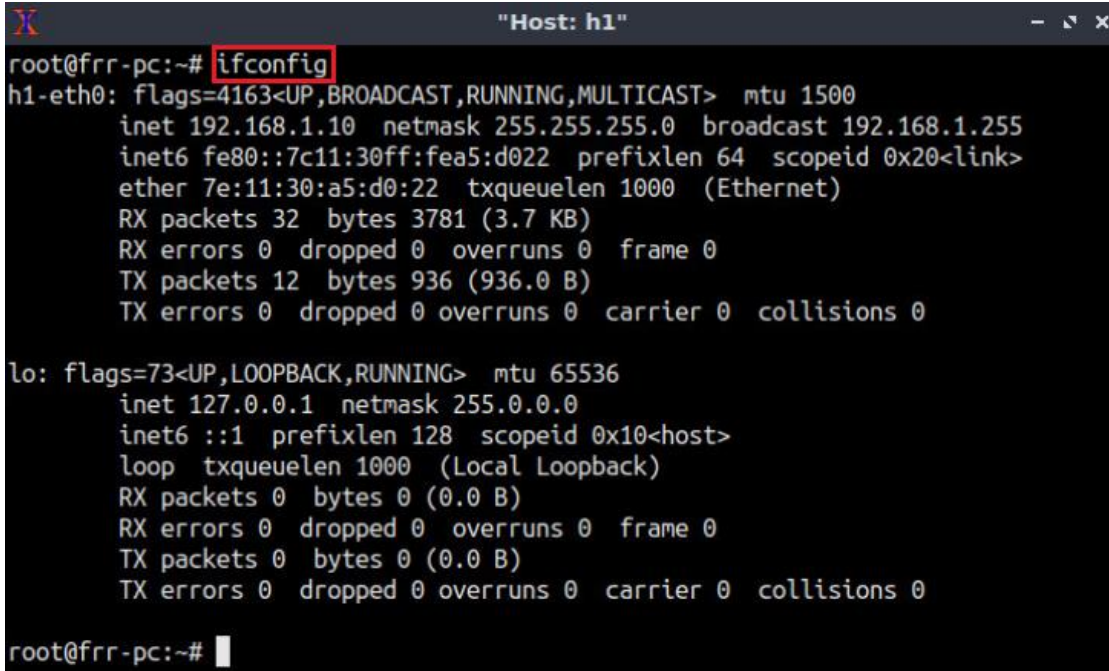


Figure 12. Opening a terminal on host *h1*.

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

```
ifconfig
```



```
root@frr-pc:~# ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.10 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::7c11:30ff:fea5:d022 prefixlen 64 scopeid 0x20<link>
    ether 7e:11:30:a5:d0:22 txqueuelen 1000 (Ethernet)
    RX packets 32 bytes 3781 (3.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 12 bytes 936 (936.0 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@frr-pc:~#
```

Figure 13. The output of `ifconfig` command.

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

```
route
```

```

Host: h1
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
  inet 192.168.1.10 netmask 255.255.255.0 broadcast 192.168.1.255
  inet6 fe80::7c11:30ff:fea5:d022 prefixlen 64 scopeid 0x20<link>
  ether 7e:11:30:a5:d0:22 txqueuelen 1000 (Ethernet)
  RX packets 32 bytes 3781 (3.7 KB)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 12 bytes 936 (936.0 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@frr-pc:~# 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@frr-pc:~#
  
```

Figure 14. Displaying the default route in host h1.

**Step 4.** To verify the hosts h2 and h3 default route, proceed by repeating from step 1 to step 3. Similar results should be observed.

**Step 5.** You will validate that the router interfaces are configured according to Table 2. To proceed, hold right-click on router r1 and select *Terminal*.

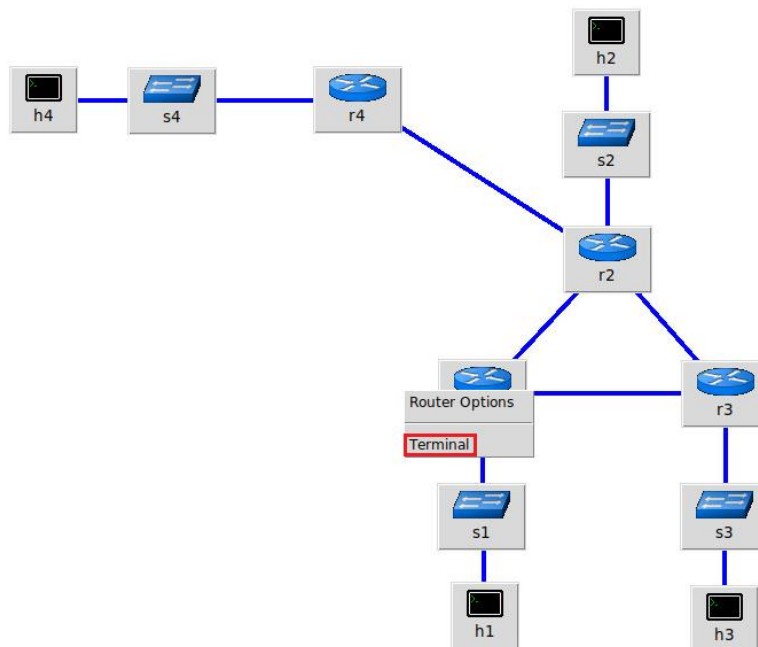
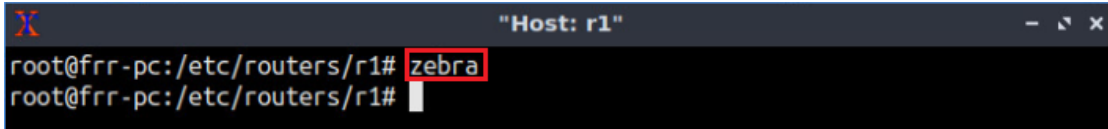


Figure 15. Opening a terminal on router r1.

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

```
zebra
```



```

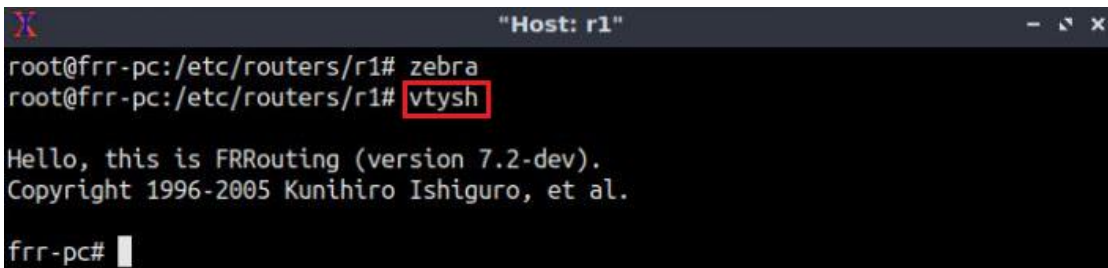
Host: r1
root@frr-pc:/etc/routers/r1# zebra
root@frr-pc:/etc/routers/r1#

```

Figure 16. Starting zebra daemon.

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

```
vttysh
```



```

Host: r1
root@frr-pc:/etc/routers/r1# zebra
root@frr-pc:/etc/routers/r1# vtysh

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

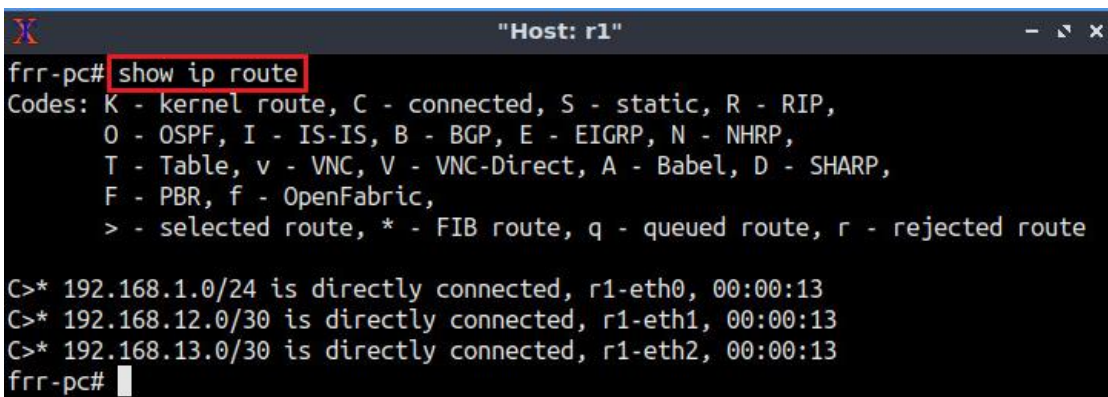
frr-pc#

```

Figure 17. Starting vtysh on router r1.

**Step 8.** Type the following command on router r1 terminal to verify the routing table. It will list all the directly connected networks. Notice that 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
frr-pc# 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, 00:00:13
C>* 192.168.12.0/30 is directly connected, r1-eth1, 00:00:13
C>* 192.168.13.0/30 is directly connected, r1-eth2, 00:00:13
frr-pc#

```

Figure 18. Displaying the routing table of router r1.

**Step 9.** To verify the routing table of the other routers, proceed similarly by following from step 5 to step 8 on the corresponding routers.

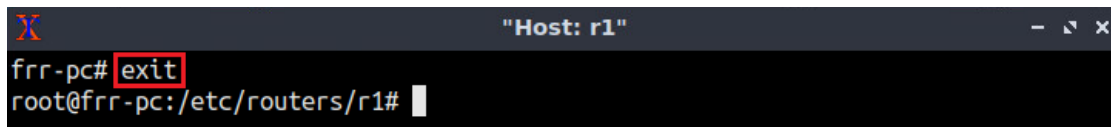
### 3 Configuring OSPF

In this section, you will configure the OSPF routing protocol in routers r1, r2, and r3. OSPF will be handling the routing among the routers r1, r2, and r3. To configure OSPF in each router, you must enable the interfaces that need to send and receive OSPF packets with other routers in the same area. You will run single-area OSPF and assign the number 0 for the area (also known as the backbone area).

#### 3.1 Configuring single-area OSPF in routers r1, r2, and r3

**Step 1.** In router's r1 terminal, type the following command to exit from router's r1 CLI.

```
exit
```

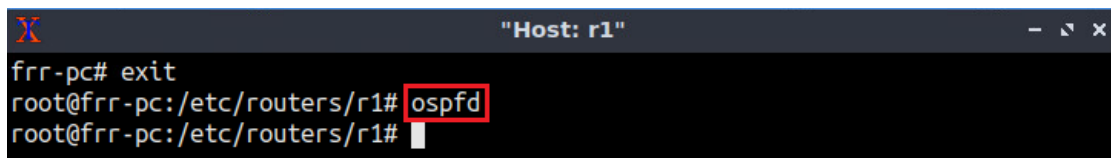


```
Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1#
```

Figure 19. Exiting from configuration mode.

**Step 2.** In the terminal of router r1, enable and start the OSPF routing protocol by typing the following command:

```
ospfd
```

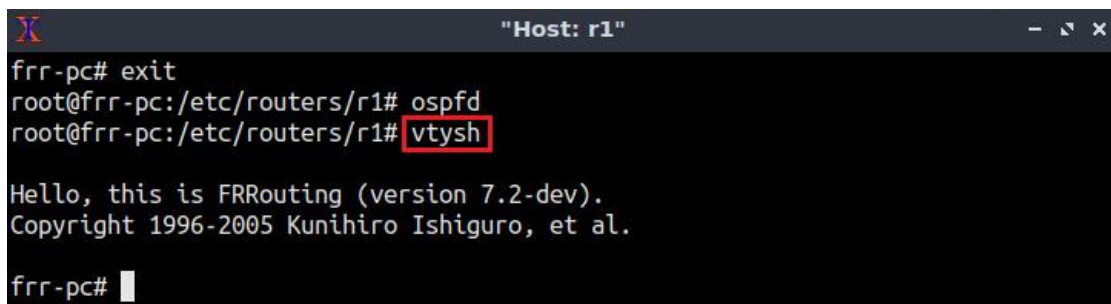


```
Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1#
```

Figure 20. Starting the OSPF daemon in router r1.

**Step 3.** To load the shell for executing the FRR commands, issue the following command:

```
vttysh
```



```
Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vttysh

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

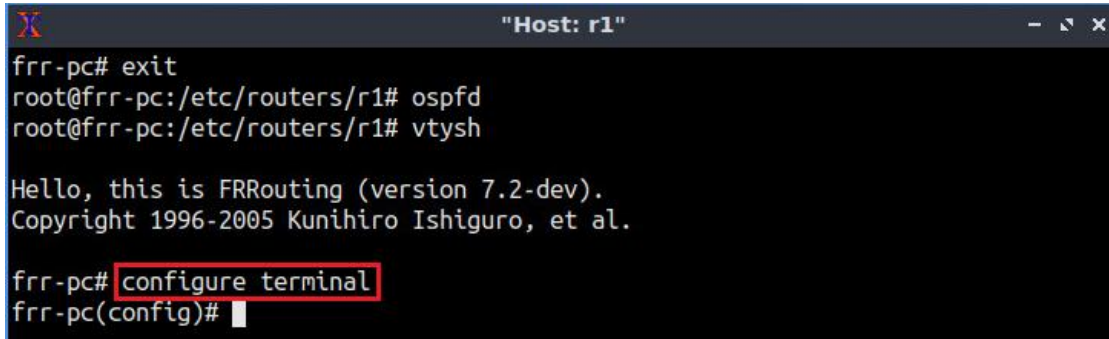
frr-pc#
```

Figure 21. Starting vtysh on router r1.

**Step 4.** To enter configuration mode in router r1, issue the following command:



```
configure terminal
```



```
"Host: r1"
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

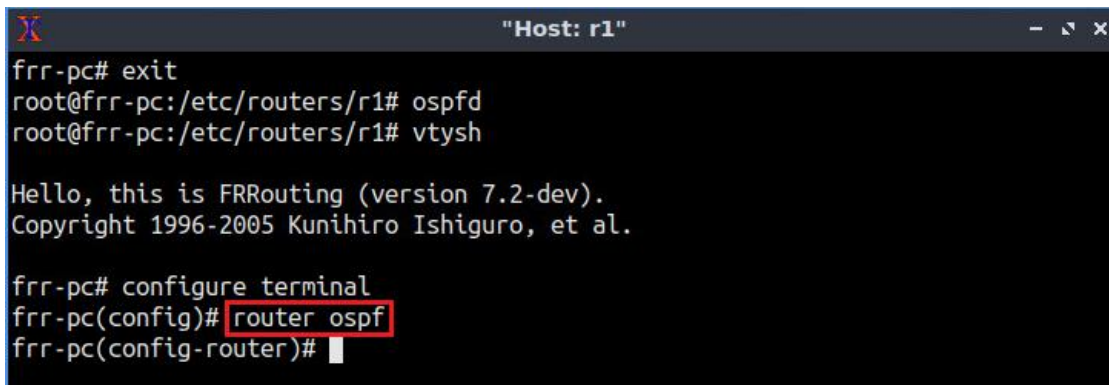
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)#
```

Figure 22. Enabling configuration mode in router r1.

**Step 5.** To configure OSPF, type the command shown below. This command will enable OSPF configuration mode where you can assign the networks directly connected to the router r1.

```
router ospf
```



```
"Host: r1"
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

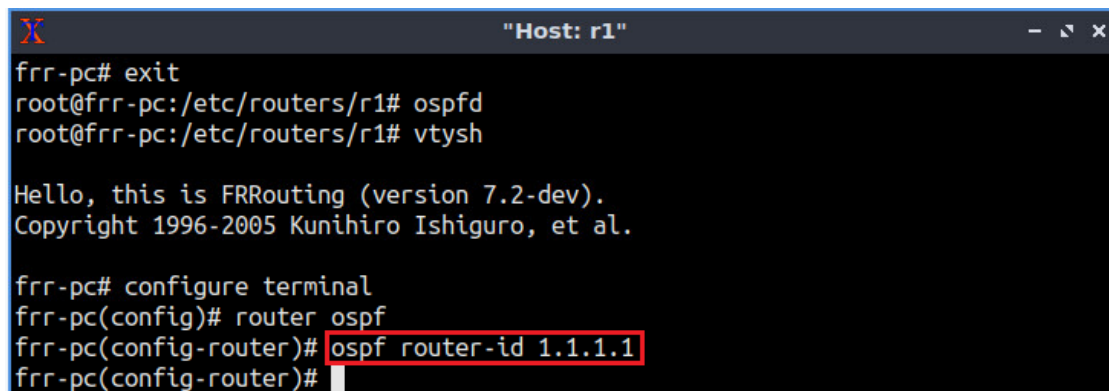
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)#
```

Figure 23. Configuring OSPF on router r1.

**Step 6.** To establish router r1 ID, issue the following command:

```
ospf router-id 1.1.1.1
```



```
"Host: r1"
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

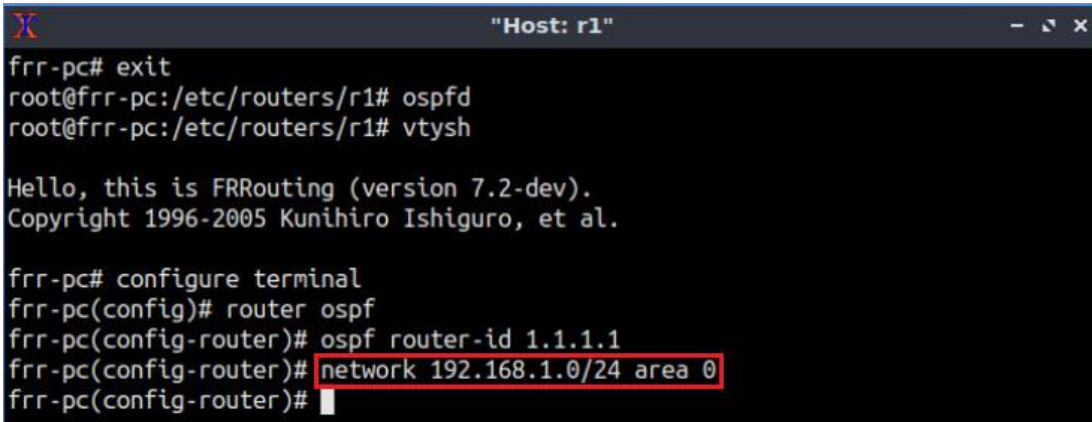
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 1.1.1.1
frr-pc(config-router)#
```

Figure 24. Assigning the OSPF router ID of 1.1.1.1 to router r1.

**Step 7.** Type the following command to enable the interface *r1-eth0* to participate in the OSPF process in area 0.

```
network 192.168.1.0/24 area 0
```

A terminal window titled "Host: r1" showing the configuration process. The user enters 'exit' from frr-pc, then 'ospfd' and 'vtysh' from the root prompt. A welcome message for FRRouting (version 7.2-dev) is displayed. The user enters 'configure terminal', then 'router ospf', 'ospf router-id 1.1.1.1', and finally 'network 192.168.1.0/24 area 0', which is highlighted with a red box. The prompt returns to frr-pc(config-router)#.

```
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

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

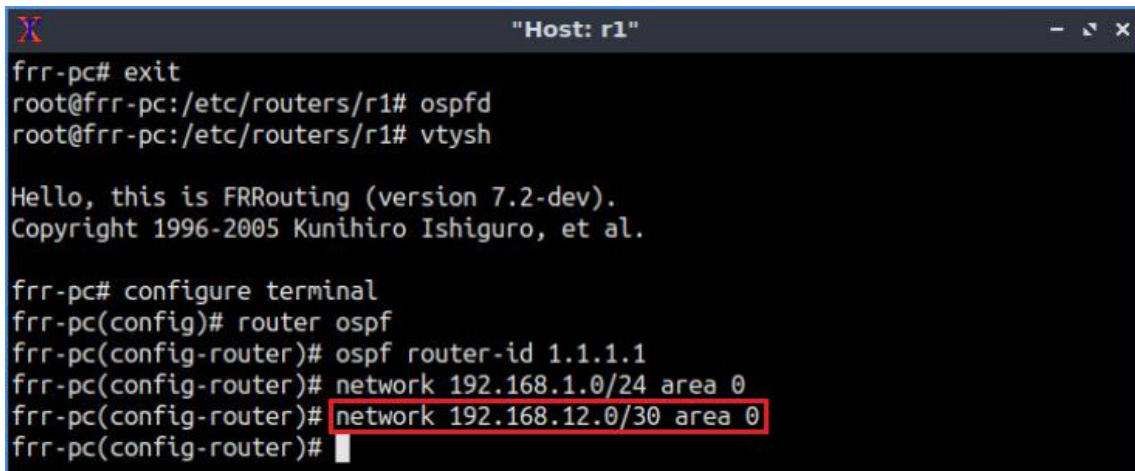
frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 1.1.1.1
frr-pc(config-router)# network 192.168.1.0/24 area 0
frr-pc(config-router)#
```

Figure 25. Enabling interface *r1-eth0* to participate in the OSPF process of area 0.

The command highlighted in the figure above, defines *r1-eth0* as the interface on which OSPF runs and the area associated with that interface (i.e., area 0).

**Step 8.** Type the following command to enable the interface *r1-eth1* to participate in the OSPF process in area 0.

```
network 192.168.12.0/30 area 0
```

A terminal window titled "Host: r1" showing the configuration process. The user enters 'exit' from frr-pc, then 'ospfd' and 'vtysh' from the root prompt. A welcome message for FRRouting (version 7.2-dev) is displayed. The user enters 'configure terminal', then 'router ospf', 'ospf router-id 1.1.1.1', 'network 192.168.1.0/24 area 0', and finally 'network 192.168.12.0/30 area 0', which is highlighted with a red box. The prompt returns to frr-pc(config-router)#.

```
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

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

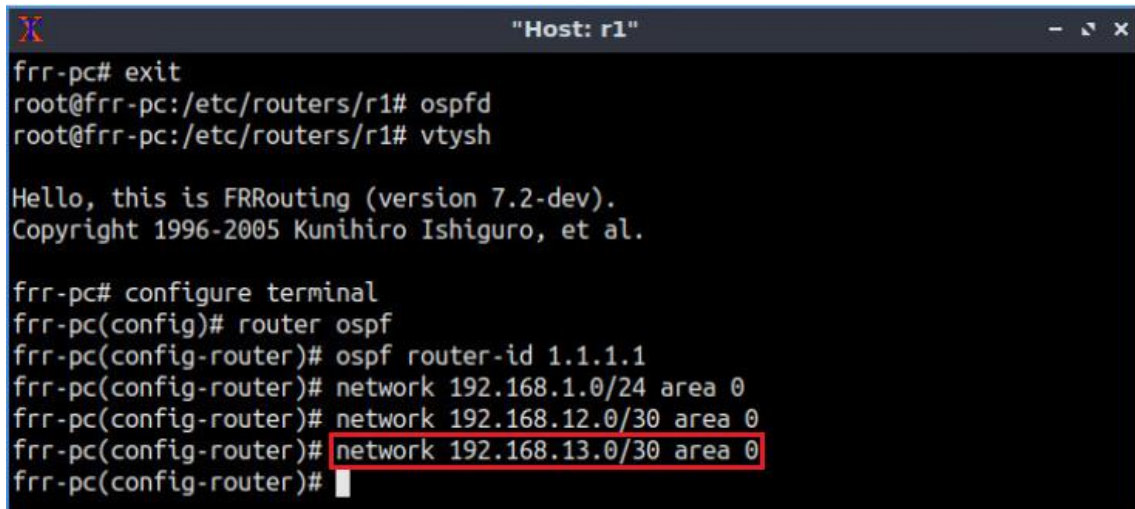
frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 1.1.1.1
frr-pc(config-router)# network 192.168.1.0/24 area 0
frr-pc(config-router)# network 192.168.12.0/30 area 0
frr-pc(config-router)#
```

Figure 26. Enabling interface *r1-eth1* to participate in the OSPF process of area 0.

The command highlighted in the figure above, defines *r1-eth1* as the interface on which OSPF runs and the area associated with that interface (i.e., area 0).

**Step 9.** Type the following command to enable the interface *r1-eth2* to participate in the OSPF process in area 0.

```
network 192.168.13.0/30 area 0
```

A terminal window titled "Host: r1" showing the configuration of OSPF on a router. The user enters 'exit' from the frr-pc prompt, then 'ospfd' and 'vtysh' from the root prompt. The terminal displays the FRRouting version 7.2-dev banner. The user enters 'configure terminal' to enter configuration mode, then 'router ospf' to enter OSPF configuration mode. The user configures the router ID as 1.1.1.1 and adds three networks to area 0: 192.168.1.0/24, 192.168.12.0/30, and 192.168.13.0/30. The last command is highlighted with a red box.

```
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

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

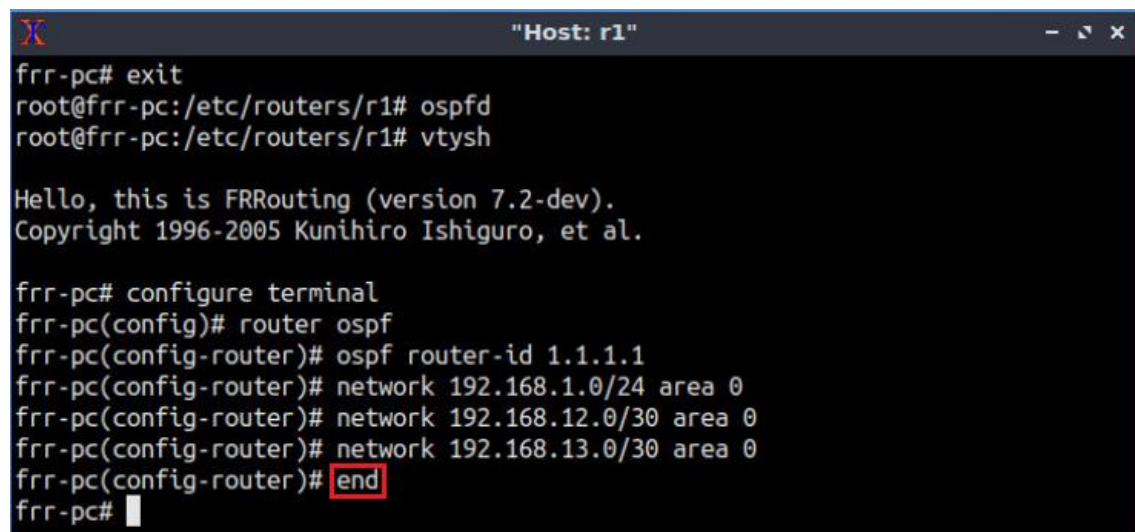
frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 1.1.1.1
frr-pc(config-router)# network 192.168.1.0/24 area 0
frr-pc(config-router)# network 192.168.12.0/30 area 0
frr-pc(config-router)# network 192.168.13.0/30 area 0
frr-pc(config-router)#
```

Figure 27. Enabling interface *r1-eth2* to participate in the OSPF process of area 0.

The command highlighted in the figure above, defines *r1-eth2* as the interface on which OSPF runs and the area associated with that interface (i.e., area 0).

**Step 10.** To exit OSPF configuration mode, type the following command:

```
end
```

A terminal window titled "Host: r1" showing the completion of OSPF configuration. The user enters 'end' to exit the OSPF configuration mode, which is highlighted with a red box. The terminal returns to the frr-pc prompt.

```
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 1.1.1.1
frr-pc(config-router)# network 192.168.1.0/24 area 0
frr-pc(config-router)# network 192.168.12.0/30 area 0
frr-pc(config-router)# network 192.168.13.0/30 area 0
frr-pc(config-router)# end
frr-pc#
```

Figure 28. Exiting OSPF configuration mode.

**Step 11.** 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 terminal, issue the commands depicted below.

```

Host: r2
frr-pc# exit
root@frr-pc:/etc/routers/r2# ospfd
root@frr-pc:/etc/routers/r2# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 2.2.2.2
frr-pc(config-router)# network 192.168.2.0/24 area 0
frr-pc(config-router)# network 192.168.12.0/30 area 0
frr-pc(config-router)# network 192.168.23.0/30 area 0
frr-pc(config-router)# network 205.165.200.224/30 area 0
frr-pc(config-router)# end
frr-pc#

```

Figure 29. Configuring OSPF in router r2.

Consider the above figure. After configuring OSPF in router r2, the interfaces *r2-eth0*, *r2-eth1*, and *r2-eth2* will participate in the routing process of area 0.

**Step 12.** Router r3 is configured similarly to router r2 but with different IP addresses (see Table 2). Those steps are summarized in the following figure. To proceed, in router r3 terminal, issue the commands depicted below.

```

Host: r3
frr-pc# exit
root@frr-pc:/etc/routers/r3# ospfd
root@frr-pc:/etc/routers/r3# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 3.3.3.3
frr-pc(config-router)# network 192.168.3.0/24 area 0
frr-pc(config-router)# network 192.168.13.0/30 area 0
frr-pc(config-router)# network 192.168.23.0/30 area 0
frr-pc(config-router)# end
frr-pc#

```

Figure 30. Configuring OSPF in router r3.

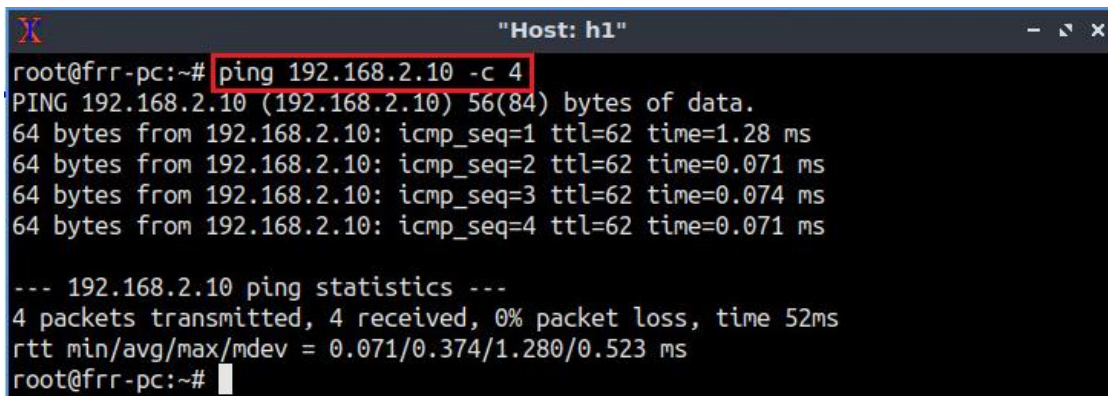
Consider the above figure. After configuring OSPF in router r3, the interfaces *r3-eth0*, *r3-eth1*, and *r3-eth2* will participate in the routing process of area 0.

### 3.2 Verifying Connectivity

After configuring single-area OSPF in routers r1, r2, and r3, they start exchanging OSPF packets until they converge, (i.e., all the three routers have the same topology table, link state database) and thus, the hosts h1, h2, and h3 can ping each other.

**Step 1.** In host h1 terminal, type the following command to test the connectivity between host h1 and host h2.

```
ping 192.168.2.10 -c 4
```

A terminal window titled "Host: h1" showing a successful ping command. The command "ping 192.168.2.10 -c 4" is entered and highlighted with a red box. The output shows four successful ping attempts with varying response times. The statistics section indicates 4 packets transmitted, 4 received, and 0% packet loss.

```
root@frr-pc:~# ping 192.168.2.10 -c 4
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=62 time=1.28 ms
64 bytes from 192.168.2.10: icmp_seq=2 ttl=62 time=0.071 ms
64 bytes from 192.168.2.10: icmp_seq=3 ttl=62 time=0.074 ms
64 bytes from 192.168.2.10: icmp_seq=4 ttl=62 time=0.071 ms

--- 192.168.2.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 52ms
rtt min/avg/max/mdev = 0.071/0.374/1.280/0.523 ms
root@frr-pc:~#
```

Figure 31. Testing connectivity between hosts h1 and h2.

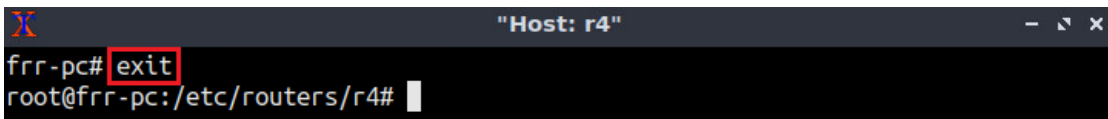
**Step 2.** Similarly, repeat step 1 in hosts h2 and h3 to test connectivity with each other. Host h4 will not be reachable since there is not route to the host.

## 4 Configuring a static route

In this section, you will configure static routing in router r4, which is the external network, so that in later stages the internal network (r1, r2, and r3) running OSPF can access the Internet.

**Step 1.** In router r4 terminal, type the following command to exit from the router's command line interface.

```
exit
```

A terminal window titled "Host: r4" showing the "exit" command being entered and highlighted with a red box. The prompt changes from "frr-pc#" to "root@frr-pc:/etc/routers/r4#".

```
frr-pc# exit
root@frr-pc:/etc/routers/r4#
```

Figure 32. Exiting from router r4's command line interface.

**Step 2.** To enable the static routing daemon in router r4, type the following command:

```
staticd
```

```
frr-pc# exit
root@frr-pc:/etc/routers/r4# staticd
root@frr-pc:/etc/routers/r4#
```

Figure 33. Starting the static routing daemon.

**Step 3.** To load the shell for executing the FRR commands, type the following command:

```
vttysh
```

```
frr-pc# exit
root@frr-pc:/etc/routers/r4# staticd
root@frr-pc:/etc/routers/r4# vtysh

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

frr-pc#
```

Figure 34. Starting vtysh in router r4.

**Step 4.** To enter configuration mode in router r4, type the following command:

```
configure terminal
```

```
frr-pc# exit
root@frr-pc:/etc/routers/r4# staticd
root@frr-pc:/etc/routers/r4# vtysh

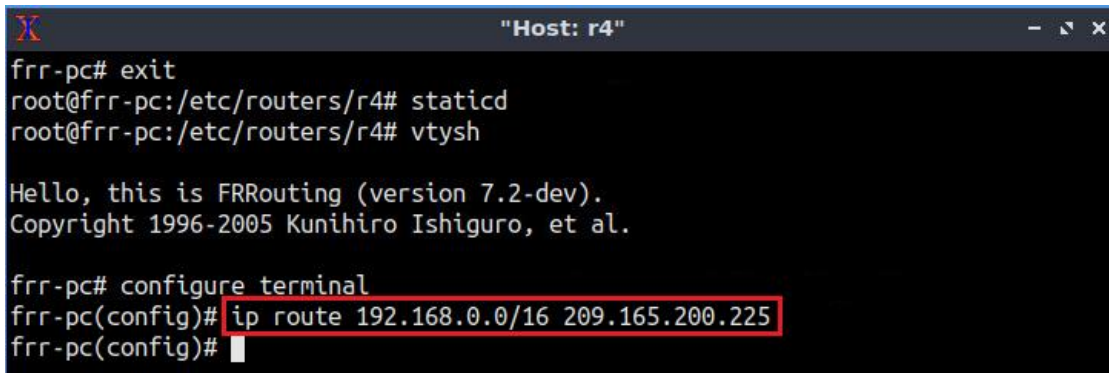
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)#
```

Figure 35. Enabling configuration mode on router r4.

**Step 5.** Router r4 must establish a static route to the internal network, consisting of routers r1, r2, and r3, along with their subnets, so that it can forward traffic to this network. The static route configured on router r4 will aggregate all the subnets within the internal network into one address, i.e., 192.168.0.0/16. To do so, you will use the `ip route` command followed by the address to be matched on, followed by the IP address of the next hop.

```
ip route 192.168.0.0/16 209.165.200.225
```



```

Host: r4
frr-pc# exit
root@frr-pc:/etc/routers/r4# staticd
root@frr-pc:/etc/routers/r4# vtysh

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

frr-pc# configure terminal
frr-pc(config)# ip route 192.168.0.0/16 209.165.200.225
frr-pc(config)#

```

Figure 36. Configuring static route on router r4.

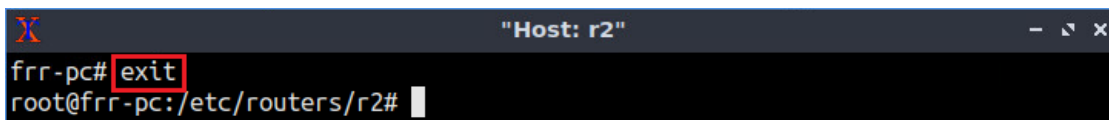
Consider the figure above. The network 192.168.0.0/16 is the network address that aggregates all the subnets in the internal network, and 209.165.200.225 is the IP address of the interface *r2-eth3* (next hop).

## 5 Propagating default routes in OSPF

In this section, you will establish this connectivity by configuring the static routes in router r2 and assigning a default gateway to the Internet. After that, you will propagate the default gateway from router r2 to routers r1 and r3 using OSPF protocol so that the internal network can ping the webserver.

### 5.1 Configuring a static route in router r2

**Step 1.** In router r2 terminal, type the following command to exit from the router's command line interface.



```

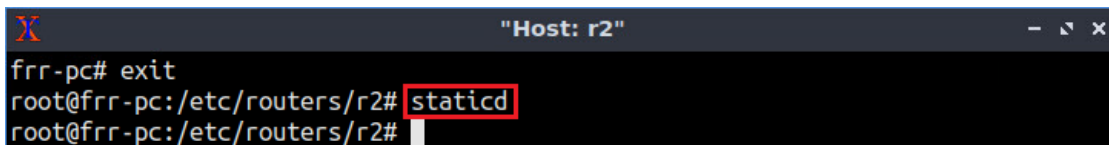
Host: r2
frr-pc# exit
root@frr-pc:/etc/routers/r2#

```

Figure 37. Exiting from router r2's command line interface.

**Step 2.** To enable static routing in router r2, type the following command:

```
staticd
```



```

Host: r2
frr-pc# exit
root@frr-pc:/etc/routers/r2# staticd
root@frr-pc:/etc/routers/r2#

```

Figure 38. Starting the static routing daemon.

**Step 3.** To load the shell for executing the FRR commands, type the following command:

```
vtysh
```

```
frr-pc# exit
root@frr-pc:/etc/routers/r2# staticd
root@frr-pc:/etc/routers/r2# vtysh

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

frr-pc#
```

Figure 39. Starting vtysh in router r2.

**Step 4.** To enter configuration mode in router r2, issue the following command:

```
configure terminal
```

```
frr-pc# exit
root@frr-pc:/etc/routers/r2# staticd
root@frr-pc:/etc/routers/r2# vtysh

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

frr-pc# configure terminal
frr-pc(config)#
```

Figure 40. Enabling configuration mode in router r2.

**Step 5.** To configure a default route on r2 to the Internet, issue the following command. This will inform router r2 to forward any unknown IP address to the interface assigned 209.165.200.226, i.e., interface *r2-eth3*. Therefore, router r2 will forward any unknown traffic to the ISP.

```
ip route 0.0.0.0/0 209.165.200.226
```

```
frr-pc# exit
root@frr-pc:/etc/routers/r2# staticd
root@frr-pc:/etc/routers/r2# vtysh

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

frr-pc# configure terminal
frr-pc(config)# ip route 0.0.0.0/0 209.165.200.226
frr-pc(config)#
```

Figure 41. Configuring static route in router r2.

**Step 6.** To exit configuration mode, type the following command:

```
end
```



```

frr-pc# exit
root@frr-pc:/etc/routers/r2# staticd
root@frr-pc:/etc/routers/r2# vtysh

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

frr-pc# configure terminal
frr-pc(config)# ip route 0.0.0.0/0 209.165.200.226
frr-pc(config)# end
frr-pc#

```

Figure 42. Exiting configuration mode.

**Step 7.** To verify that the default route has been added to router r2 routing table, type the following command:

```
show ip route
```

```

frr-pc# 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

S>* 0.0.0.0/0 [1/0] via 209.165.200.226, r2-eth3, 00:18:15
O>* 192.168.1.0/24 [110/20] via 192.168.12.1, r2-eth1, 04:16:01
O 192.168.2.0/24 [110/10] is directly connected, r2-eth0, 04:16:23
C>* 192.168.2.0/24 is directly connected, r2-eth0, 04:16:48
O>* 192.168.3.0/24 [110/20] via 192.168.23.2, r2-eth2, 04:12:45
O 192.168.12.0/30 [110/10] is directly connected, r2-eth1, 04:16:11
C>* 192.168.12.0/30 is directly connected, r2-eth1, 04:16:48
O>* 192.168.13.0/30 [110/20] via 192.168.12.1, r2-eth1, 04:12:45
   * via 192.168.23.2, r2-eth2, 04:12:45
O 192.168.23.0/30 [110/10] is directly connected, r2-eth2, 04:16:00
C>* 192.168.23.0/30 is directly connected, r2-eth2, 04:16:48
O 209.165.200.224/30 [110/10] is directly connected, r2-eth3, 04:15:33
C>* 209.165.200.224/30 is directly connected, r2-eth3, 04:16:48
frr-pc#

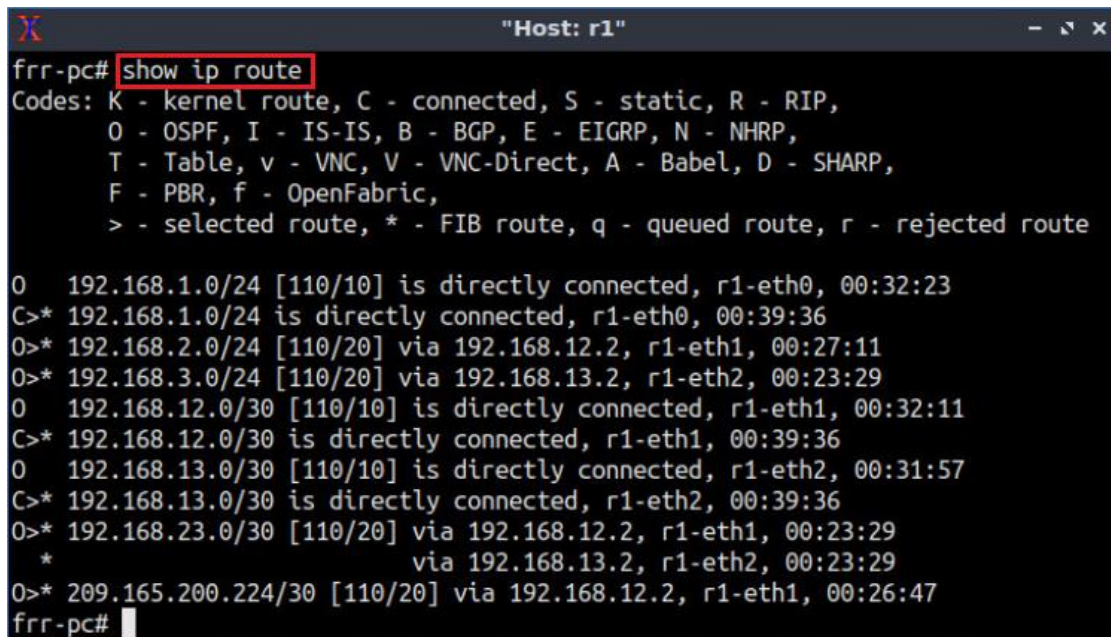
```

Figure 43. Displaying the routing table of router r2.

## 5.2 Propagating the default route to routers r1 and r3 in OSPF

**Step 1.** In router r1 terminal, type the following command to see the routing table.

```
show ip route
```



```

Host: r1
frr-pc# 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

O 192.168.1.0/24 [110/10] is directly connected, r1-eth0, 00:32:23
C>* 192.168.1.0/24 is directly connected, r1-eth0, 00:39:36
O>* 192.168.2.0/24 [110/20] via 192.168.12.2, r1-eth1, 00:27:11
O>* 192.168.3.0/24 [110/20] via 192.168.13.2, r1-eth2, 00:23:29
O 192.168.12.0/30 [110/10] is directly connected, r1-eth1, 00:32:11
C>* 192.168.12.0/30 is directly connected, r1-eth1, 00:39:36
O 192.168.13.0/30 [110/10] is directly connected, r1-eth2, 00:31:57
C>* 192.168.13.0/30 is directly connected, r1-eth2, 00:39:36
O>* 192.168.23.0/30 [110/20] via 192.168.12.2, r1-eth1, 00:23:29
    * via 192.168.13.2, r1-eth2, 00:23:29
O>* 209.165.200.224/30 [110/20] via 192.168.12.2, r1-eth1, 00:26:47
frr-pc#

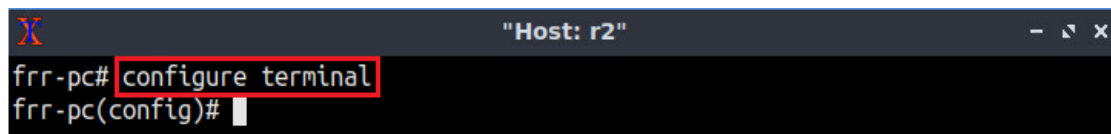
```

Figure 44. Displaying the routing table of router r1.

Consider the figure above. The routing table of router r1 does not contain the default route as configured in router r2. When the default route is configured in one router, the route has to be propagated to the other routers in the OSPF area so that the default route may be added to the routing table.

**Step 2.** In the previous subsection, you configured a default route in router r2. However, this route was not propagated in OSPF. To advertise this route in OSPF protocol to other routers running OSPF in the same area first start by executing the following command in router r2's terminal.

```
configure terminal
```



```

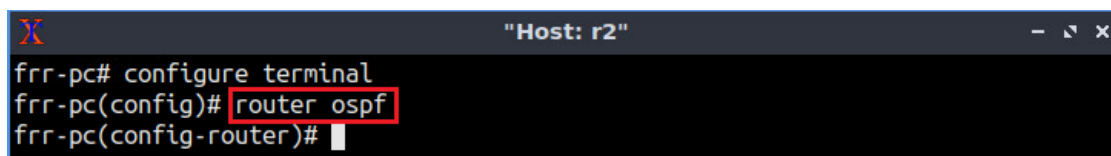
Host: r2
frr-pc# configure terminal
frr-pc(config)#

```

Figure 45. Enabling configuration mode on router r2.

**Step 3.** To enable OSPF configuration mode, write the below command.

```
router ospf
```



```

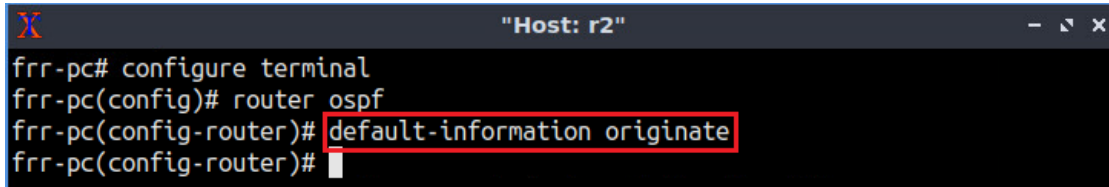
Host: r2
frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)#

```

Figure 46. Enabling OSPF configuration mode in router r2.

**Step 4.** To propagate the default route, write the below command.

```
default-information originate
```



```

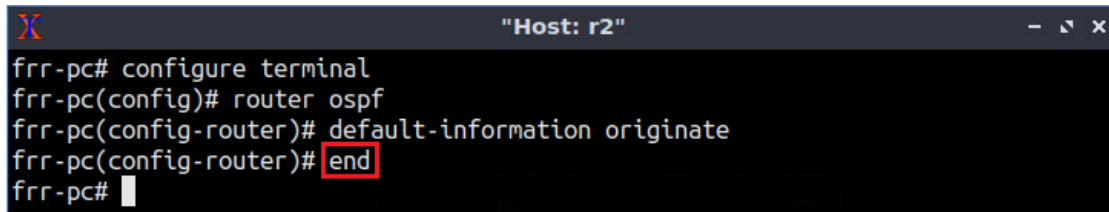
frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# default-information originate
frr-pc(config-router)#

```

Figure 47. Propagating the default route in OSPF.

**Step 5.** To exit OSPF configuration mode type the following command:

```
end
```



```

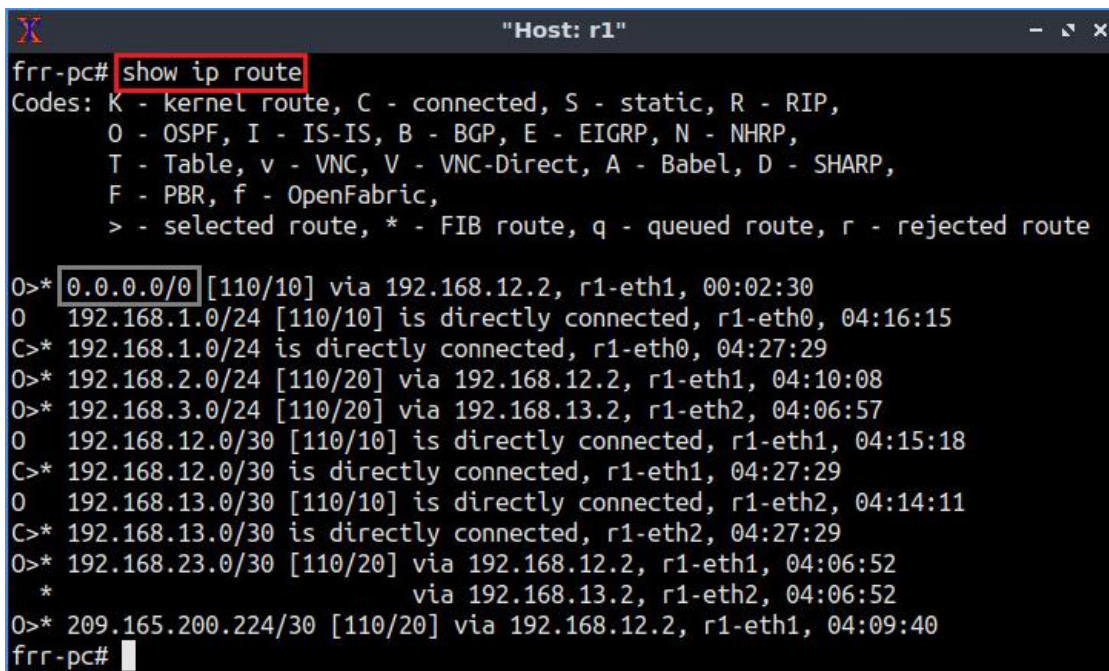
frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# default-information originate
frr-pc(config-router)# end
frr-pc#

```

Figure 48. Exiting OSPF configuration mode in router r2.

**Step 6.** In router r1 terminal, issue the following command to view the routing table of router r1.

```
show ip route
```



```

frr-pc# 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

O>* 0.0.0.0/0 [110/10] via 192.168.12.2, r1-eth1, 00:02:30
O   192.168.1.0/24 [110/10] is directly connected, r1-eth0, 04:16:15
C>* 192.168.1.0/24 is directly connected, r1-eth0, 04:27:29
O>* 192.168.2.0/24 [110/20] via 192.168.12.2, r1-eth1, 04:10:08
O>* 192.168.3.0/24 [110/20] via 192.168.13.2, r1-eth2, 04:06:57
O   192.168.12.0/30 [110/10] is directly connected, r1-eth1, 04:15:18
C>* 192.168.12.0/30 is directly connected, r1-eth1, 04:27:29
O   192.168.13.0/30 [110/10] is directly connected, r1-eth2, 04:14:11
C>* 192.168.13.0/30 is directly connected, r1-eth2, 04:27:29
O>* 192.168.23.0/30 [110/20] via 192.168.12.2, r1-eth1, 04:06:52
   *                via 192.168.13.2, r1-eth2, 04:06:52
O>* 209.165.200.224/30 [110/20] via 192.168.12.2, r1-eth1, 04:09:40
frr-pc#

```

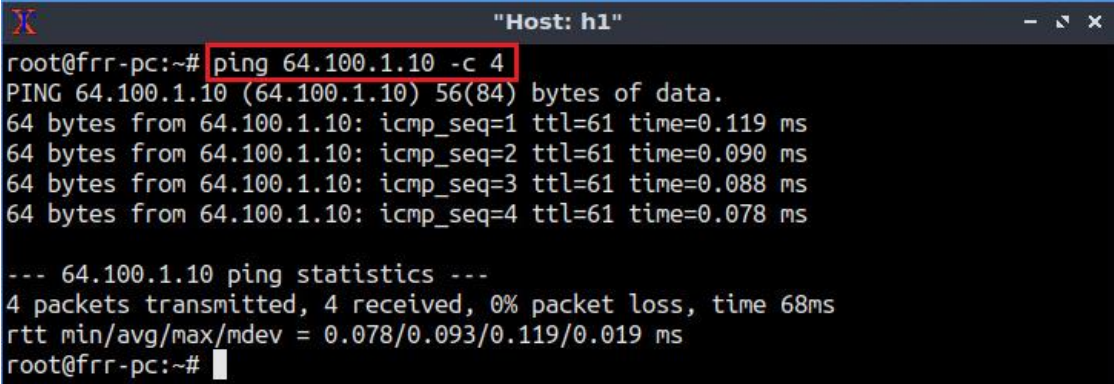
Figure 49. Displaying the routing table of router r1.

### 5.3 Verifying Connectivity

After configuring OSPF and static routes in the routers, you will verify if the configuration was applied correctly by testing the connectivity between the hosts in the topology.

**Step 1.** Test connectivity between hosts h1 and h4 using the `ping` command. In host h1's terminal, type the commands specified below to ping from h1 to h4. The figure below shows a successful connectivity test.

```
ping 64.100.1.10 -c 4
```



```
root@frr-pc:~# ping 64.100.1.10 -c 4
PING 64.100.1.10 (64.100.1.10) 56(84) bytes of data.
64 bytes from 64.100.1.10: icmp_seq=1 ttl=61 time=0.119 ms
64 bytes from 64.100.1.10: icmp_seq=2 ttl=61 time=0.090 ms
64 bytes from 64.100.1.10: icmp_seq=3 ttl=61 time=0.088 ms
64 bytes from 64.100.1.10: icmp_seq=4 ttl=61 time=0.078 ms

--- 64.100.1.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 68ms
rtt min/avg/max/mdev = 0.078/0.093/0.119/0.019 ms
root@frr-pc:~#
```

Figure 50. Connectivity test between hosts h1 and h2.

**Step 2.** Similarly, perform a connectivity test from hosts h2 and h3 to the destination IP address 64.100.1.10. Results will show a successful connectivity test.

This concludes lab 5. Stop the emulation and then exit out of MiniEdit.

## References

1. Moy, J. "RFC 1247: OSPF version 2." 1991.
2. Moy, J. "RFC 2328: OSPF version 2." 1998.
3. Coltun, R., Ferguson, D., Moy, J., Lindem, A. "RFC 5340, OSPF for IPv6." 2008.
4. Peterson, L. "Computer networks: a systems approach." 6th Edition, 2021.
5. Tanenbaum, D. Wetherall. "Computer networks." 5th Edition. 2012.
6. J. Kurose, K. Ross. "Computer networking, a top-down approach." 7th Edition. 2017.
7. Teare, D., Vachon, B., Graziani, R. "Implementing Cisco IP routing (ROUTE) foundation learning guide: (CCNP ROUTE 300-101)." 2014.
8. Linux foundation collaborative projects. "FRR routing documentation." 2017 [Online] Available: <http://docs.frrouting.org/en/latest/>.



UNIVERSITY OF  
**SOUTH CAROLINA**

## OPEN SHORTEST PATH FIRST

### Lab 6: OSPFv2 Virtual Link

Document Version: **10-22-2021**



Award 1829698

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

## Contents

Overview .....	3
Objectives.....	3
Lab settings .....	3
Lab roadmap .....	3
1 Introduction .....	3
1.1 OSPF virtual link .....	4
1.2 OSPF virtual link use cases .....	4
1.3 Design considerations and limitations of OSPF virtual links .....	5
2 Lab topology.....	5
2.1 Lab settings.....	6
2.2 Loading the topology.....	7
2.3 Load the zebra daemon and verify the configuration .....	10
3 Configuring multi-area OSPF.....	14
3.1 Configuring OSPF in routers r1, r2, and r3 .....	14
3.2 Verifying OSPF neighbors, routing information, and connectivity. ....	19
4 Configuring OSPF virtual link .....	21
4.1 Verifying OSPF neighbors, routing information, and connectivity. ....	22
References .....	24

## Overview

This lab introduces the concept of a virtual link, a feature of the Open Shortest Path First (OSPF) protocol employed to link an area that is not contiguous to the backbone area (i.e., area 0). The OSPF protocol requires a direct connection of the nonbackbone areas to the backbone area. In this way, the OSPF protocol redistributes the routes received from the nonbackbone areas. This lab explains how to configure a virtual link on routers participating in the OSPF process.

## Objectives

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

1. Understand the concept of OSPF virtual link.
2. Configure multi-area OSPF.
3. Identify the scenarios where a virtual link can be employed.
4. Configure OSPF virtual link to connect a non-contiguous area to the backbone area.
5. Understand the limitations of using a virtual link.

## Lab settings

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

Table 1. Credentials to access the Client 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 multi-area OSPF.
4. Section 4: Configuring OSPF virtual link.

### 1 Introduction

## 1.1 OSPF virtual link

The OSPF protocol requires a backbone area (i.e., area 0) when more than one area is configured. All the nonbackbone areas must be directly connected to the backbone area. In this way, the nonbackbone areas inject their routes into the backbone area, and the backbone area can redistribute these routes to the other areas. However, there are scenarios where nonbackbone areas are not contiguous to the backbone area, meaning that the OSPF protocol will not exchange routing information with the non-contiguous areas. Nevertheless, the OSPF protocol provides the virtual link feature, allowing non-contiguous areas to connect to the backbone area across a transit area.

## 1.2 OSPF virtual link use cases

Virtual links are used in two cases:

1. Linking a nonbackbone area that cannot be physically connected to the backbone area.
2. Connecting partitioned parts of the backbone area.

Consider Figure 1. In this scenario, the backbone area (i.e., area 0) is divided in two parts. A virtual link can be configured in routers r2 and r4, to create a logical connection between the two parts of the backbone across a transit area (i.e., area 1). Therefore, the routers can participate in the OSPF process.

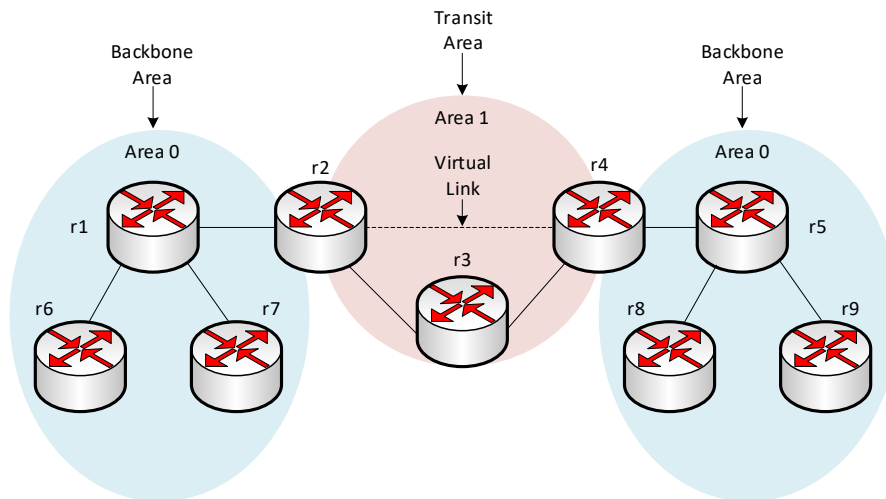


Figure 1. OSPF virtual link use case. A virtual link can be used in a scenario where the backbone area is separated by a transit area.

Consider Figure 2. In this scenario, area 2 is not contiguous to the backbone area. Thus, configuring a virtual link between routers r1 and r3 will provide a logical path to area 0. In this topology, area 1 is used as a transit area allowing area 2 to participate in the OSPF process. A transit network can carry data traffic that is neither locally originated nor locally destined<sup>12</sup>.



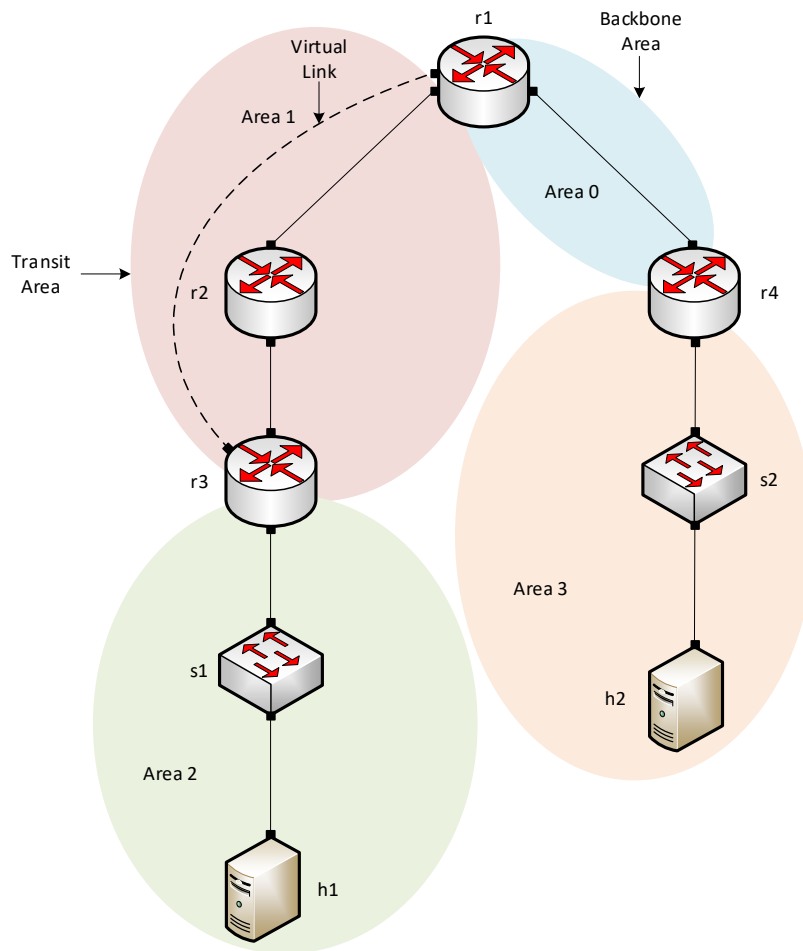


Figure 2. OSPF virtual link use case. In this scenario, a virtual link is employed to connect a nonbackbone area (i.e., area 2) to the backbone area (i.e., area 0) via a transit area (i.e., area 1).

### 1.3 Design considerations and limitations of OSPF virtual links

OSPF virtual links should be used only for temporary connections or as a backup connection. When designing a network, it is not a good practice to use virtual links as a primary resource, specifically when the virtual link connects two parts of the backbone area (see Figure 1). This consideration leverages the fact that the OSPF process will be subject to any failure in the transit area. Moreover, a virtual link cannot traverse more than one area, and the transit area cannot be a stub area. In a stub area OSPF Hello packets are not flooded to optimize the size of the link-state database. If there is a need to connect an area by crossing two nonbackbone areas, two virtual links must be configured in each area.

## 2 Lab topology

Consider Figure 3. The topology comprises four OSPF areas, where areas 1 and 3 are physically connected to the backbone area (i.e., area 0). Notice that there is no direct connection from area 2 to the backbone area. The following sections show the steps to configure routers r3 and r1 to establish a virtual link that will connect area 2 to the backbone using area 1 as a transit area.

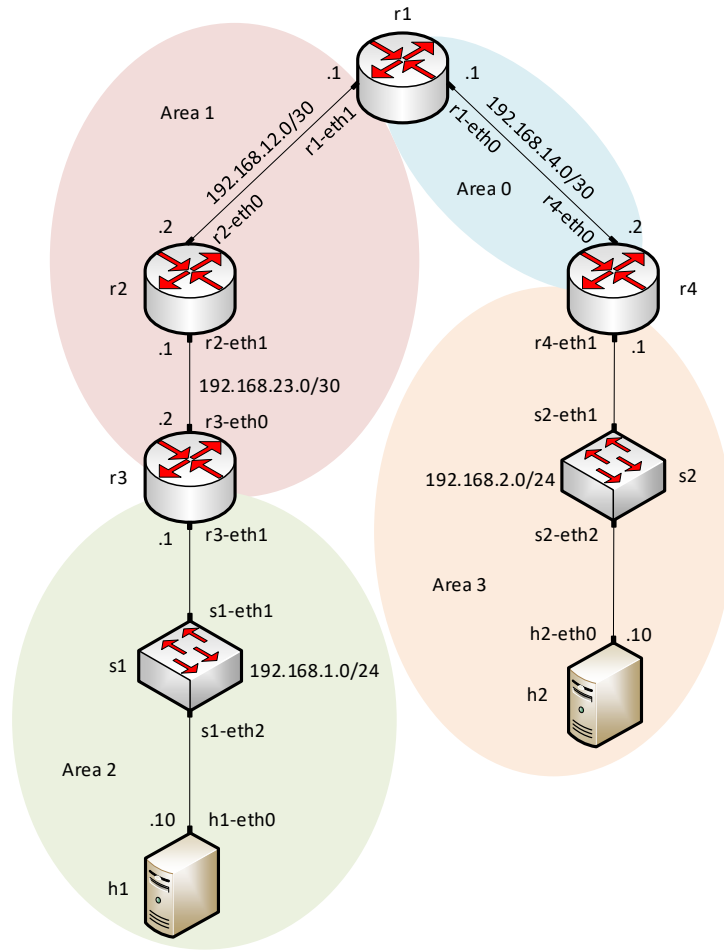


Figure 3. Lab topology.

## 2.1 Lab settings

Table 2 contains information about the devices and their corresponding interfaces, IPv4 addresses, subnet, and default gateway.

Table 2. Topology Information.

Device	Interface	IPv4 Address	Subnet	Default Gateway
	r1-eth0	192.168.14.1	/30	N/A

Router 1 (r1)	r1-eth1	192.168.12.1	/30	N/A
Router 2 (r2)	r2-eth0	192.168.12.2	/30	N/A
	r2-eth1	192.168.23.1	/30	N/A
Router 3 (r1)	r3-eth0	192.168.23.2	/30	N/A
	r3-eth1	192.168.1.1	/24	N/A
Router 4 (r3)	r4-eth0	192.168.14.2	/30	N/A
	r4-eth1	192.168.2.1	/24	N/A
Host 1 (h1)	h1-eth0	192.168.1.10	/24	192.168.1.1
Host 2 (h2)	h2-eth0	192.168.2.10	/24	192.168.2.1

## 2.2 Loading the topology

**Step 1.** Launch MiniEdit by double clicking on the shortcut located on the desktop. When prompted for a password, type `password`.



Figure 4. MiniEdit shortcut.

**Step 2.** In MiniEdit's menu bar, click on *File* then *open* to load the lab topology. Locate the *lab6.mn* topology file in the default directory, `/home/frr/OSPF_Labs/lab6` and click on *Open*.

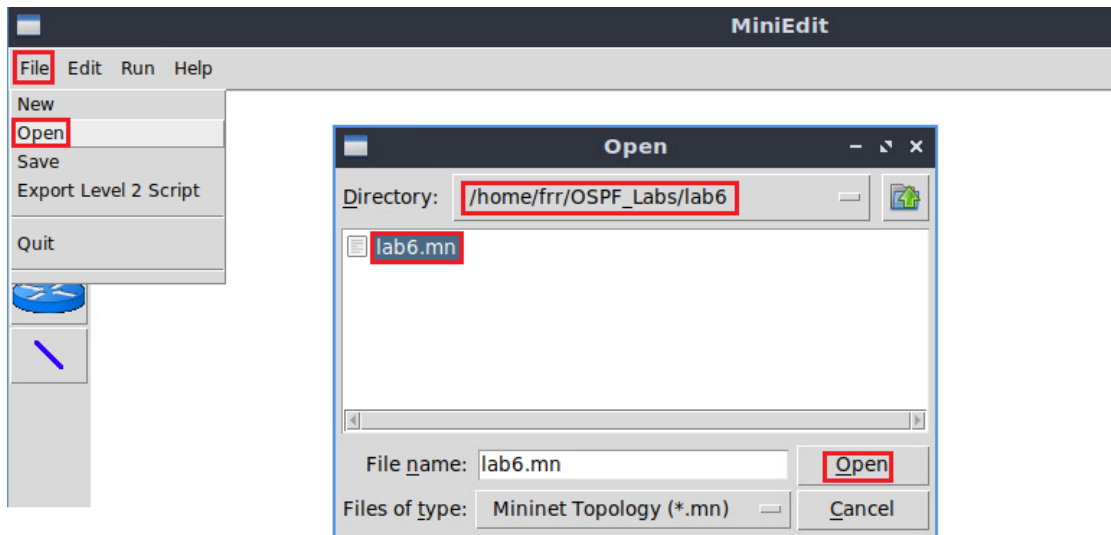


Figure 5. MiniEdit's open dialog.

**Step 3.** Click on the *Run* button located on the lower left-hand side to run the emulation.



Figure 6. Starting the emulation.

At this point, the topology is loaded with all the required network components. Next, you will execute a script that will configure the interfaces of the routers.

Automating the configuration of interfaces with the script is intended to facilitate the lab experience and to focus on the OSPF experiment.

**Step 4.** Open the Linux terminal by clicking on the icon located in the taskbar.

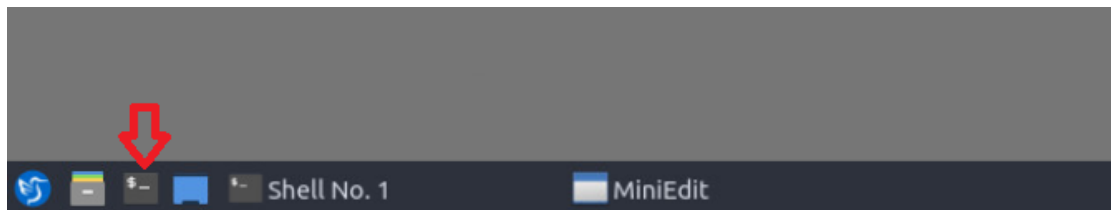
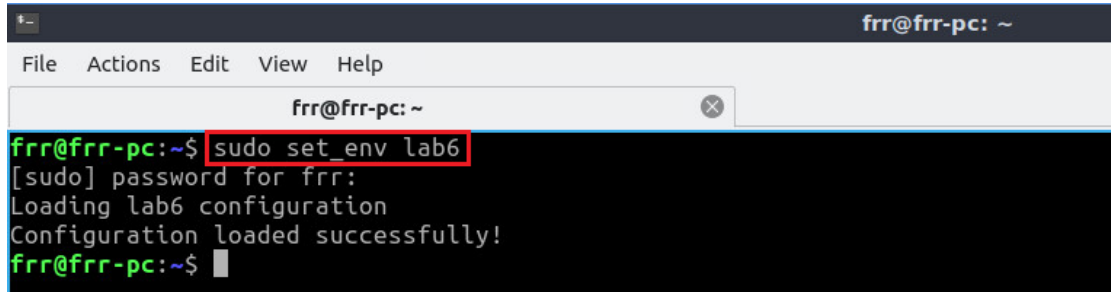


Figure 7. Opening Linux terminal.

**Step 5.** Issue the command shown below to set the lab environment. This command configures the IP addresses of the routers' interface according to Table 2. If a password is required, type `password`.

```
sudo set_env lab6
```

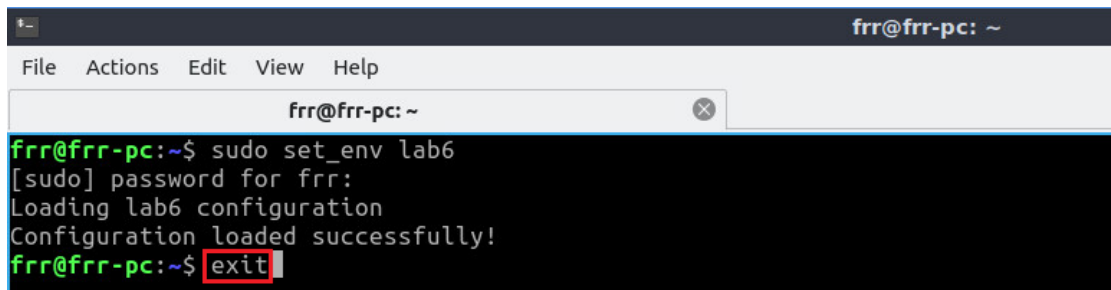
A terminal window titled 'frr@frr-pc: ~' with a menu bar (File, Actions, Edit, View, Help). The terminal shows the command 'sudo set\_env lab6' being entered and executed. The output is: '[sudo] password for frr:', 'Loading lab6 configuration', and 'Configuration loaded successfully!'. The prompt returns to 'frr@frr-pc:~\$'.

```
frr@frr-pc:~$ sudo set_env lab6
[sudo] password for frr:
Loading lab6 configuration
Configuration loaded successfully!
frr@frr-pc:~$
```

Figure 8. Setting the lab environment.

**Step 6.** Type the following command to exit the Linux terminal.

```
exit
```

A terminal window titled 'frr@frr-pc: ~' with a menu bar (File, Actions, Edit, View, Help). The terminal shows the command 'exit' being entered and executed. The output is: '[sudo] password for frr:', 'Loading lab6 configuration', and 'Configuration loaded successfully!'. The prompt returns to 'frr@frr-pc:~\$'.

```
frr@frr-pc:~$ sudo set_env lab6
[sudo] password for frr:
Loading lab6 configuration
Configuration loaded successfully!
frr@frr-pc:~$ exit
```

Figure 9. Exiting the Linux terminal.

**Step 7.** Click on Mininet's terminal, i.e., the one launched when MiniEdit was started.

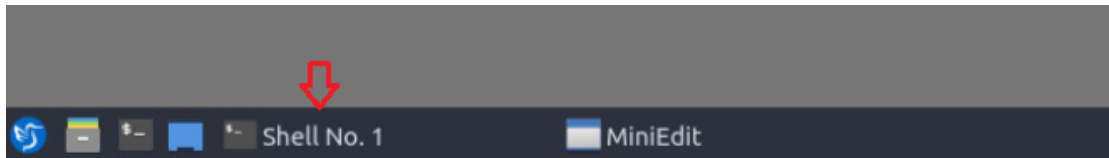


Figure 10. Opening Mininet's terminal.

**Step 8.** Issue the following command to display the interface names and connections.

```
links
```

```

mininet> links
r1-eth0<->r4-eth0 (OK OK)
s1-eth1<->h1-eth0 (OK OK)
r1-eth1<->r2-eth0 (OK OK)
r2-eth1<->r3-eth0 (OK OK)
r3-eth1<->s1-eth2 (OK OK)
s2-eth1<->h2-eth0 (OK OK)
s2-eth2<->r4-eth1 (OK OK)
mininet>
    
```

Figure 11. Displaying network interfaces.

In the figure above, the link displayed within the gray box indicates that interface *eth1* of switch *s1* connects to interface *eth0* of host *h1* (i.e., *s1-eth1<->h1-eth0*).

### 2.3 Load the zebra daemon and verify the configuration

In this section, you will verify that the IP addresses listed in Table 2 and inspect the routing table of the routers.

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

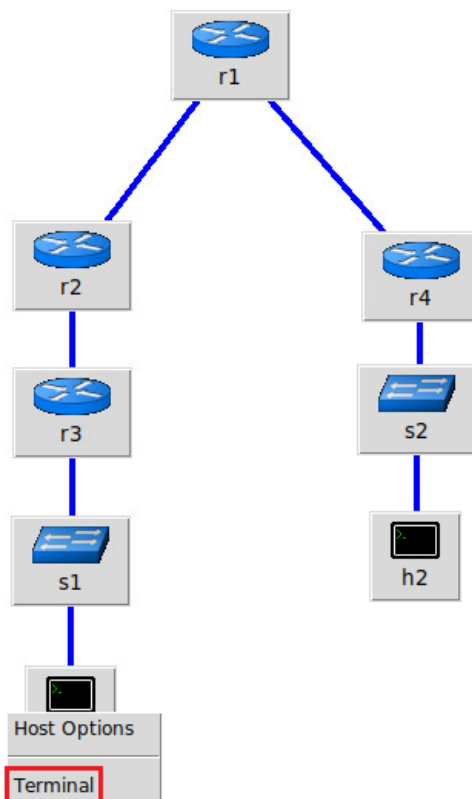
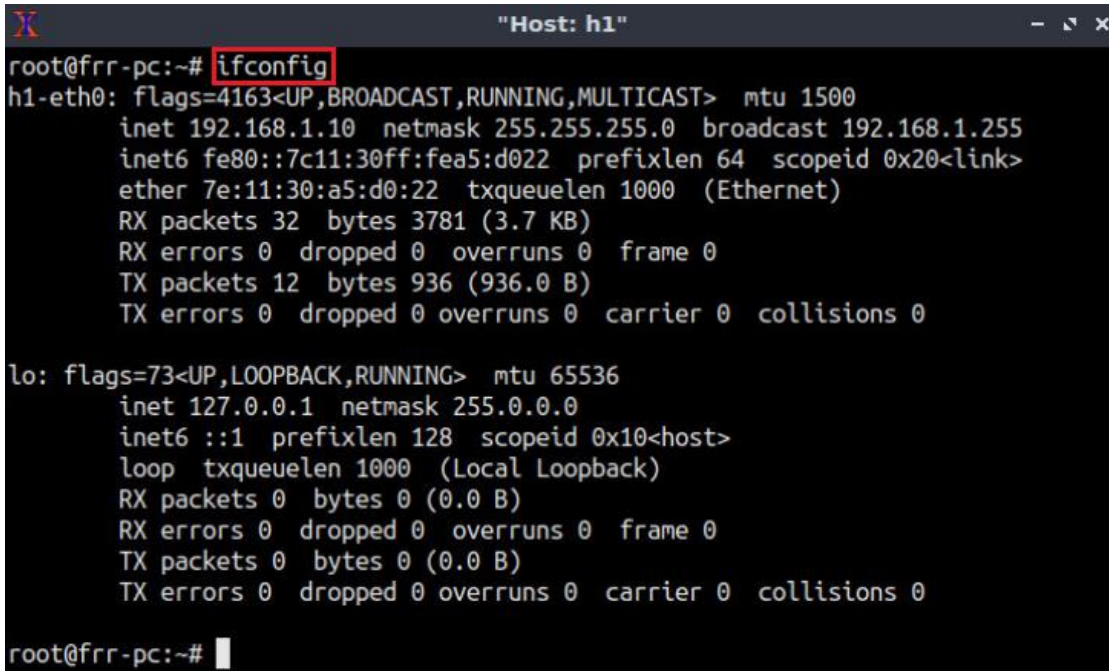


Figure 12. Opening a terminal on host *h1*.

**Step 2.** In the host h1 terminal, type the command shown below to verify that the IP address was assigned successfully. You will confirm that host h1 has two interfaces. Interface *h1-eth0* is configured with the IP address 192.168.1.10 and the subnet mask 255.255.255.0 and a loopback interface.

```
ifconfig
```



```
root@frr-pc:~# ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.10 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::7c11:30ff:fea5:d022 prefixlen 64 scopeid 0x20<link>
    ether 7e:11:30:a5:d0:22 txqueuelen 1000 (Ethernet)
    RX packets 32 bytes 3781 (3.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 12 bytes 936 (936.0 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@frr-pc:~#
```

Figure 13. Displaying host h1 interfaces.

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

```
route
```

```

Host: h1
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
  inet 192.168.1.10 netmask 255.255.255.0 broadcast 192.168.1.255
  inet6 fe80::7c11:30ff:fea5:d022 prefixlen 64 scopeid 0x20<link>
  ether 7e:11:30:a5:d0:22 txqueuelen 1000 (Ethernet)
  RX packets 32 bytes 3781 (3.7 KB)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 12 bytes 936 (936.0 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@frr-pc:~# 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@frr-pc:~#

```

Figure 14. Displaying host h1 default gateway.

**Step 4.** To verify the hosts h2 and h3 default route, proceed by repeating from step 1 to step 3. Similar results should be observed.

**Step 5.** You will validate that the router interfaces are configured according to Table 2. To proceed, hold right-click on router r1 and select *Terminal*.



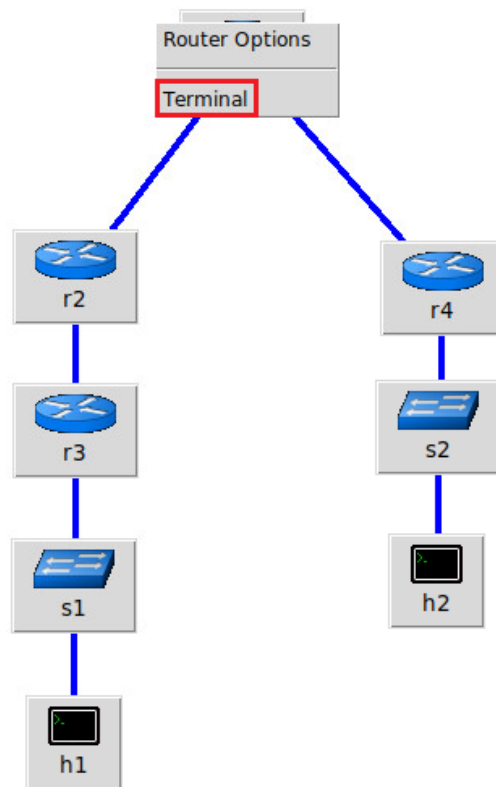


Figure 15. Opening a terminal on router r1.

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

```
zebra
```

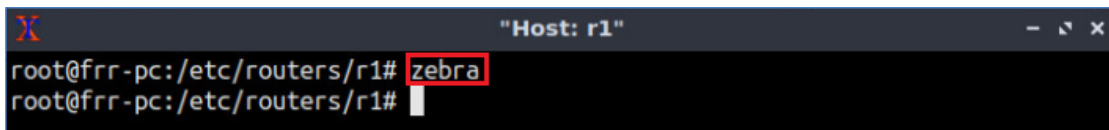


Figure 16. Starting zebra daemon.

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

```
vtys
```

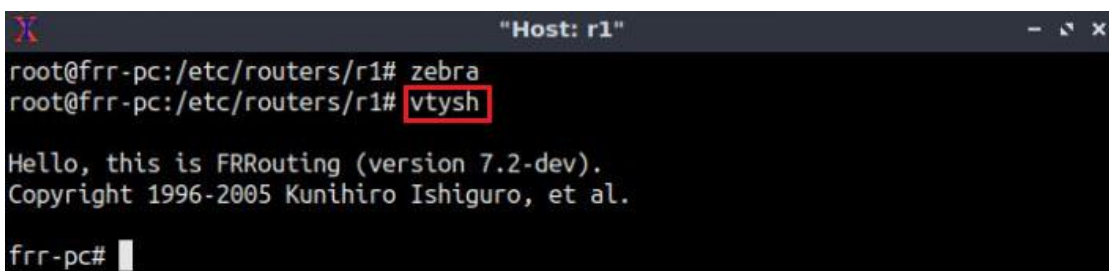


Figure 17. Starting vtysh on router r1.

**Step 8.** Type the following command on router r1 terminal to verify the routing table. It will list all the directly connected networks. Notice that 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
```

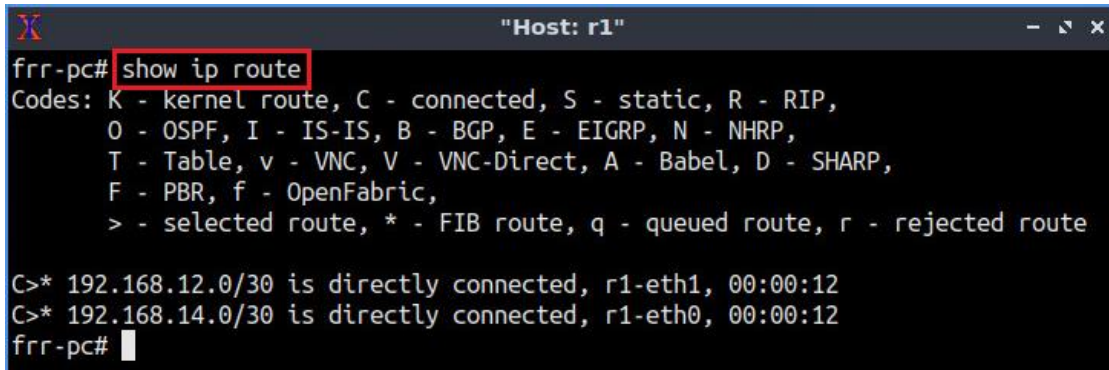


Figure 18. Displaying routing table of router r1.

**Step 9.** To verify the routing table of the other routers, proceed similarly by following from step 5 to step 8 on the corresponding router terminal.

### 3 Configuring multi-area OSPF

In this section, you will configure multi-area OSPF. Router r1 is in the backbone area (i.e., area 0), and routers r2 and r4 are neighbors of router r1 (i.e., areas 1 and 3 are adjacent to area 0). Notice that router r3 is not a neighbor of router r1 (i.e., area 2 is not a neighbor of area 0).

#### 3.1 Configuring OSPF in routers r1, r2, and r3

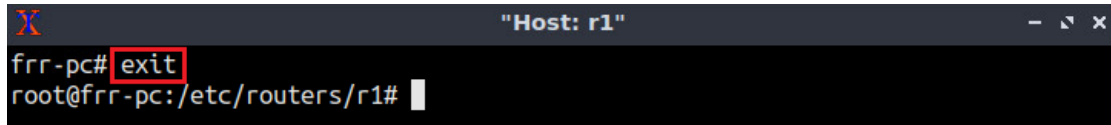
**Step 1.** Navigate to router r1's terminal by selecting the router's r1 terminal from the taskbar.



Figure 19. Opening router r1's terminal.

**Step 2.** In router r1's terminal, type the command below to exit the vtysh session.

```
exit
```



```

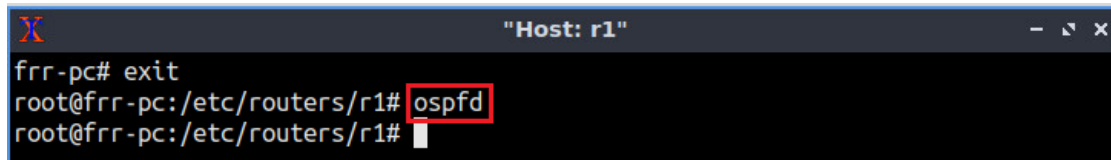
frr-pc# exit
root@frr-pc:/etc/routers/r1#

```

Figure 20. Exiting the vtysh session.

**Step 3.** Enable and start the OSPF routing protocol by typing the following command:

```
ospfd
```



```

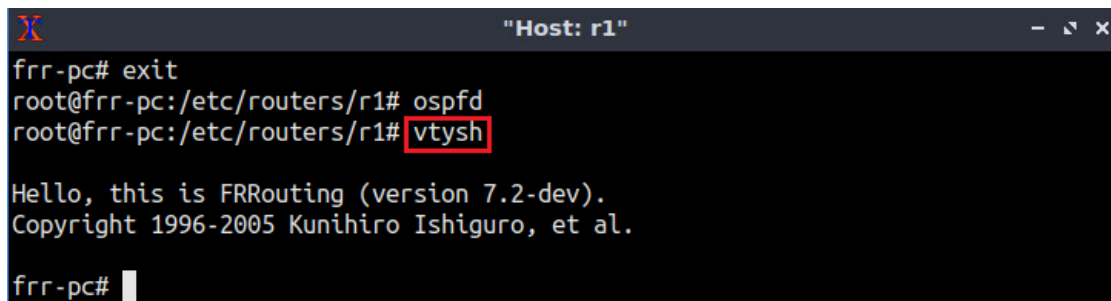
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1#

```

Figure 21. Starting the OSPF daemon.

**Step 4.** In order to enter router r1's terminal, type the following command:

```
vttysh
```



```

frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

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

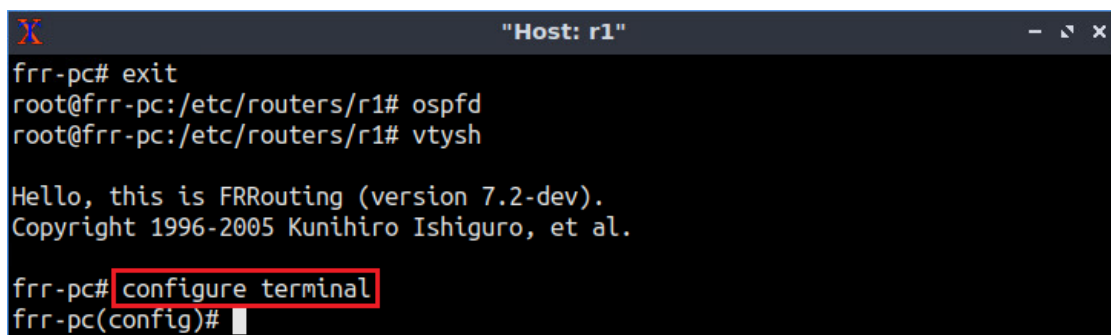
frr-pc#

```

Figure 22. Starting vtysh in router r1.

**Step 5.** At this point, you have entered router r1's terminal, to enter configuration mode in router r1, issue the following command:

```
configure terminal
```



```

frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

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

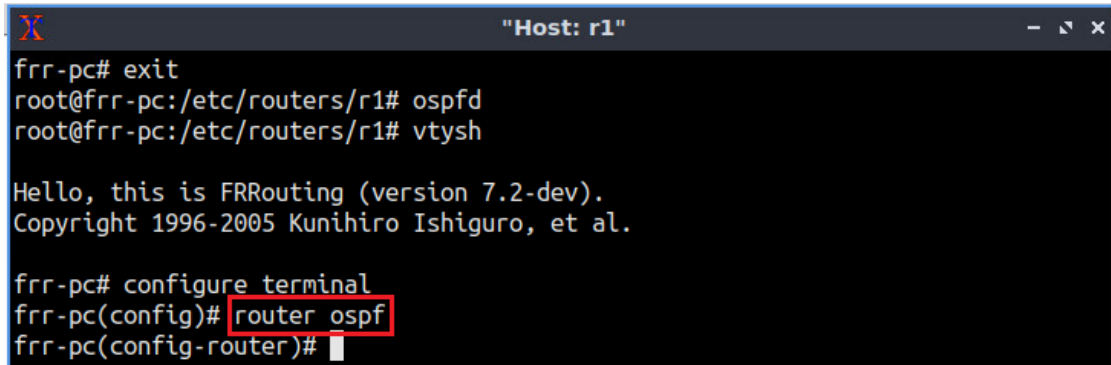
frr-pc# configure terminal
frr-pc(config)#

```

Figure 23. Entering configuration mode in router r1.

**Step 6.** To configure OSPF, type the command shown below. This command will enable OSPF configuration mode.

```
router ospf
```

A terminal window titled "Host: r1" showing the configuration process. The user starts at the root prompt, exits the shell, and enters 'ospfd' and 'vtysh'. After a welcome message, they enter 'configure terminal' and then 'router ospf', which is highlighted with a red box. The prompt changes to 'frr-pc(config-router)#'.

```
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

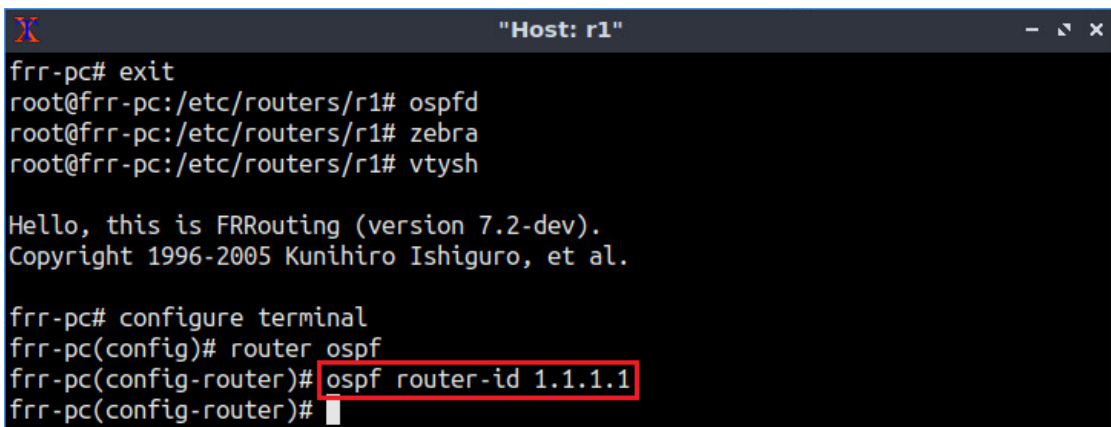
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)#
```

Figure 24. Configuring OSPF in router r1.

**Step 7.** Issue the following command to assign an identification tag to router r1.

```
ospf router-id 1.1.1.1
```

A terminal window titled "Host: r1" showing the configuration process. The user starts at the root prompt, exits the shell, and enters 'ospfd', 'zebra', and 'vtysh'. After a welcome message, they enter 'configure terminal', 'router ospf', and then 'ospf router-id 1.1.1.1', which is highlighted with a red box. The prompt changes to 'frr-pc(config-router)#'.

```
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# zebra
root@frr-pc:/etc/routers/r1# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 1.1.1.1
frr-pc(config-router)#
```

Figure 25. Configuring OSPF router ID in router r1.

**Step 8.** Type the following command to enable the interface *r1-eth1* to participate in the OSPF process in area 1.

```
network 192.168.12.0/30 area 1
```

```
Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# zebra
root@frr-pc:/etc/routers/r1# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 1.1.1.1
frr-pc(config-router)# network 192.168.12.0/30 area 1
frr-pc(config-router)#
```

Figure 26. Enabling interface *r1-eth1* to participate in the OSPF process in area 1.

**Step 9.** Issue the following command to enable the interface *r1-eth0* to participate in the OSPF process in area 1.

```
network 192.168.14.0/30 area 0
```

```
Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# zebra
root@frr-pc:/etc/routers/r1# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 1.1.1.1
frr-pc(config-router)# network 192.168.12.0/30 area 1
frr-pc(config-router)# network 192.168.14.0/30 area 0
frr-pc(config-router)#
```

Figure 27. Enabling interface *r1-eth0* to participate in the OSPF process in area 1.

**Step 10.** Type the following command to exit router configuration mode.

```
end
```

```

Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# zebra
root@frr-pc:/etc/routers/r1# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 1.1.1.1
frr-pc(config-router)# network 192.168.12.0/30 area 1
frr-pc(config-router)# network 192.168.14.0/30 area 0
frr-pc(config-router)# end
frr-pc#

```

Figure 28. Exiting router configuration mode.

Consider the figure above. After configuring OSPF in router r1, the interfaces *r1-eth1*, and *r1-eth0* will participate in the routing process of area 1 and area 0 respectively.

**Step 11.** 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 terminal, issue the commands depicted below.

```

Host: r2
frr-pc# exit
root@frr-pc:/etc/routers/r2# ospfd
root@frr-pc:/etc/routers/r2# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 2.2.2.2
frr-pc(config-router)# network 192.168.12.0/30 area 1
frr-pc(config-router)# network 192.168.23.0/30 area 1
frr-pc(config-router)# end
frr-pc#

```

Figure 29. Router r2 configuration summary.

Consider the figure above. After configuring OSPF in router r2, the interfaces *r2-eth0*, and *r2-eth1* will participate in the routing process of area 1.

**Step 12.** Router r3 is configured similarly to router r2 but with different IP addresses (see Table 2). Those steps are summarized in the following figure. To proceed, in router r3 terminal, issue the commands depicted below.

```

Host: r3
frr-pc# exit
root@frr-pc:/etc/routers/r3# ospfd
root@frr-pc:/etc/routers/r3# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 3.3.3.3
frr-pc(config-router)# network 192.168.23.0/30 area 1
frr-pc(config-router)# network 192.168.1.0/24 area 2
frr-pc(config-router)# end
frr-pc#

```

Figure 30. Router r3 configuration summary.

Consider the figure above. After configuring OSPF in router r3, the interfaces *r3-eth0* and *r3-eth1* will participate in the routing process of area 1 and area 2 respectively.

**Step 13.** Router r4 is configured similarly to router r4 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.

```

Host: r4
frr-pc# exit
root@frr-pc:/etc/routers/r4# ospfd
root@frr-pc:/etc/routers/r4# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 4.4.4.4
frr-pc(config-router)# network 192.168.14.0/30 area 0
frr-pc(config-router)# network 192.168.2.0/24 area 3
frr-pc(config-router)# end
frr-pc#

```

Figure 31. Router r4 configuration summary.

Consider the figure above. After configuring OSPF in router r4, the interfaces *r4-eth0*, and *r4-eth1* will participate in the routing process of area 0 and area 3 respectively.

### 3.2 Verifying OSPF neighbors, routing information, and connectivity.

**Step 1.** In router's r1 terminal, issue the following command to verify router r1's neighbor.

```
show ip ospf neighbor
```

```

frr-pc# show ip ospf neighbor
Neighbor ID   Pri  State           Dead Time Address      Interface      RXmtL  RqstL  DBsmL
2.2.2.2       1    Full/Backup     31.612s  192.168.12.2 r1-eth1:192.168.12.1  0      0      0
4.4.4.4       1    Full/Backup     33.944s  192.168.14.2 r1-eth0:192.168.14.1  0      0      0
frr-pc#

```

Figure 32. Displaying router r1's OSPF neighbors.

The figure above shows OSPF neighbor information as a function of the interfaces. The columns of the table represent the following information.

- Neighbor ID: Indicates the neighbor router ID. The output of the command above shows the neighbors of router r1.
- Priority: Router priority of the neighbor and neighbor state.
- State: The *Full* state indicates that the local router (i.e., router r1) established a full neighbor relationship with the remote OSPF neighbor.
- Dead Time: It shows the elapsed time since the local router (i.e., router r1) received the last OSPF Hello packet.
- Address: Depicts the primary IPv4 address that router uses to participate in the OSPF process.
- Interface: It shows the interface that participates in the OSPF process.

**Step 2.** To display all the routing information in the routing table of router r1, issue the following command:

```
show ip route
```

```

frr-pc# 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


O>* 192.168.2.0/24 [110/20] via 192.168.14.2, r1-eth0, 00:28:08
O   192.168.12.0/30 [110/10] is directly connected, r1-eth1, 00:42:53
C>* 192.168.12.0/30 is directly connected, r1-eth1, 00:59:58
O   192.168.14.0/30 [110/10] is directly connected, r1-eth0, 00:41:32
C>* 192.168.14.0/30 is directly connected, r1-eth0, 00:59:58
O>* 192.168.23.0/30 [110/20] via 192.168.12.2, r1-eth1, 00:33:07
frr-pc#

```

Figure 33. Displaying router r1's routing table.

Consider the figure above. The output displays the networks learned along with additional attributes, such as the routing protocol that discovered the network. For example, networks with the symbol `O` on the left are learned from OSPF, whereas those with symbol `C` are directly connected to the device. Furthermore, the symbol `>` on the left of the displayed network indicates that this entry is selected to reach the network. As for the

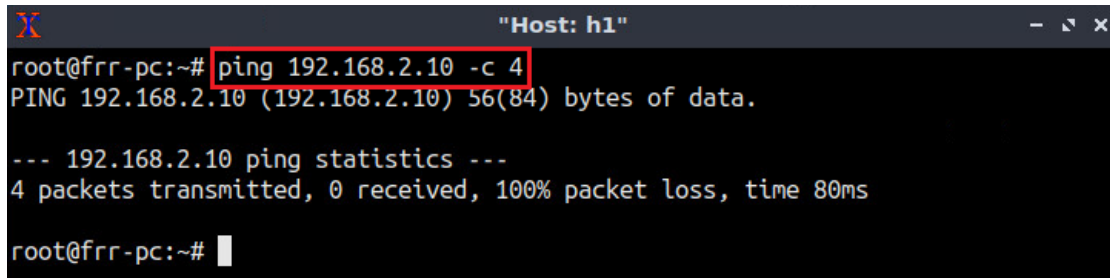


symbol , this indicates that the route is inserted in Forwarding Information Base (FIB) of the router.

Notice that the route to the network 192.168.1.0/24 located in area 2 is missing because the autonomous system is not physically connected to the backbone area (i.e., area 0).

**Step 3.** In the host h1 terminal, issue the following command to test the connectivity between host h1 and host h2. You will notice an unsuccessful connectivity test.

```
ping 192.168.2.10 -c 4
```



```

X "Host: h1"
root@frr-pc:~# ping 192.168.2.10 -c 4
PING 192.168.2.10 (192.168.2.10) 56(84) bytes of data.

--- 192.168.2.10 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 80ms

root@frr-pc:~#

```

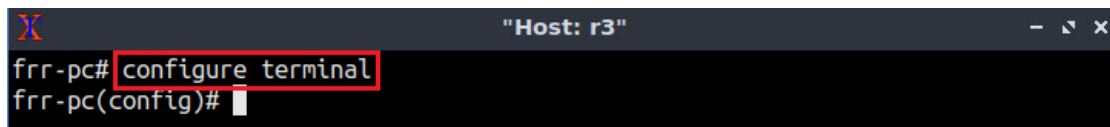
Figure 34. Performing a connectivity test from host h1 to host h2.

## 4 Configuring OSPF virtual link

In the previous section, you configured a multi-area OSPF where area 1 and area 3 are adjacent to the backbone area (i.e., area 0). However, the networks in area 2 are not learned via OSPF because area 2 is not physically connected to area 0. To solve this, you will configure a virtual link from area 2 to area 0 using area 1 as a transit area. In this way area 2 will be able to learn and exchange routes via OSPF.

**Step 1.** In router r3, issue the following command to enter configuration mode.

```
configure terminal
```



```

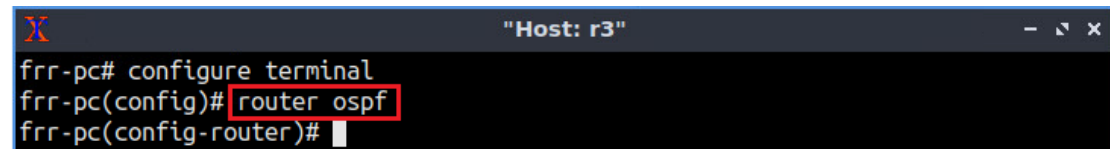
X "Host: r3"
frr-pc# configure terminal
frr-pc(config)#

```

Figure 35. Entering configuration mode in router r3.

**Step 2.** In order to configure OSPF, type the command shown below. This command will enable OSPF configuration mode.

```
router ospf
```



```

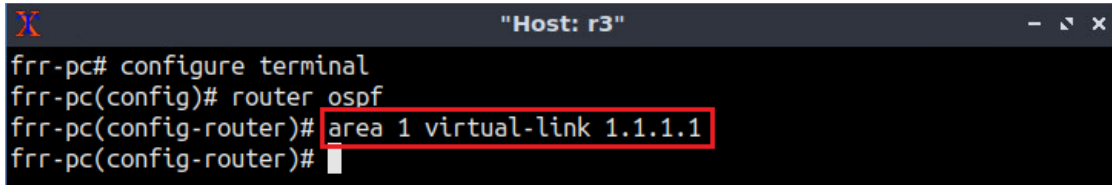
X "Host: r3"
frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)#

```

Figure 36. Configuring OSPF in router r3.

**Step 3.** Issue the following command to configure a virtual link between router r1 and router r3.

```
area 1 virtual-link 1.1.1.1
```



```

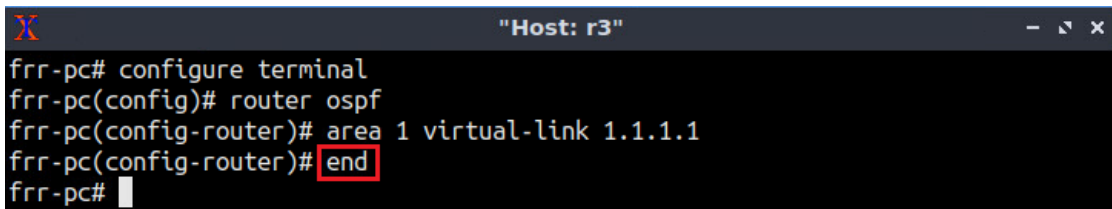
Host: r3
frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# area 1 virtual-link 1.1.1.1
frr-pc(config-router)#
  
```

Figure 37. Configuring a virtual link in router r3.

At this point, routers r3 (with Neighbor ID 3.3.3.3) and r1 (with Neighbor ID 1.1.1.1) know how to reach each other through a transit area (i.e., area 1). Therefore, they can form adjacency across the virtual link.

**Step 4.** Type the following command to exit configuration mode.

```
end
```

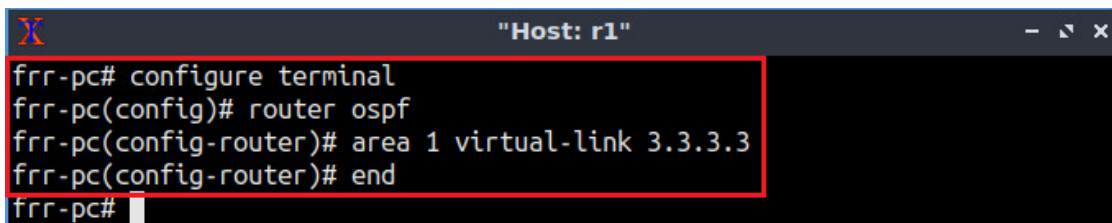


```

Host: r3
frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# area 1 virtual-link 1.1.1.1
frr-pc(config-router)# end
frr-pc#
  
```

Figure 38. Exiting configuration mode in router r3.

**Step 5.** Router r1 is configured similarly. Follow the steps summarized below to configure the virtual link on the other end.



```

Host: r1
frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# area 1 virtual-link 3.3.3.3
frr-pc(config-router)# end
frr-pc#
  
```

Figure 39. Router r1 configuration summary.

#### 4.1 Verifying OSPF neighbors, routing information, and connectivity.

**Step 1.** In router's r1 terminal, issue the following command to verify router r1's neighbor.

```
show ip ospf neighbor
```

```

Host: r1
frr-pc# show ip ospf neighbor
Neighbor ID      Pri State           Dead Time Address      Interface      RXmtL RqstL DBsmL
2.2.2.2          1 Full/Backup      34.763s 192.168.12.2   r1-eth1:192.168.12.1  0    0    0
4.4.4.4          1 Full/Backup      37.070s 192.168.14.2   r1-eth0:192.168.14.1  0    0    0
3.3.3.3          1 Full/DROther     35.399s 192.168.23.2   VLINK0           0    0    0
frr-pc#

```

Figure 40. Displaying router r1's OSPF neighbors.

The figure above shows OSPF neighbor information as a function of the interfaces. The columns of the table represent the following information.

- Neighbor ID: Indicates the neighbor router ID. The output of the command above shows the neighbors of router r1.
- Priority: Router priority of the neighbor and neighbor state.
- State: The *Full* state indicates that the local router (i.e., router r1) established a full neighbor relationship with the remote OSPF neighbor.
- Dead Time: It shows the elapsed time since the local router (i.e., router r1) received the last OSPF Hello packet.
- Address: Depicts the primary IPv4 address that router uses to participate in the OSPF process.
- Interface: It shows the interface that participates in the OSPF process.

Notice that router r1 (neighbor ID 1.1.1.1) and router r3 (neighbor ID 3.3.3.3) are connected through the virtual link *VLINK0*.

**Step 2.** To display all the routing information in the routing table of router r1, issue the following command:

```
show ip route
```

```

Host: r1
frr-pc# 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

O>* 192.168.1.0/24 [110/30] via 192.168.12.2, r1-eth1, 00:15:29
O>* 192.168.2.0/24 [110/20] via 192.168.14.2, r1-eth0, 01:02:53
O   192.168.12.0/30 [110/10] is directly connected, r1-eth1, 01:17:38
C>* 192.168.12.0/30 is directly connected, r1-eth1, 01:34:43
O   192.168.14.0/30 [110/10] is directly connected, r1-eth0, 01:16:17
C>* 192.168.14.0/30 is directly connected, r1-eth0, 01:34:43
O>* 192.168.23.0/30 [110/20] via 192.168.12.2, r1-eth1, 01:07:52
frr-pc#

```

Figure 41. Displaying router r1's routing table.

Consider the figure above. The output displays the networks learned along with additional attributes, such as the routing protocol that discovered the network. For example, networks with the symbol  $\odot$  on the left are learned from OSPF, whereas those with symbol  $\square$  are directly connected to the device. Furthermore, the symbol  $\boxtimes$  on the left of the displayed network indicates that this entry is selected to reach the network. As for the symbol  $\ast$ , this indicates that the route is inserted in Forwarding Information Base (FIB) of the router.

Notice that the route to the network 192.168.1.0/24 has a next-hop IP 192.168.12.2 which corresponds to router r2.

**Step 3.** In the host h1 terminal, issue the following command to test the connectivity between host h1 and host h2.

```
ping 192.168.2.10 -c 4
```

```

Host: h1
root@frr-pc:~# ping 192.168.2.10 -c 4
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=60 time=0.907 ms
64 bytes from 192.168.2.10: icmp_seq=2 ttl=60 time=0.093 ms
64 bytes from 192.168.2.10: icmp_seq=3 ttl=60 time=0.084 ms
64 bytes from 192.168.2.10: icmp_seq=4 ttl=60 time=0.080 ms

--- 192.168.2.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 64ms
rtt min/avg/max/mdev = 0.080/0.291/0.907/0.355 ms
root@frr-pc:~#

```

Figure 42. Performing a connectivity test from host h1 to host h2.

The figure above shows a successful connectivity test.

This concludes lab 6. Stop the emulation and then exit out of MiniEdit.

## References

1. Moy, J. "RFC 1247: OSPF version 2." 1991.
2. Moy, J. "RFC 2328: OSPF version 2." 1998.
3. Coltun, R., Ferguson, D., Moy, J., Lindem, A. "RFC 5340, OSPF for IPv6." 2008.
4. Peterson, L. "Computer networks: a systems approach." 6th Edition, 2021.
5. Tanenbaum, D. Wetherall. "Computer networks." 5th Edition. 2012.
6. J. Kurose, K. Ross. "Computer networking, a top-down approach." 7th Edition. 2017.
7. Teare, D., Vachon, B., Graziani, R. "Implementing Cisco IP routing (ROUTE) foundation learning guide: (CCNP ROUTE 300-101)." 2014.
8. Linux foundation collaborative projects. "FRR routing documentation." 2017 [Online] Available: <http://docs.frrouting.org/en/latest/>.



UNIVERSITY OF  
**SOUTH CAROLINA**

## OPEN SHORTEST PATH FIRST

### Exercise 2: Configuring OSPF Virtual Link

Document Version: **10-22-2021**



Award 1829698

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

## Contents

1	Exercise description .....	3
1.1	Credentials .....	3
2	Setting the environment.....	4
3	Deliverables.....	6

## 1 Exercise description

This exercise is a practice to the Open Shortest Path First (OSPF) virtual link. The OSPF protocol requires all the nonbackbone areas to be directly connected to the backbone area (i.e., area 0). In cases, where a direct connection is not possible, a virtual link can be configured. Consider the figure below. Area 2 is not directly connected to the backbone area. In this exercise, you will configure a virtual link between routers r1 and r3 to connect area 2 to the backbone area. Therefore, host h1 can communicate with host h2.

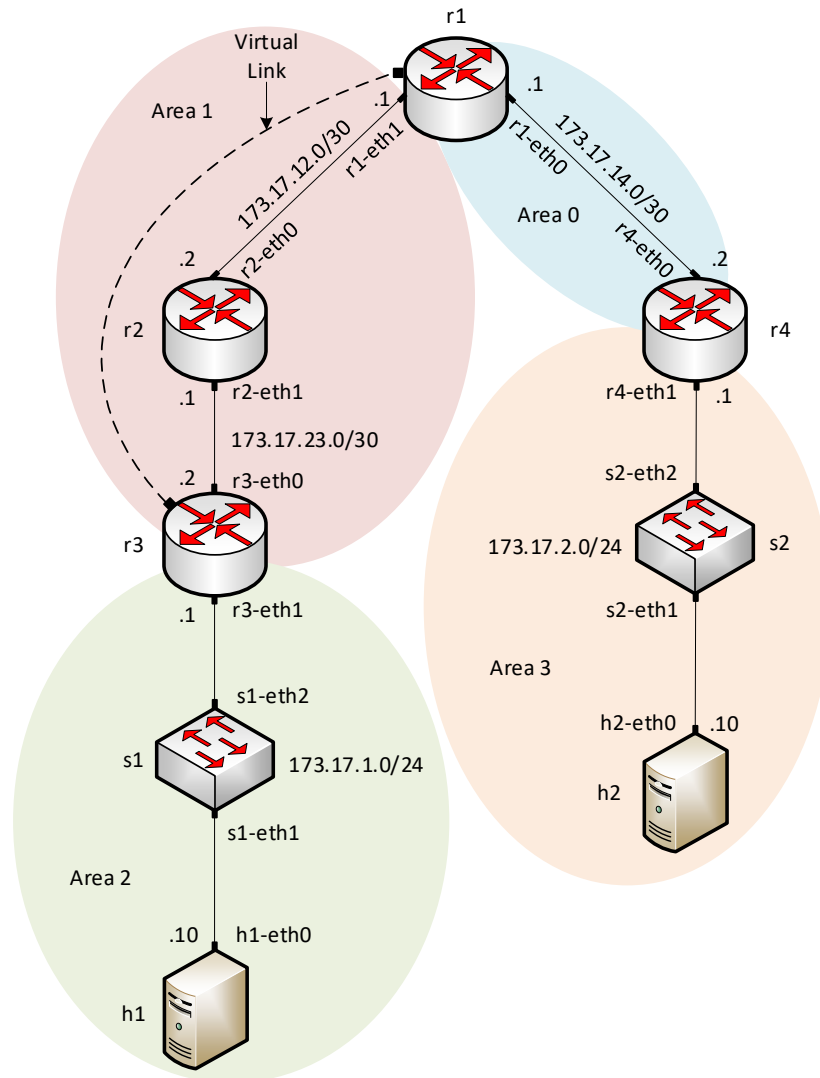


Figure 1. Exercise topology.

### 1.1 Credentials

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

Table 1. Credentials to access the Client machine.

Device	Account	Password
Client	admin	password

## 2 Setting the environment

Follow the steps below to set the environment.

**Step 1.** Open MiniEdit by double-clicking the shortcut on the desktop. If a password is required type `password`.



Figure 2. MiniEdit shortcut.

**Step 2.** Load the topology located at `/home/frr/OSPF_Labs/Exercise2/`.

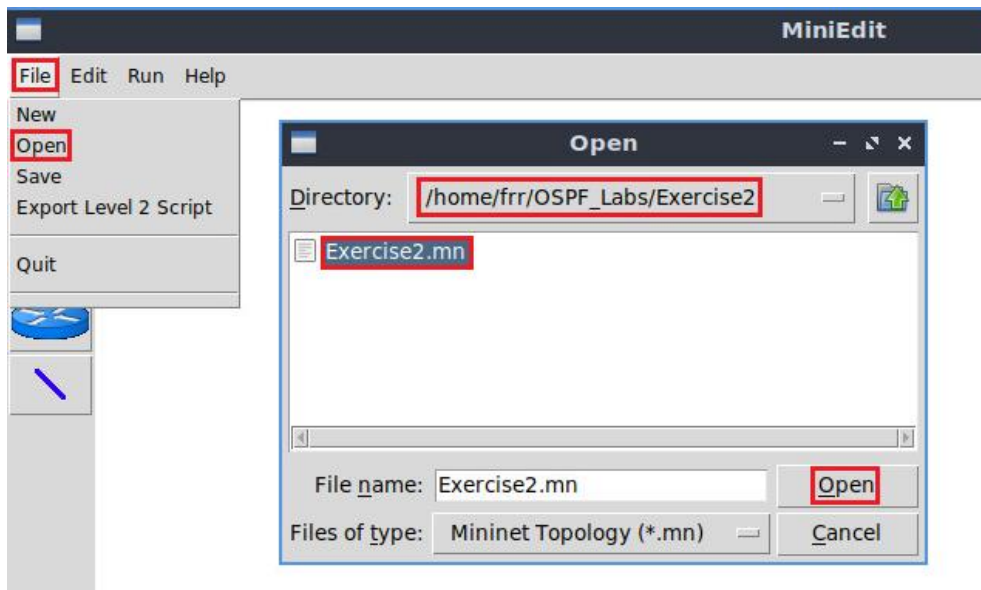


Figure 3. MiniEdit's open dialog.

**Step 3.** Run the emulation by clicking on the button located on the lower left-hand side.



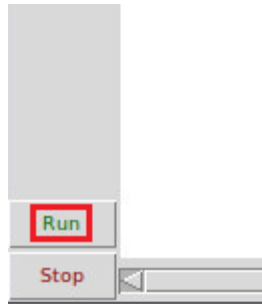


Figure 4. Starting the emulation.

**Step 4.** Open the Linux terminal by clicking on the icon located in the taskbar.

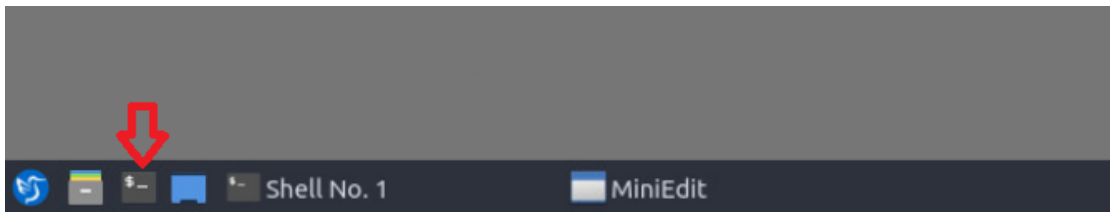


Figure 5. Opening Linux terminal.

**Step 5.** Issue the command shown below to set the lab environment. This command configures the IP addresses in all routers. If a password is required, type `password`.

```
sudo set_env exercise2
```

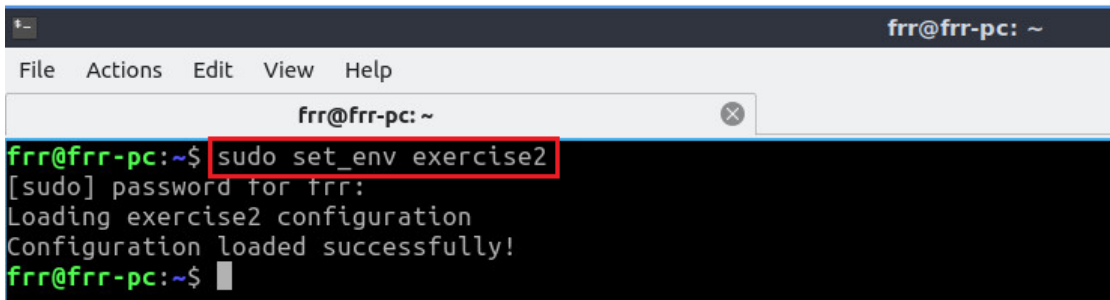


Figure 6. Setting the lab environment.

**Step 6.** Type the following command to exit the Linux terminal.

```
exit
```

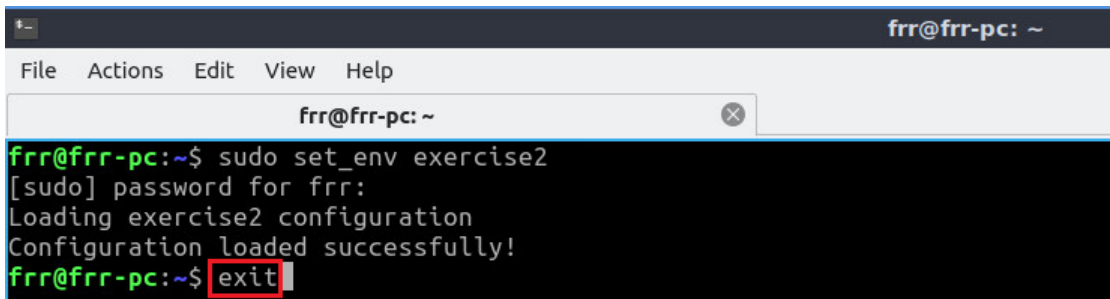


Figure 7. Exiting the Linux terminal.

### 3 Deliverables

Follow the steps below to complete the exercise.

- a) Open a terminal in each router, then, start zebra and the OSPF daemons.
- b) Access each router CLI using the `vttysh` command.
- c) Verify the directly connected routes in each router.
- d) Assign an OSPF router ID to each router.
- e) Configure router r1 to participate in the OSPF process of areas 0 and 1.
- f) Configure router r2 to participate in the OSPF process of area 1.
- g) Configure router r3 to participate in the OSPF process of areas 1 and 2.
- h) Configure router r4 to participate in the OSPF process of areas 0 and 3.
- i) Set the passive interfaces in routers r3 and r4.
- j) Verify the routing and OSPF neighbor tables of each router.
- k) Perform a connectivity test from host h1 to host h2. The connectivity test will not be successful.
- l) Configure a virtual link in router r3.
- m) Similarly, configure a virtual link in router r1.
- n) Repeat **k)**. In this test, the connectivity between host h1 and host h2 must be successful.
- o) Repeat **j)**.



UNIVERSITY OF  
**SOUTH CAROLINA**

## **OPEN SHORTEST PATH FIRST**

### **Lab 7: OSPFv2 Authentication**

Document Version: **10-22-2021**



Award 1829698

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

## Contents

Overview .....	3
Objectives.....	3
Lab settings .....	3
Lab roadmap .....	3
1 Introduction .....	3
1.1 OSPF authentication modes.....	4
1.2 OSPF authentication types .....	5
2 Lab topology.....	6
2.1 Lab settings.....	7
2.2 Loading the topology.....	8
2.3 Load the zebra daemon and verify the configuration .....	10
3 Configuring multi-area OSPF.....	14
3.1 Configuring OSPF in routers r1, r2, and r3 .....	14
3.2 Verifying OSPF neighbors, routing information, and connectivity .....	18
4 Configuring OSPF interface-based authentication using simple password.....	20
4.1 Verifying the configuration .....	21
5 Configuring OSPF area-based authentication using MD5 .....	24
5.1 Verifying the configuration .....	26
References .....	28

## Overview

This lab introduces Open Shortest Path First (OSPF) authentication, which is used to secure the exchange of routing information between neighbor routers. This lab shows how to authenticate OSPF routing information using two modes of authentications: a simple password and a cryptographic algorithm. Moreover, the lab shows and discusses the limitations of the simple password mode and how the cryptographic algorithm mitigates the shortcomings of the former.

## Objectives

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

1. Understand the importance of routing authentication.
2. Configure multi-area OSPF.
3. Authenticate the exchange of routing information between OSPF participants.
4. Identify the limitations of the OSPF authentication modes.

## Lab settings

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

Table 1. Credentials to access the Client 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 multi-area OSPF.
4. Section 4: Configuring OSPF interface-based authentication using simple password.
5. Section 5: Configuring OSPF area-based authentication using MD5.

## 1 Introduction

Network security can be compromised by an attacker gaining access to a router to redirect traffic by injecting malicious routing information. For instance, an attacker can redirect traffic to pass through a device that captures traffic without changing the destination. In this way, the attacker leaves few footprints, making it difficult to detect, mitigate and debug such attacks. Figure 1 describes this situation where router r1 has been compromised to redirect the traffic through router r2. In this way, the attacker can collect data from the traffic going to route r3.

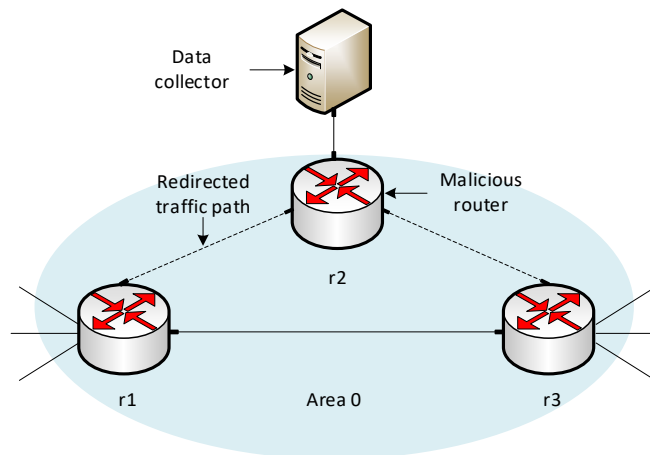


Figure 1. Router r2 injects into router r1 a malicious route to redirect traffic originating from router r1 to pass through router r2, which collects the traffic going to router r3.

The cause of such problems originates from the way routing protocols such as OSPF exchange routing information. OSPF is a link-state routing protocol that establishes neighbor relationships by periodically discovering and maintaining a common database with neighbor routers. A way to prevent this type of attack is by using routing protocol authentication methods to ensure the participants of the routing process are trustworthy.

### 1.1 OSPF authentication modes

The OSPF protocol supports routing authentication based on an authentication key (i.e., password) known by the sending and receiving routers. The aim of this process is validating the legitimacy of the routes exchanged by the OSPF participants. The types of authentications available in the OSPF protocol are<sup>12</sup>:

1. Null authentication (i.e., no authentication).
2. Simple password.
3. Cryptographic authentication.

By default, the OSPF protocol does not perform any authentication on routing updates. The first type does not examine whether the OSPF header is authentic or not. The only validation process performed is verifying the OSPF packet checksum to detect data

corruption. The second method consists of setting a password that is exchanged with the neighbors.

Using the simple password authentication method prevents malicious routers from inadvertently joining the network. However, this type of encryption is vulnerable to passive attacks, which can be performed by anyone with physical access to the network collecting OSPF packets (e.g., hello packets). The attacker can inspect OSPF packet headers and obtain the secret key which is sent over the network and is the same in each pair of routers. Therefore, the attack described in Figure 1 can be delivered.

The OSPF protocol supports the Message Digest 5 (MD5) cryptographic hash algorithm to authenticate and verify the integrity of OSPF packets. The algorithm takes as input a message of arbitrary length to produce a 128-bit output known as message digest of the input. The strength of the algorithm leverage on the computational difficulty of creating two outputs with the same message digest. Additionally, the OSPF protocol includes in the packet header a nondecreasing sequence number to protect against replay attacks. Therefore, using the MD5 cryptographic hash algorithm protects the OSPF participants against replay attacks.

## 1.2 OSPF authentication types

OSPF supports area-based and interface-based authentication. The area-based authentication applies to packets received by the interfaces pertaining to an OSPF area. On the other hand, interface-based is restricted only to packets received by a specific interface. Moreover, a router participating in the OSPF process can use both simple password and MD5 authentication in different interfaces.

Consider Figure 2. Router r2 authenticates router r1 based on the interface using the simple password mode. On the other hand, the authentication between router r2 and router r3 is performed using an area-based authentication using the cryptographic mode (i.e., MD5). Notice that if there is a conflict between the interface-based and area-based mode, the interface-based authentication type takes precedence.

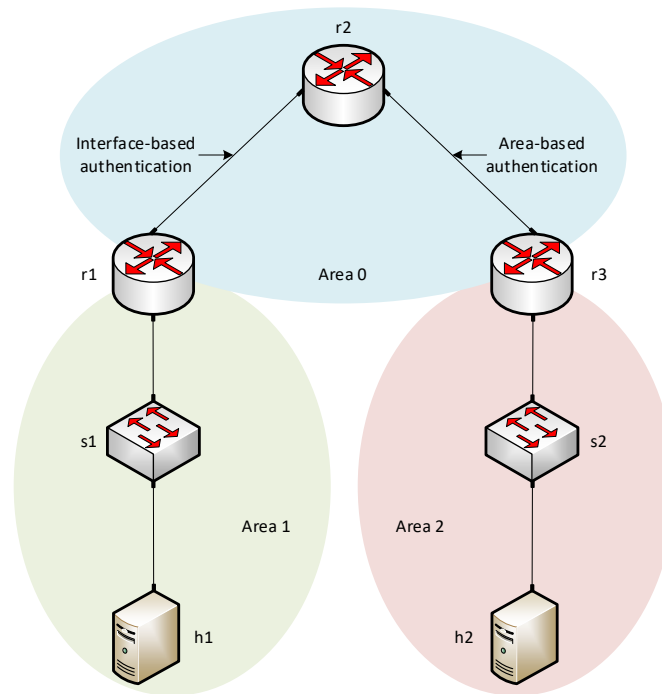


Figure 2. Router r2 authenticates router r1 based on the interface using the simple password mode. The authentication between router r2 and router r3 is performed using an area-based authentication using the cryptographic mode (i.e., MD5).

## 2 Lab topology

Consider Figure 3. The topology comprises three OSPF areas, where area 0 is the backbone area and areas 1 and 2 are physically connected. You will configure a multi-area OSPF. Then, you will configure OSPF authentication using a simple password between router r1 and router r2 based on their interfaces. On the other hand, the authentication between router r2 and router r3 will be performed using the cryptographic authentication method based on the OSPF participants' area.



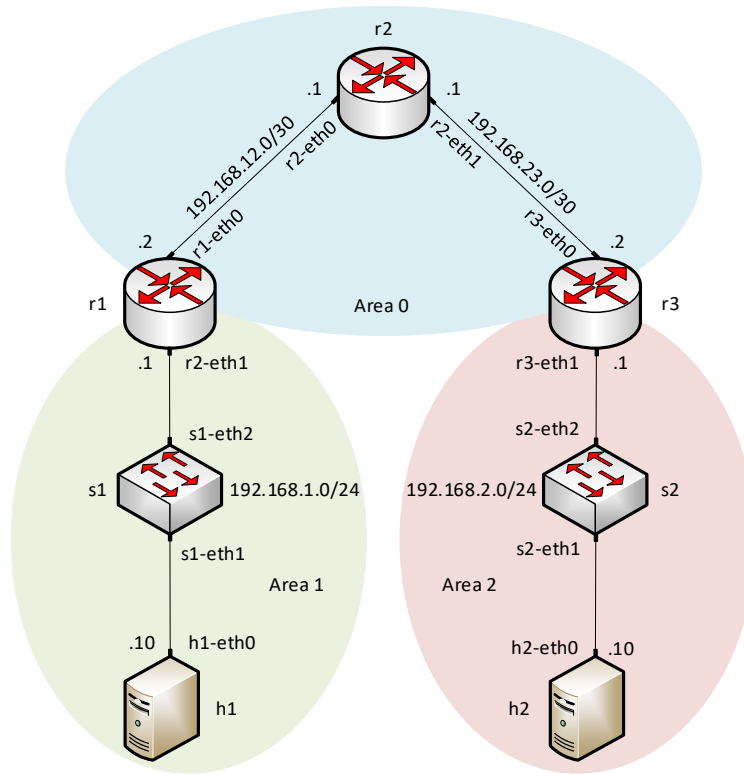


Figure 3. Lab topology.

## 2.1 Lab settings

Table 2 contains information about the devices and their corresponding interfaces, IPv4 addresses, subnet, and default gateway.

Table 2. Topology Information.

Device	Interface	IPv4 Address	Subnet	Default Gateway
Router 1 (r1)	r1-eth0	192.168.12.1	/30	N/A
	r1-eth1	192.168.23.1	/30	N/A
Router 2 (r2)	r2-eth0	192.168.12.2	/30	N/A
	r2-eth1	192.168.11	/24	N/A
Router 3 (r1)	r3-eth0	192.168.23.2	/30	N/A
	r3-eth1	192.168.2.1	/24	N/A
Host 1 (h1)	h1-eth0	192.168.1.10	/24	192.168.1.1
Host 2 (h2)	h2-eth0	192.168.2.10	/24	192.168.2.1

## 2.2 Loading the topology

**Step 1.** Launch MiniEdit by double clicking on the shortcut located on the desktop. When prompted for a password, type `password`.



Figure 4. MiniEdit shortcut.

**Step 2.** In MiniEdit's menu bar, click on *File* then *open* to load the lab topology. Locate the *lab7.mn* topology file in the default directory, `/home/frr/OSPF_Labs/lab7` and click on *Open*.

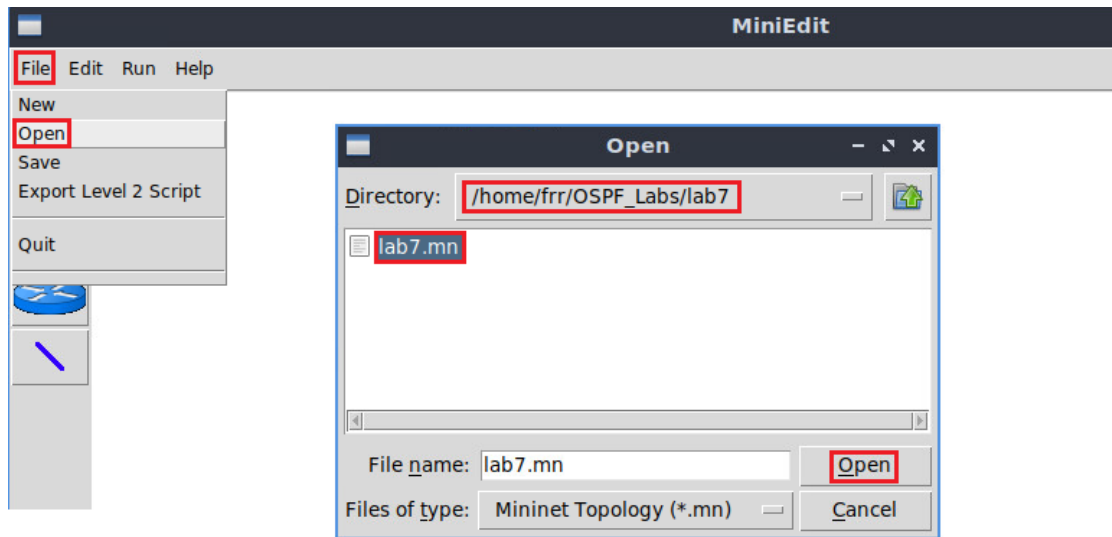


Figure 5. MiniEdit's open dialog.

**Step 3.** Click on the *Run* button located on the lower left-hand side to run the emulation.



Figure 6. Starting the emulation.

At this point, the topology is loaded with all the required network components. Next, you will execute a script that will configure the interfaces of the routers.

Automating the configuration of interfaces with the script is intended to facilitate the lab experience and to focus on the OSPF experiment.

**Step 4.** Open the Linux terminal by clicking on the icon located in the taskbar.

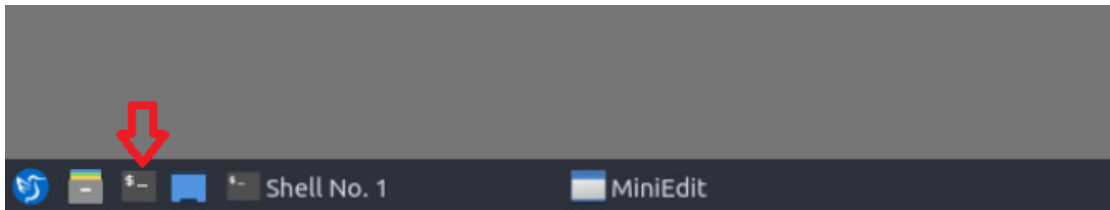


Figure 7. Opening Linux terminal.

**Step 5.** Issue the command shown below to set the lab environment. This command configures the IP addresses of the routers' interface according to Table 2. If a password is required, type `password`.

```
sudo set_env exercise1
```

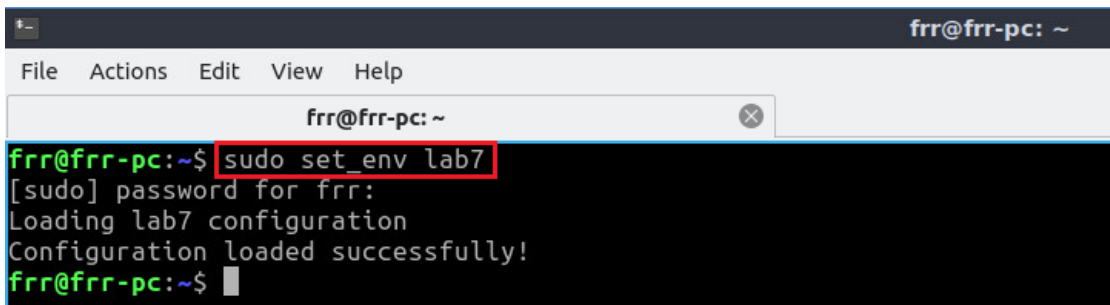


Figure 8. Setting the lab environment.

**Step 6.** Type the following command to exit the Linux terminal.

```
exit
```

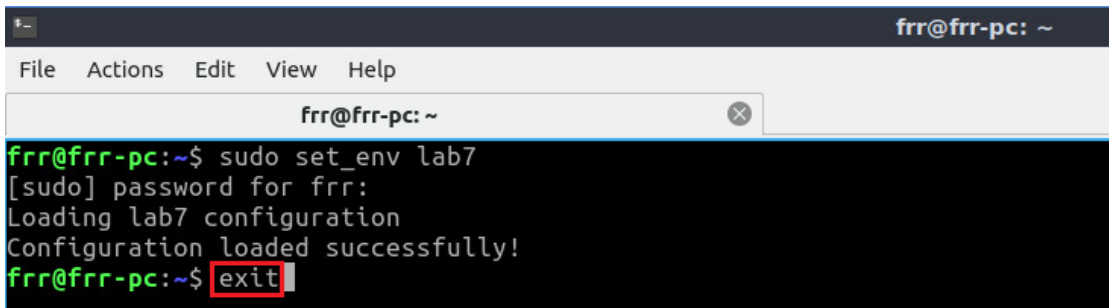


Figure 9. Exiting the Linux terminal.

**Step 7.** Click on Mininet's terminal, i.e., the one launched when MiniEdit was started.

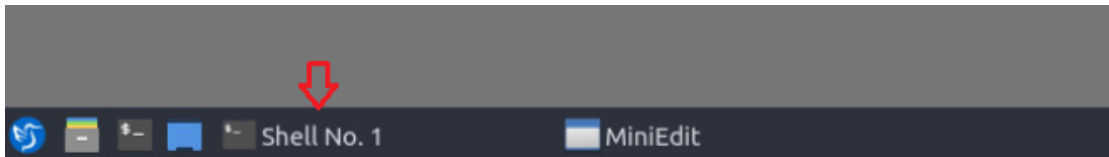


Figure 10. Opening Mininet's terminal.

**Step 8.** Issue the following command to display the interface names and connections.

```
links
```

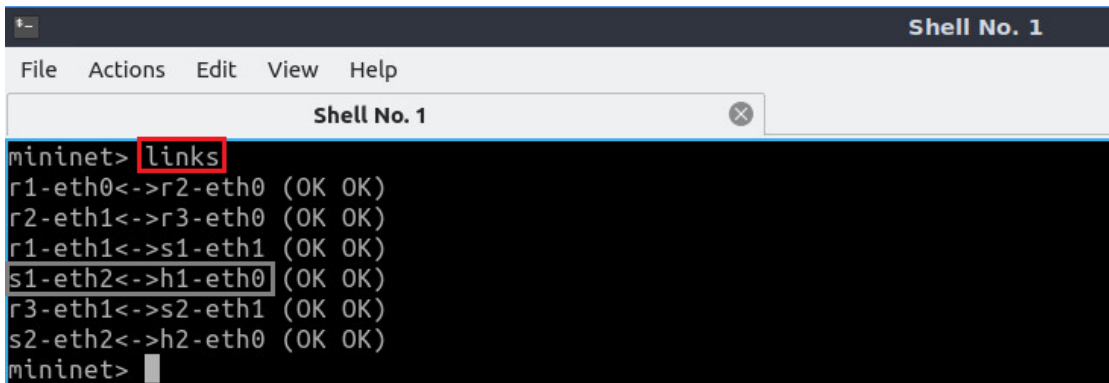


Figure 11. Displaying network interfaces.

In the figure above, the link displayed within the gray box indicates that interface *eth2* of switch *s2* connects to interface *eth0* of host *h1* (i.e., *s2-eth2<->h1-eth0*).

### 2.3 Load the zebra daemon and verify the configuration

In this section, you will verify that the IP addresses listed in Table 2 and inspect the routing table of the routers.

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

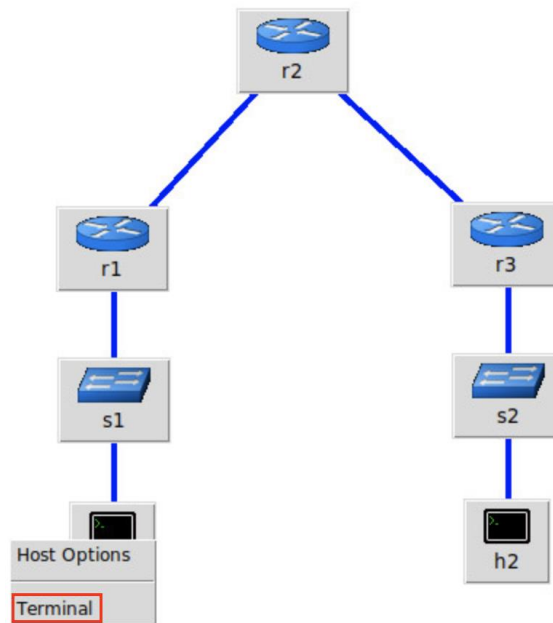


Figure 12. Opening host h1 terminal.

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

```
ifconfig
```

```

root@frr-pc:~# ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.10 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::7c11:30ff:fea5:d022 prefixlen 64 scopeid 0x20<link>
    ether 7e:11:30:a5:d0:22 txqueuelen 1000 (Ethernet)
    RX packets 32 bytes 3781 (3.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 12 bytes 936 (936.0 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@frr-pc:~#
  
```

Figure 13. Displaying host h1 interfaces and IP addresses.

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

route

```

"Host: h1"
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
  inet 192.168.1.10 netmask 255.255.255.0 broadcast 192.168.1.255
  inet6 fe80::7c11:30ff:fea5:d022 prefixlen 64 scopeid 0x20<link>
  ether 7e:11:30:a5:d0:22 txqueuelen 1000 (Ethernet)
  RX packets 32 bytes 3781 (3.7 KB)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 12 bytes 936 (936.0 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@frr-pc:~# 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@frr-pc:~#

```

Figure 14. Displaying host h1 default route.

**Step 4.** To verify the hosts h2 and h3 default route, proceed by following from step 1 to step 3. Similar results should be observed.

**Step 5.** You will validate that the router interfaces are configured according to Table 2. To proceed, hold right-click on router r1 and select *Terminal*.

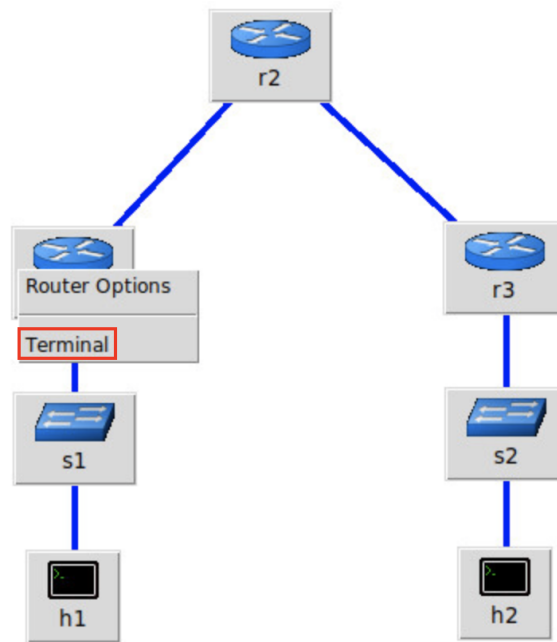


Figure 15. Opening a terminal on router r1.

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

```
zebra
```

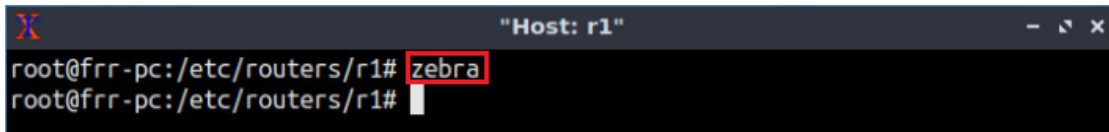


Figure 16. Starting zebra daemon.

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

```
vtysh
```

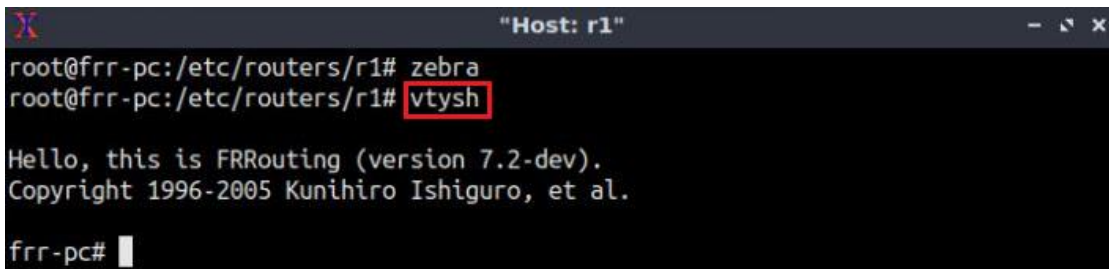
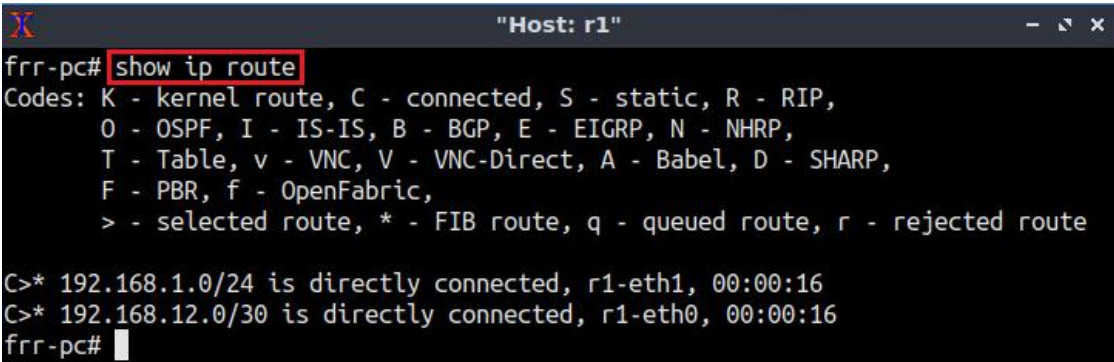


Figure 17. Starting vtysh on router r1.

**Step 8.** Type the following command on router r1 terminal to verify the routing table. It will list all the directly connected networks. Notice that 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
```



```
frr-pc# 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-eth1, 00:00:16
C>* 192.168.12.0/30 is directly connected, r1-eth0, 00:00:16
frr-pc#
```

Figure 18. Displaying routing table of router r1.

**Step 9.** To verify the routing table of the other routers, proceed similarly by following from step 5 to step 8 on the corresponding router terminal.

### 3 Configuring multi-area OSPF

In this section, you will configure multi-area OSPF according to the topology shown in Figure 3.

#### 3.1 Configuring OSPF in routers r1, r2, and r3

**Step 1.** Open router r1's terminal by selecting the icon from the taskbar as shown below.



Figure 19. Opening router r1's terminal.

**Step 2.** In router r1's terminal, type the command below to exit the vtysh session.

```
exit
```



```

Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1#

```

Figure 20. Exiting the vtysh session.

**Step 3.** Enable and start the OSPF routing protocol by typing the following command:

```
ospfd
```

```

Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1#

```

Figure 21. Starting the OSPF daemon.

**Step 4.** To enter router r1's terminal, type the following command:

```
vttysh
```

```

Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

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

frr-pc#

```

Figure 22. Starting vtysh in router r1.

**Step 5.** At this point, you have entered router r1's terminal, to enter configuration mode in router r1, issue the following command:

```
configure terminal
```

```

Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

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

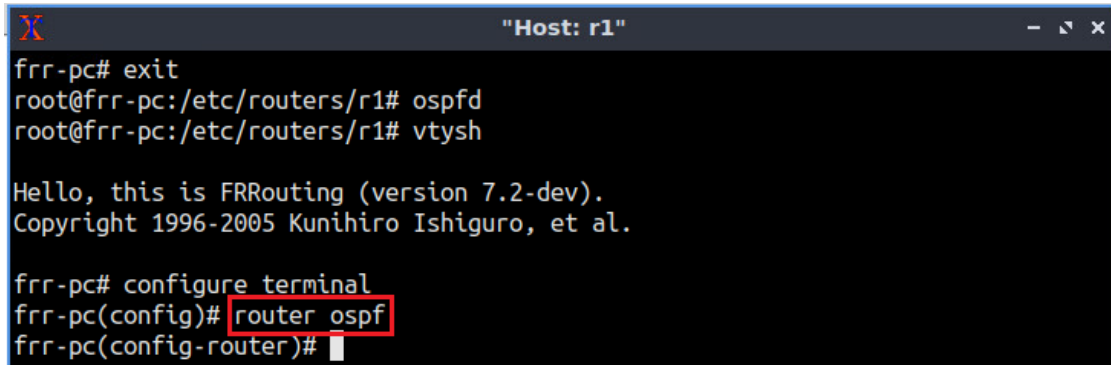
frr-pc# configure terminal
frr-pc(config)#

```

Figure 23. Entering configuration mode in router r1.

**Step 6.** To configure OSPF type the command shown below. This command will enable OSPF configuration mode.

```
router ospf
```

A terminal window titled "Host: r1" showing the configuration of OSPF on a router. The user enters 'exit' to leave the router configuration, then 'ospfd' to start the daemon, and 'vtysh' to enter configuration mode. After a welcome message, they enter 'configure terminal', then 'router ospf'. The command 'router ospf' is highlighted with a red box.

```
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

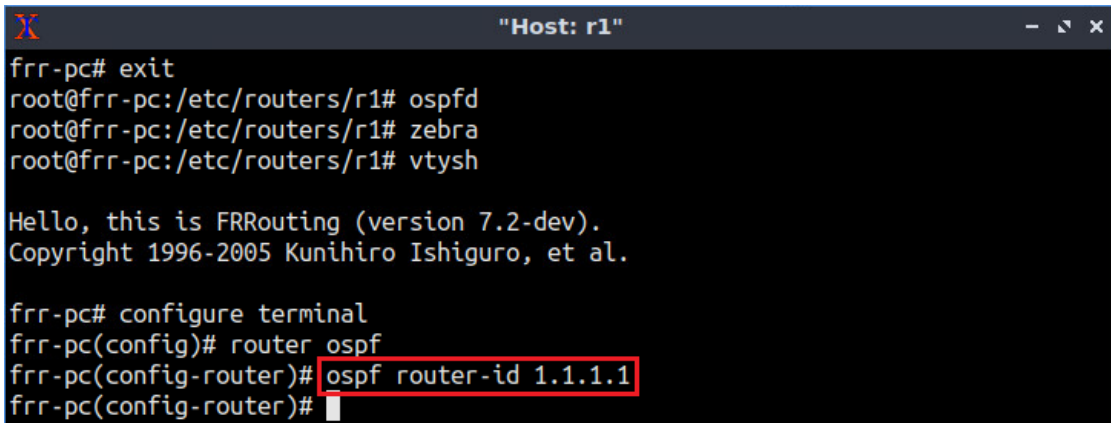
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)#
```

Figure 24. Configuring OSPF in router r1.

**Step 7.** Issue the following command to assign an identification tag to router r1.

```
ospf router-id 1.1.1.1
```

A terminal window titled "Host: r1" showing the configuration of the OSPF router ID. The user enters 'exit', 'ospfd', and 'vtysh'. After the welcome message, they enter 'configure terminal', then 'router ospf', and finally 'ospf router-id 1.1.1.1'. The command 'ospf router-id 1.1.1.1' is highlighted with a red box.

```
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# zebra
root@frr-pc:/etc/routers/r1# vtysh

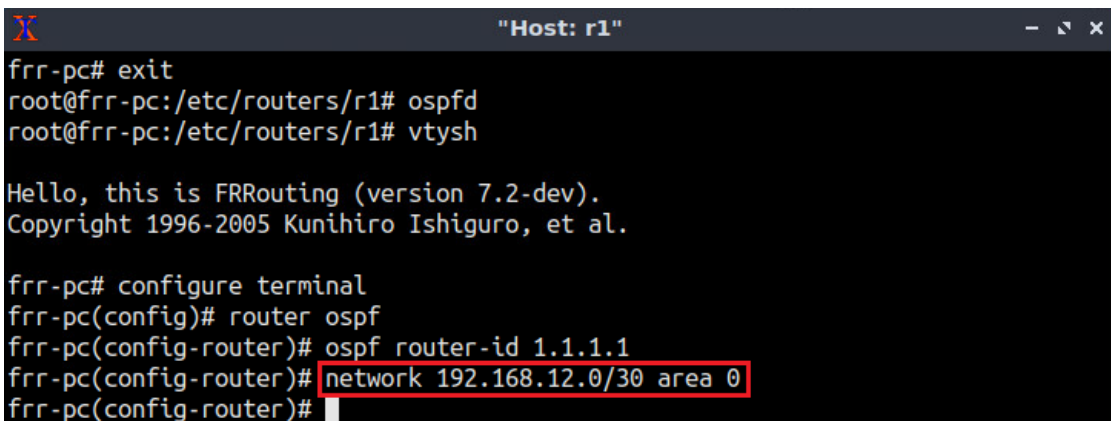
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 1.1.1.1
frr-pc(config-router)#
```

Figure 25. Configuring OSPF router ID in router r1.

**Step 8.** Type the following command to enable the interface *r1-eth0* to participate in the OSPF process in area 0.

```
network 192.168.12.0/30 area 0
```

A terminal window titled "Host: r1" showing the configuration of the OSPF process. The user enters 'exit', 'ospfd', and 'vtysh'. After the welcome message, they enter 'configure terminal', then 'router ospf', 'ospf router-id 1.1.1.1', and finally 'network 192.168.12.0/30 area 0'. The command 'network 192.168.12.0/30 area 0' is highlighted with a red box.

```
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

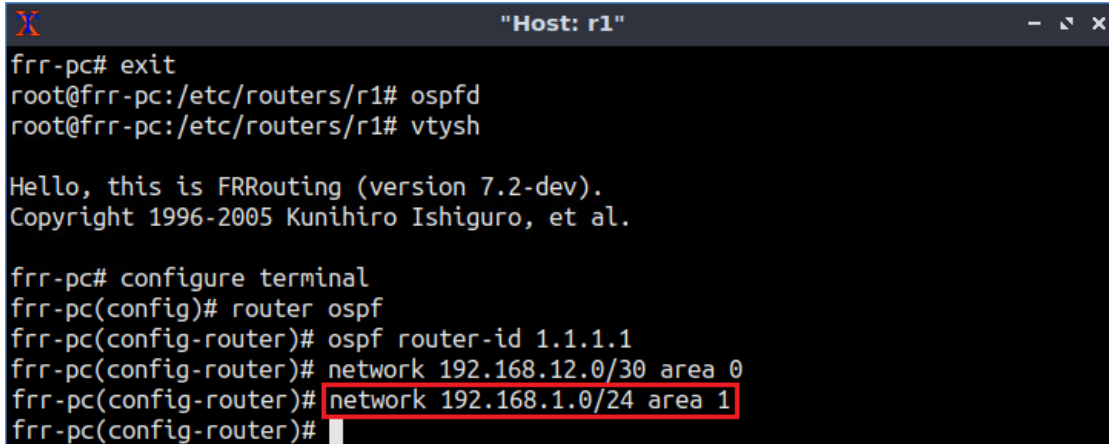
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 1.1.1.1
frr-pc(config-router)# network 192.168.12.0/30 area 0
frr-pc(config-router)#
```

Figure 26. Enabling interface *r1-eth0* to participate in the OSPF process in area 0.

**Step 9.** Issue the following command to enable the interface *r1-eth1* to participate in the OSPF process in area 1.

```
network 192.168.1.0/24 area 1
```

A terminal window titled "Host: r1" showing the configuration of OSPF on a router. The user enters 'exit' to leave the previous configuration, then 'ospfd' to start the daemon, and 'vtysh' to enter configuration mode. A welcome message for FRRouting (version 7.2-dev) is displayed. The user enters 'configure terminal' to enter global configuration mode, then 'router ospf' to enter OSPF configuration mode. The user sets the router ID to 1.1.1.1, then configures area 0 with the network 192.168.12.0/30. Finally, the user configures area 1 with the network 192.168.1.0/24, which is highlighted with a red box in the original image.

```
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

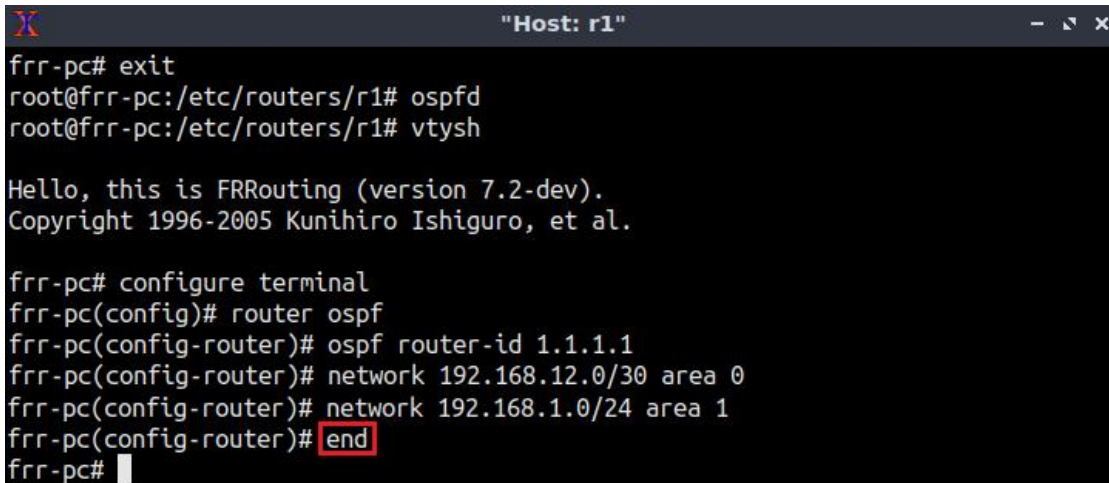
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 1.1.1.1
frr-pc(config-router)# network 192.168.12.0/30 area 0
frr-pc(config-router)# network 192.168.1.0/24 area 1
frr-pc(config-router)#
```

Figure 27. Enabling interface *r1-eth1* to participate in the OSPF process in area 1.

**Step 10.** Type the following command to exit router configuration mode.

```
end
```

A terminal window titled "Host: r1" showing the continuation of the OSPF configuration. The user enters 'end' to exit the OSPF configuration mode, which is highlighted with a red box in the original image. The prompt returns to 'frr-pc#'.

```
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 1.1.1.1
frr-pc(config-router)# network 192.168.12.0/30 area 0
frr-pc(config-router)# network 192.168.1.0/24 area 1
frr-pc(config-router)# end
frr-pc#
```

Figure 28. Exiting router configuration mode.

Consider the figure above. After configuring OSPF in router *r1*, the interfaces *r1-eth0*, and *r1-eth1* will participate in the routing process of area 0 and area 1, respectively.

**Step 11.** 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* terminal, issue the commands summarized below.

```

Host: r2
frr-pc# exit
root@frr-pc:/etc/routers/r2# ospfd
root@frr-pc:/etc/routers/r2# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 2.2.2.2
frr-pc(config-router)# network 192.168.12.0/30 area 0
frr-pc(config-router)# network 192.168.23.0/30 area 0
frr-pc(config-router)# end
frr-pc#

```

Figure 29. Router r2 configuration summary.

Consider the figure above. After configuring OSPF in router r2, the interfaces *r2-eth0*, and *r2-eth1* will participate in the routing process of area 0.

**Step 12.** Router r3 is configured similarly to router r2 but with different IP addresses (see Table 2). Those steps are summarized in the following figure. To proceed, in router r3 terminal, issue the commands depicted below.

```

Host: r3
frr-pc# exit
root@frr-pc:/etc/routers/r3# ospfd
root@frr-pc:/etc/routers/r3# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 3.3.3.3
frr-pc(config-router)# network 192.168.23.0/30 area 0
frr-pc(config-router)# network 192.168.2.0/24 area 2
frr-pc(config-router)# end
frr-pc#

```

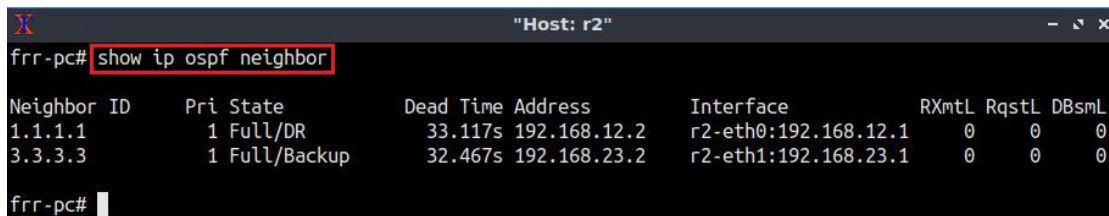
Figure 30. Router r3 configuration summary.

Consider the figure above. After configuring OSPF in router r3, the interfaces *r3-eth0* and *r3-eth1* will participate in the routing process of area 0 and area 2, respectively.

### 3.2 Verifying OSPF neighbors, routing information, and connectivity

**Step 1.** In router's r2 terminal, issue the following command to verify its neighbors.

```
show ip ospf neighbor
```



```

frr-pc# show ip ospf neighbor
Neighbor ID      Pri State           Dead Time Address           Interface           RXmtL RqstL DBsmL
1.1.1.1          1 Full/DR          33.117s 192.168.12.2       r2-eth0:192.168.12.1 0      0      0
3.3.3.3          1 Full/Backup      32.467s 192.168.23.2       r2-eth1:192.168.23.1 0      0      0
frr-pc#

```

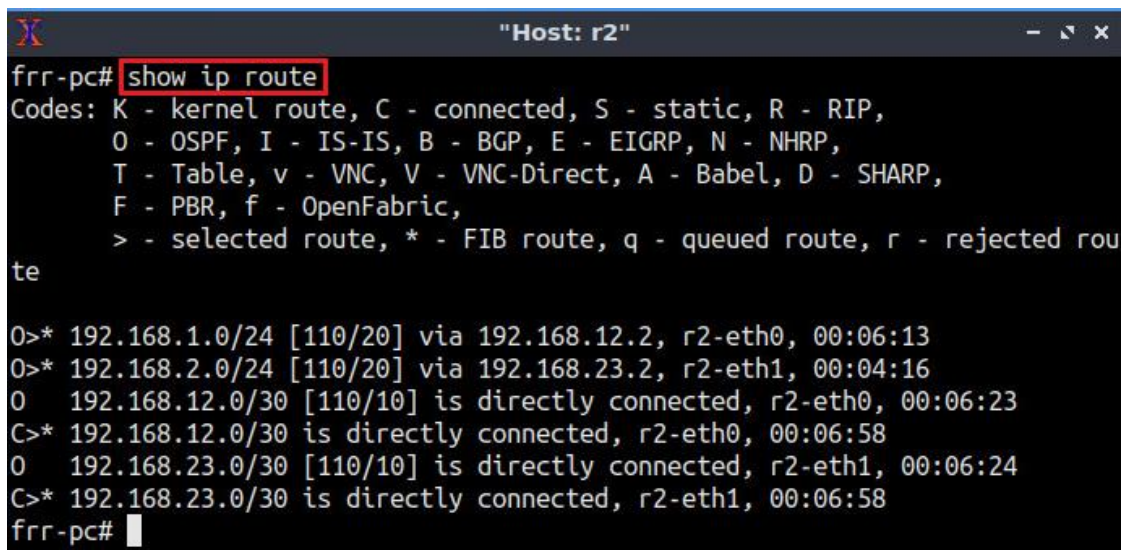
Figure 31. Displaying router r2's OSPF neighbors.

The figure above shows OSPF neighbor information as a function of the interfaces. The columns of the table represent the following information.

- Neighbor ID: Indicates the neighbor router ID. The output of the command above shows the neighbors of router r2.
- Priority: Router priority of the neighbor and neighbor state.
- State: The *Full* state indicates that the local router (i.e., router r2) established a full neighbor relationship with the remote OSPF neighbor.
- Dead Time: It shows the elapsed time since the local router (i.e., router r2) received the last OSPF hello packet.
- Address: Depicts the primary IPv4 address that router uses to participate in the OSPF process.
- Interface: It shows the interface that participates in the OSPF process.

**Step 2.** To display the routing table of router r2, issue the following command:

```
show ip route
```



```


frr-pc# 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

O>* 192.168.1.0/24 [110/20] via 192.168.12.2, r2-eth0, 00:06:13
O>* 192.168.2.0/24 [110/20] via 192.168.23.2, r2-eth1, 00:04:16
O   192.168.12.0/30 [110/10] is directly connected, r2-eth0, 00:06:23
C>* 192.168.12.0/30 is directly connected, r2-eth0, 00:06:58
O   192.168.23.0/30 [110/10] is directly connected, r2-eth1, 00:06:24
C>* 192.168.23.0/30 is directly connected, r2-eth1, 00:06:58
frr-pc#

```

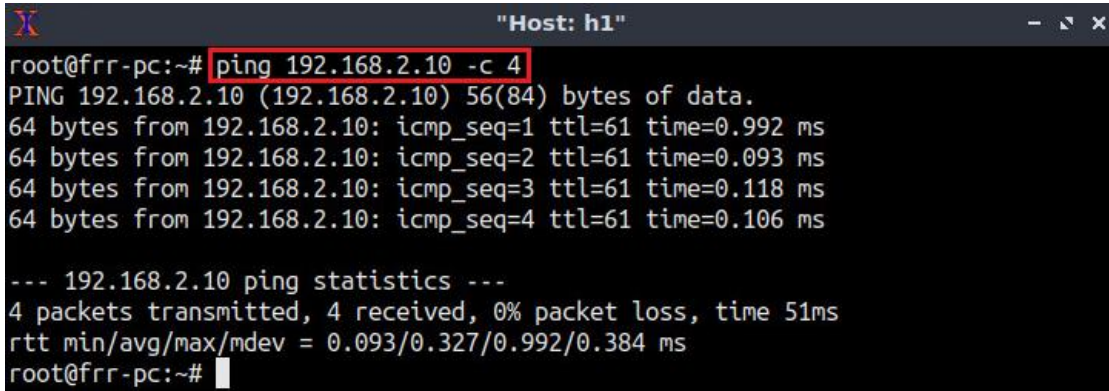
Figure 32. Displaying router r2's routing table.

Consider the above figure. The output displays the networks learned along with additional attributes, such as the routing protocol that discovered the network. For example, networks with the symbol  $\square$  on the left are learned from OSPF, whereas those with symbol  $\square$  are directly connected to the device. Furthermore, the symbol  $\square$  on the left of the displayed network indicates that this entry is selected to reach the network. As for the

symbol , this indicates that the route is inserted in Forwarding Information Base (FIB) of the router.

**Step 3.** In the host h1 terminal, issue the following command to test the connectivity between host h1 and host h2. You will observe a successful connectivity test.

```
ping 192.168.2.10 -c 4
```



```

Host: h1
root@frr-pc:~# ping 192.168.2.10 -c 4
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=61 time=0.992 ms
64 bytes from 192.168.2.10: icmp_seq=2 ttl=61 time=0.093 ms
64 bytes from 192.168.2.10: icmp_seq=3 ttl=61 time=0.118 ms
64 bytes from 192.168.2.10: icmp_seq=4 ttl=61 time=0.106 ms

--- 192.168.2.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 51ms
rtt min/avg/max/mdev = 0.093/0.327/0.992/0.384 ms
root@frr-pc:~#

```

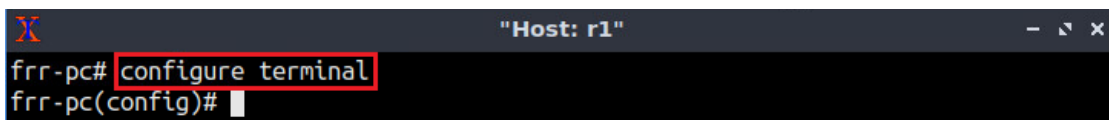
Figure 33. Testing connectivity from host h1 to host h2.

#### 4 Configuring OSPF interface-based authentication using simple password

In this section, you will set a simple password in routers r1 and r2 to authenticate OSPF routing exchange. Then, you will verify the configuration and observe the limitations of this type of authentication.

**Step 1.** In router r1, issue the following command to enter configuration mode.

```
configure terminal
```



```

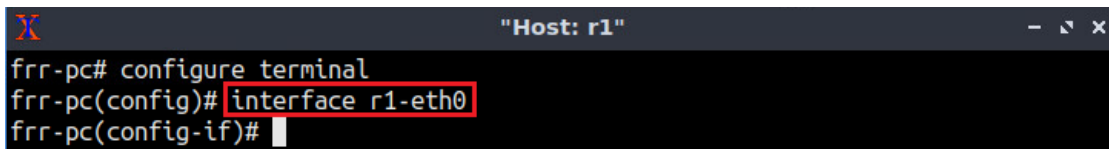
Host: r1
frr-pc# configure terminal
frr-pc(config)#

```

Figure 34. Entering configuration mode in router r1.

**Step 2.** In this step, you will configure interface *r1-eth0* to participate in the authentication process. Issue the following command to enter interface *r1-eth0* configuration.

```
interface r1-eth0
```



```

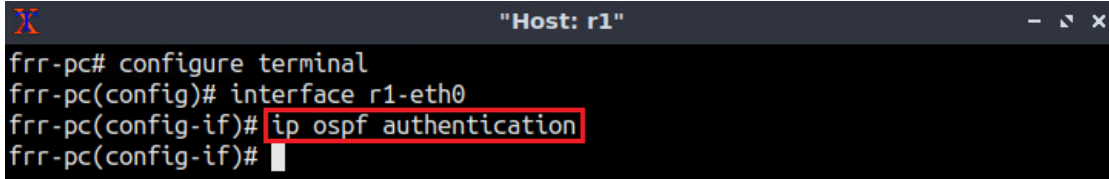
Host: r1
frr-pc# configure terminal
frr-pc(config)# interface r1-eth0
frr-pc(config-if)#

```

Figure 35. Configuring OSPF in router r3.

**Step 3.** Issue the following command to enable OSPF authentication in the interface *r1-eth0*.

```
ip ospf authentication
```

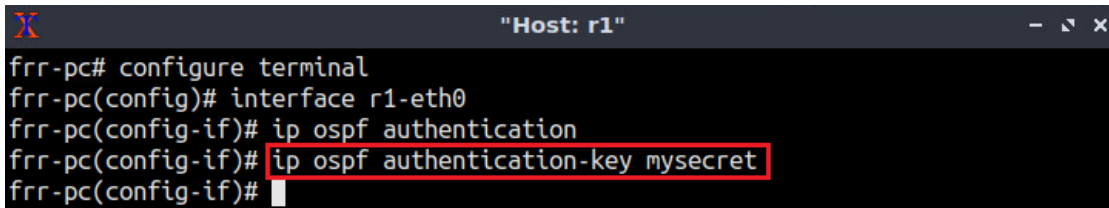


```
frr-pc# configure terminal
frr-pc(config)# interface r1-eth0
frr-pc(config-if)# ip ospf authentication
frr-pc(config-if)#
```

Figure 36. Enabling OSPF authentication in the interface *r1-eth0*.

**Step 4.** Type the following command to exit from the interface configuration. Notice that the authentication password is `mysecret`.

```
ip ospf authentication-key mysecret
```

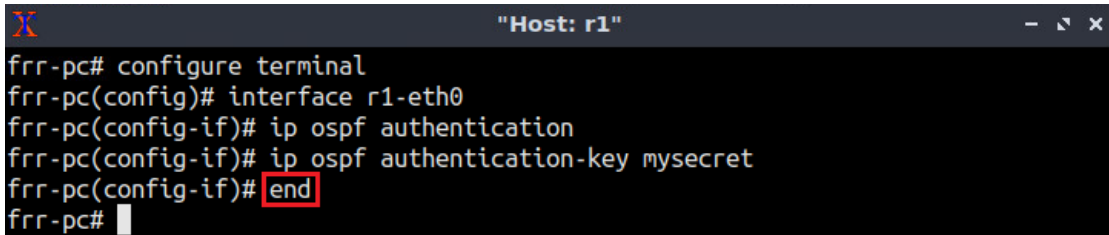


```
frr-pc# configure terminal
frr-pc(config)# interface r1-eth0
frr-pc(config-if)# ip ospf authentication
frr-pc(config-if)# ip ospf authentication-key mysecret
frr-pc(config-if)#
```

Figure 37. Setting a plain text authentication password.

**Step 5.** Type the following command to exit configuration mode.

```
end
```



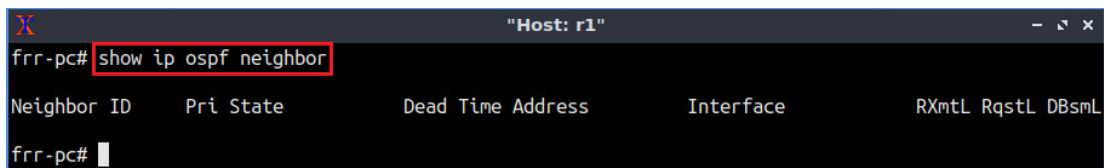
```
frr-pc# configure terminal
frr-pc(config)# interface r1-eth0
frr-pc(config-if)# ip ospf authentication
frr-pc(config-if)# ip ospf authentication-key mysecret
frr-pc(config-if)# end
frr-pc#
```

Figure 38. Exiting from configuration mode.

#### 4.1 Verifying the configuration

**Step 1.** Issue the following command to verify the OSPF neighbors in router *r1*.

```
show ip ospf neighbor
```



```

frr-pc# show ip ospf neighbor
Neighbor ID      Pri State           Dead Time Address      Interface      RXmtL RqstL DBsmL
frr-pc#

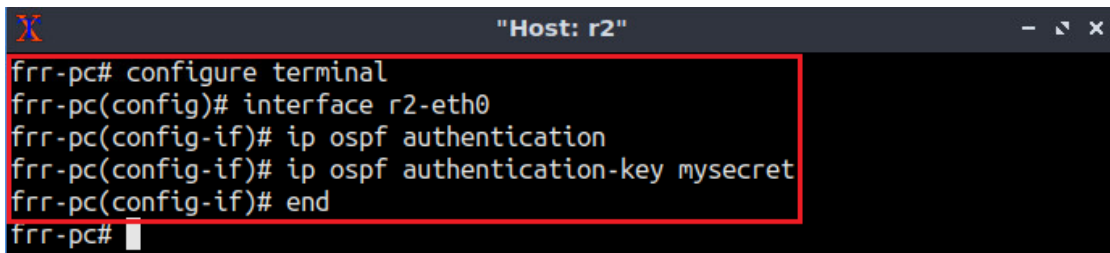
```

Figure 39. Displaying OSPF neighbors.

The figure above shows an empty OSPF neighbor table in router r1. This result shows that router r1 cannot accept routing information from a peer (i.e., router r2 with neighbor ID 2.2.2.2) that is not authenticated. Similar result is observed in the OSPF neighbor table of router r2, where the routing information of router r1 (with a neighbor ID 1.1.1.1) is also missing.

Notice that it takes around 40 seconds to refresh the OSPF neighbor table.

**Step 2.** To allow routers r1 and r2 exchanging routing information, proceed by issuing the commands summarized in the figure below.



```

frr-pc# configure terminal
frr-pc(config)# interface r2-eth0
frr-pc(config-if)# ip ospf authentication
frr-pc(config-if)# ip ospf authentication-key mysecret
frr-pc(config-if)# end
frr-pc#

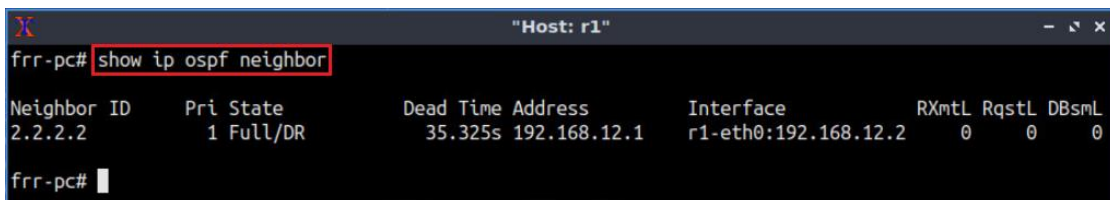
```

Figure 40. Router r2 configuration summary.

Notice that the passwords must be the same in router r1 and router r2 so that they can exchange routing information.

**Step 3.** Go back to router r1 terminal and issue the command below.

```
show ip ospf neighbor
```



```

frr-pc# show ip ospf neighbor
Neighbor ID      Pri State           Dead Time Address      Interface      RXmtL RqstL DBsmL
2.2.2.2          1 Full/DR         35.325s 192.168.12.1  r1-eth0:192.168.12.2  0    0    0
frr-pc#

```

Figure 41. Displaying OSPF neighbors.

In the figure above you will observe that router r2 appears now as a neighbor of router r1. This result indicates that routers r1 and r2 can now exchange authenticated routing information.

**Step 4.** Perform a connectivity test by issuing the following command. The result will show a successful connectivity test.

```
ping 192.168.2.10 -c 4
```



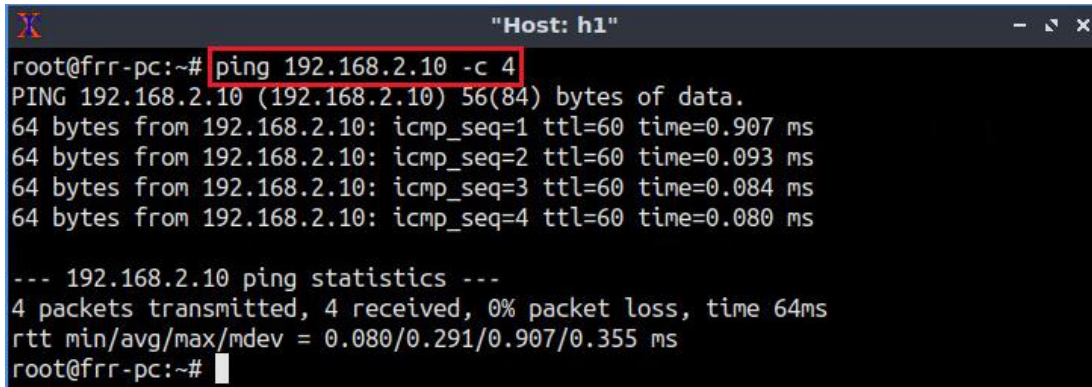


Figure 42. Performing a connectivity test between host h1 and host h2.

**Step 5.** Exit from router r1 configuration mode by issuing the following command:

```
exit
```

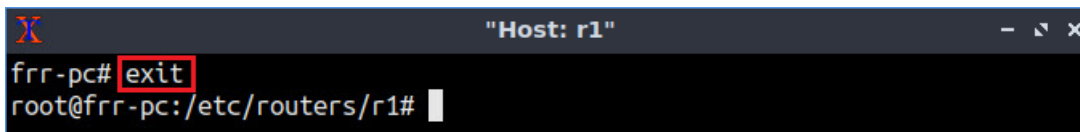


Figure 43. Exiting from router r1 configuration mode.

**Step 6.** Issue the following command to start Wireshark traffic analyzer.

```
wireshark
```

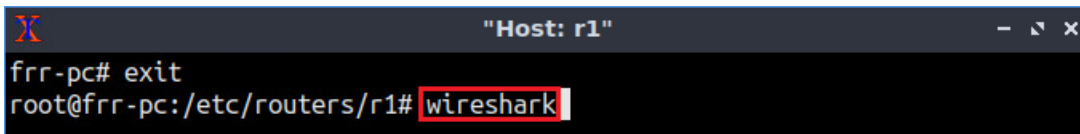


Figure 44. Starting Wireshark traffic analyzer.

**Step 7.** Select the interface *r1-eth0* then, start the traffic capturing by clicking on the button in the upper left-hand side.

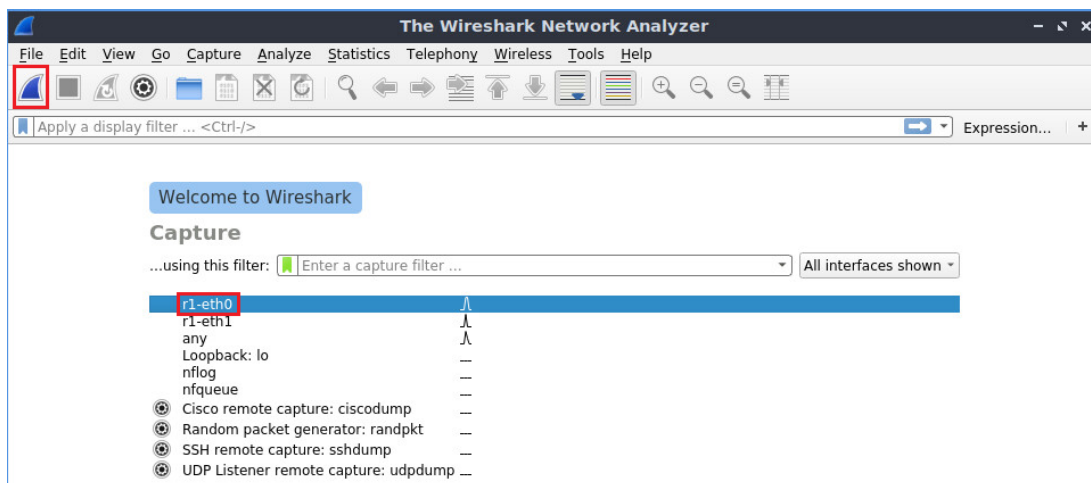


Figure 45. Starting packet capturing on interface *r1-eth0*.

**Step 8.** After capturing OSPF hello packets, stop the traffic capture by clicking on the red button located on the upper left-hand side.

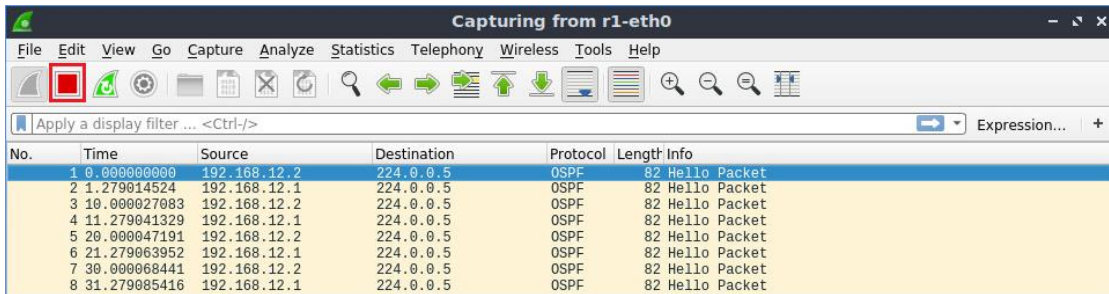


Figure 46. Stopping the packet capture.

**Step 9.** Click on any OSPF hello packet and open the OSPF header by clicking on the tabs as shown below.

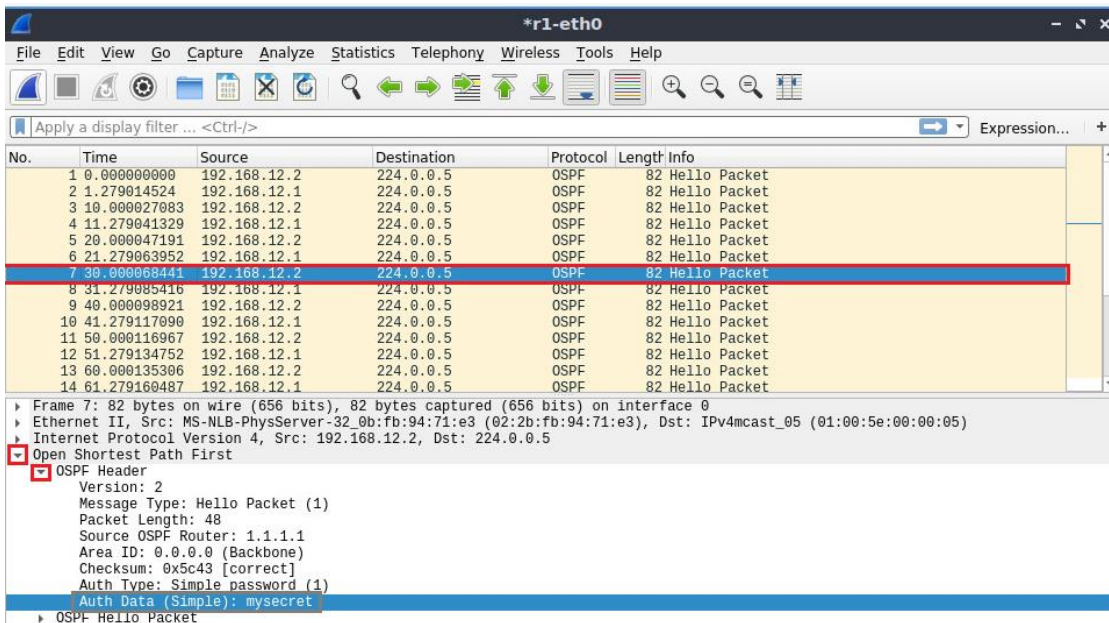


Figure 47. Inspecting OSPF packet header.

In the figure above, you will notice that the OSPF authentication field *Auth Data (Simple)* contains the password (i.e., *mysecret*).

A simple password authentication prevents malicious routers from inadvertently joining the network. However, this type of encryption is vulnerable to passive attack, which can be performed by anyone with physical access to the network capturing traffic.

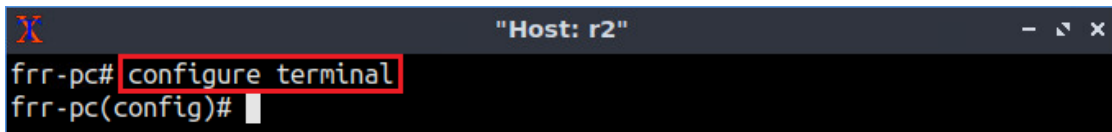
## 5 Configuring OSPF area-based authentication using MD5

The OSPF protocol counts with a more secure authentication type based on the cryptographic algorithm MD5. Using this method, the secret password is not transmitted

over the network, thus, providing an improved security with respect to the simple text authentication. In this section, you will authenticate OSPF routing information between router r2 and router r3 using the MD5 encryption standard. Moreover, you will authenticate OSPF routing information based on the area where the routers are located.

**Step 1.** In router r2 issue the following command to enter configuration mode.

```
configure terminal
```



```

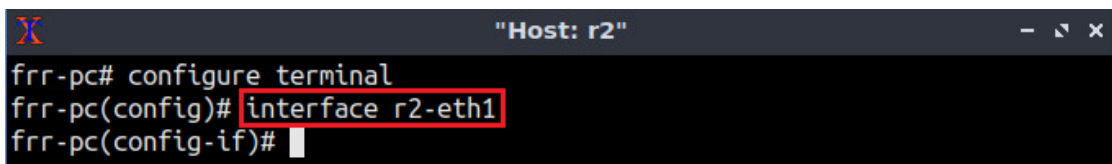
X "Host: r2"
frr-pc# configure terminal
frr-pc(config)#

```

Figure 48. Entering configuration mode in router r2.

**Step 2.** In this step, you will configure interface *r2-eth1* to participate in the authentication process. Issue the following command to enter *r2-eth1* configuration.

```
interface r2-eth1
```



```

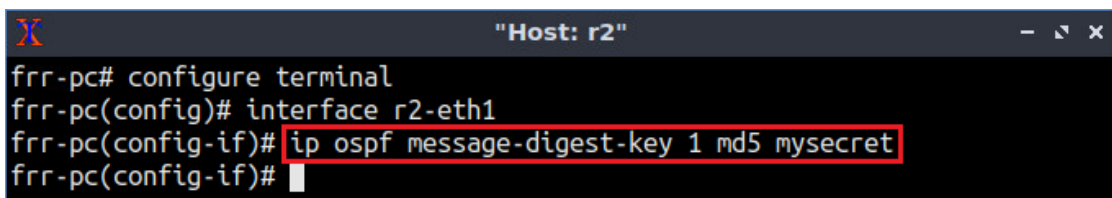
X "Host: r2"
frr-pc# configure terminal
frr-pc(config)# interface r2-eth1
frr-pc(config-if)#

```

Figure 49. Configuring OSPF in router r3.

**Step 3.** Issue the following command to assign a message digest key using `md5`. Notice that the key number is `1` and the password `mysecret`.

```
ip ospf message-digest-key 1 md5 mysecret
```



```

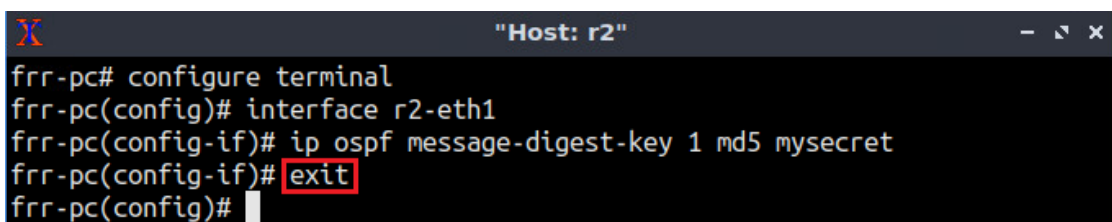
X "Host: r2"
frr-pc# configure terminal
frr-pc(config)# interface r2-eth1
frr-pc(config-if)# ip ospf message-digest-key 1 md5 mysecret
frr-pc(config-if)#

```

Figure 50. Assign a message digest key to the interface *r1-eth0*.

**Step 4.** Type the following command to exit from the interface configuration.

```
exit
```



```

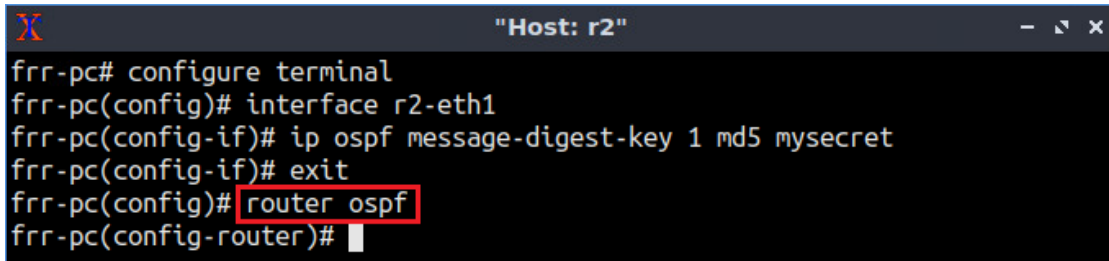
X "Host: r2"
frr-pc# configure terminal
frr-pc(config)# interface r2-eth1
frr-pc(config-if)# ip ospf message-digest-key 1 md5 mysecret
frr-pc(config-if)# exit
frr-pc(config)#

```

Figure 51. Exiting from interface configuration.

**Step 5.** Issue the following command to enter OSPF configuration.

```
router ospf
```

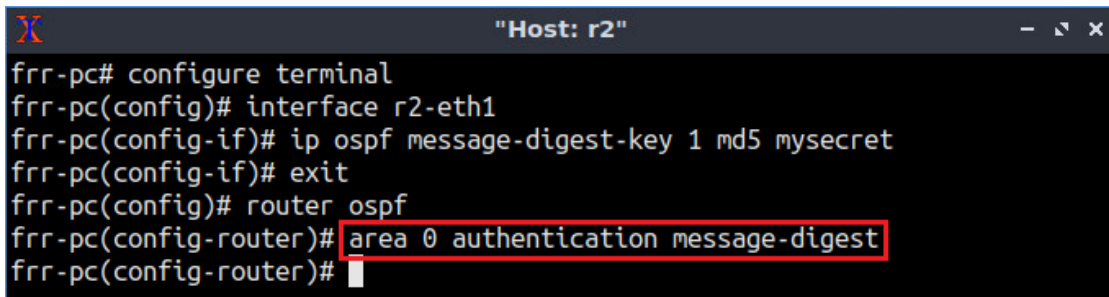


```
frr-pc# configure terminal
frr-pc(config)# interface r2-eth1
frr-pc(config-if)# ip ospf message-digest-key 1 md5 mysecret
frr-pc(config-if)# exit
frr-pc(config)# router ospf
frr-pc(config-router)#
```

Figure 52. Entering OSPF configuration.

**Step 6.** Type the following command to enable OSPF area-based authentication in router r1.

```
area 0 authentication message-digest
```

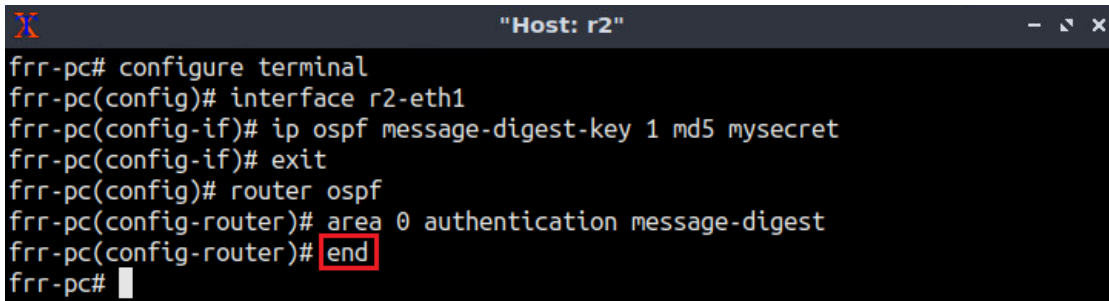


```
frr-pc# configure terminal
frr-pc(config)# interface r2-eth1
frr-pc(config-if)# ip ospf message-digest-key 1 md5 mysecret
frr-pc(config-if)# exit
frr-pc(config)# router ospf
frr-pc(config-router)# area 0 authentication message-digest
frr-pc(config-router)#
```

Figure 53. Enabling OSPF area-based authentication.

**Step 7.** Type the following command to exit configuration mode.

```
end
```



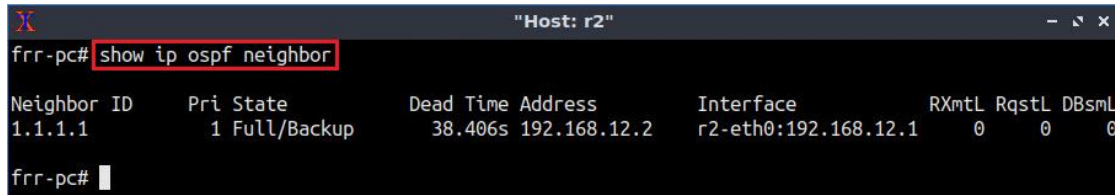
```
frr-pc# configure terminal
frr-pc(config)# interface r2-eth1
frr-pc(config-if)# ip ospf message-digest-key 1 md5 mysecret
frr-pc(config-if)# exit
frr-pc(config)# router ospf
frr-pc(config-router)# area 0 authentication message-digest
frr-pc(config-router)# end
frr-pc#
```

Figure 54. Exiting from configuration mode.

## 5.1 Verifying the configuration

**Step 1.** Issue the following command to verify the OSPF neighbors in router r2.

```
show ip ospf neighbor
```



```

frr-pc# show ip ospf neighbor
Neighbor ID    Pri State           Dead Time Address           Interface           RXmtL RqstL DBsmL
1.1.1.1        1 Full/Backup      38.406s 192.168.12.2      r2-eth0:192.168.12.1 0      0      0
frr-pc#

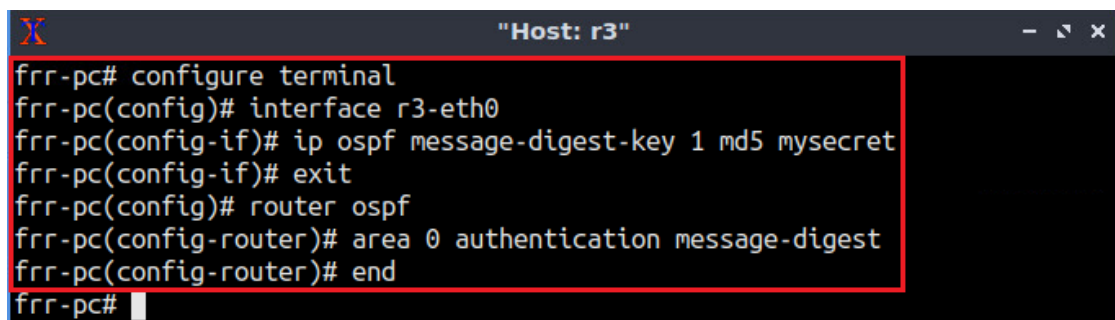
```

Figure 55. Displaying OSPF neighbors.

The figure above shows that router r2 cannot accept routing information from a peer (i.e., router r3 with neighbor ID 3.3.3.3) that is not authenticated. A similar result is observed in the OSPF neighbor table of router r3, where the routing information of router r2 (with a neighbor ID 2.2.2.2) is also missing.

Notice that it takes around 40 seconds to refresh the OSPF neighbor table.

**Step 2.** In router r3, proceed by issuing the commands summarized in the table below to allow routers r2 and r3 exchanging routing information.



```

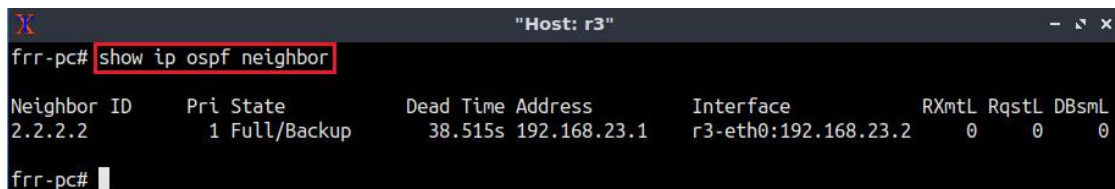
frr-pc# configure terminal
frr-pc(config)# interface r3-eth0
frr-pc(config-if)# ip ospf message-digest-key 1 md5 mysecret
frr-pc(config-if)# exit
frr-pc(config)# router ospf
frr-pc(config-router)# area 0 authentication message-digest
frr-pc(config-router)# end
frr-pc#

```

Figure 56. Router r3 configuration summary.

**Step 3.** In router r3 terminal, issue the command below.

```
show ip ospf neighbor
```



```

frr-pc# show ip ospf neighbor
Neighbor ID    Pri State           Dead Time Address           Interface           RXmtL RqstL DBsmL
2.2.2.2        1 Full/Backup      38.515s 192.168.23.1      r3-eth0:192.168.23.2 0      0      0
frr-pc#

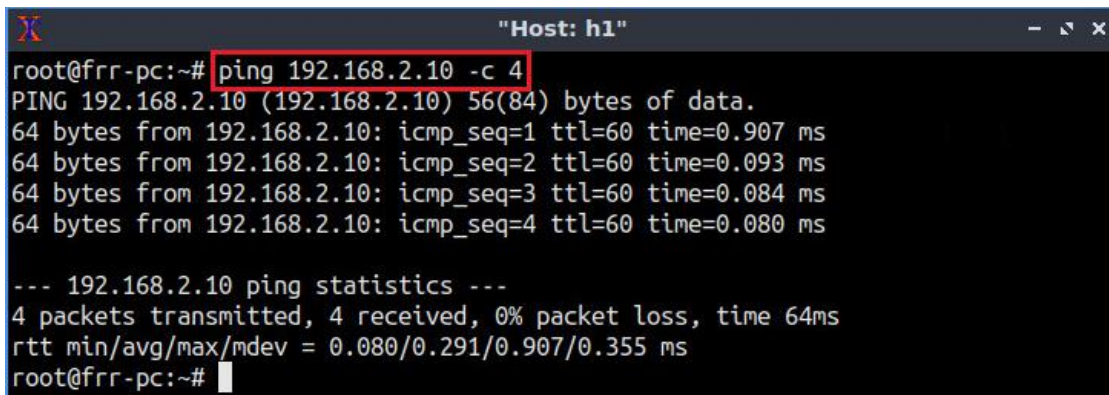
```

Figure 57. Displaying OSPF neighbors.

In the figure above you will observe that router r2 appears now as a neighbor of router r3. This result means routers r2 and r3 can now exchange routing information.

**Step 4.** Perform a connectivity test by issuing the following command. The result will show a successful connectivity test.

```
ping 192.168.2.10 -c 4
```

A terminal window titled "Host: h1" showing a ping command and its output. The command is "ping 192.168.2.10 -c 4", which is highlighted with a red box. The output shows four successful ping requests with varying response times and a summary of the statistics.

```
root@frr-pc:~# ping 192.168.2.10 -c 4
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=60 time=0.907 ms
64 bytes from 192.168.2.10: icmp_seq=2 ttl=60 time=0.093 ms
64 bytes from 192.168.2.10: icmp_seq=3 ttl=60 time=0.084 ms
64 bytes from 192.168.2.10: icmp_seq=4 ttl=60 time=0.080 ms

--- 192.168.2.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 64ms
rtt min/avg/max/mdev = 0.080/0.291/0.907/0.355 ms
root@frr-pc:~#
```

Figure 58. Performing a connectivity test between host h1 and host h2.

If there is a conflict between the interface-based and area-based mode, the interface-based authentication type takes precedence.

This concludes lab 7. Stop the emulation and then exit out of MiniEdit.

## References

1. Moy, J. "RFC 1247: OSPF version 2." 1991.
2. Moy, J. "RFC 2328: OSPF version 2." 1998.
3. Coltun, R., Ferguson, D., Moy, J., Lindem, A. "RFC 5340, OSPF for IPv6." 2008.
4. Peterson, L. "Computer networks: a systems approach." 6th Edition, 2021.
5. Tanenbaum, D. Wetherall. "Computer networks." 5th Edition. 2012.
6. J. Kurose, K. Ross. "Computer networking, a top-down approach." 7th Edition. 2017.
7. Teare, D., Vachon, B., Graziani, R. "Implementing Cisco IP routing (ROUTE) foundation learning guide: (CCNP ROUTE 300-101)." 2014.
8. Linux foundation collaborative projects. "FRR routing documentation." 2017 [Online] Available: <http://docs.frrouting.org/en/latest/>.
9. Ronald, R. "RFC 1321: The MD5 message-digest algorithm." 1992.



UNIVERSITY OF  
**SOUTH CAROLINA**

## OPEN SHORTEST PATH FIRST

### Lab 8: Setting OSPFv2 Bandwidth and Costs

Document Version: **10-22-2021**



Award 1829698

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

## Contents

Overview .....	3
Objectives.....	3
Lab settings .....	3
Lab roadmap .....	3
1 Introduction .....	3
1.1 OSPF path calculation.....	4
1.2 OSPF automatic cost calculation, reference bandwidth update and cost overriding.....	5
2 Lab topology.....	5
2.1 Lab settings.....	6
2.2 Loading the topology.....	6
2.3 Load the zebra daemon and verify the configuration .....	9
3 Configuring single-area OSPF.....	12
3.1 Configuring OSPF in routers r1, r2, and r3 .....	12
3.2 Verifying routing information and connectivity.....	17
4 Setting interface and reference bandwidth.....	19
4.1 Setting OSPF reference bandwidth .....	19
4.2 Setting bandwidth information to the interfaces of router r1 .....	20
4.3 Verifying interfaces costs and bandwidth.....	22
5 Overriding OSPF costs.....	24
5.1 Setting the path cost to router r1 interfaces .....	24
5.2 Verifying interfaces costs and bandwidth.....	25
5.3 Changing route cost in an intermediate router .....	28
References .....	29



## Overview

This lab introduces Open Shortest Path First (OSPF) bandwidth and costs, which are metrics used to determine the best path to a destination network. The lab conducts the reader through the steps to configure the interface bandwidth and override the cost resulting from the bandwidth calculation. The user will verify the routing tables and perform throughput tests to validate the results.

## Objectives

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

1. Understand the concept of OSPF bandwidth and costs.
2. Configure single-area OSPF.
3. Establish the interfaces' bandwidth of a router.
4. Understand how OSPF calculates an interface's cost.
5. Override the costs resulting from the bandwidth.
6. Perform a throughput test to validate the configuration.

## Lab settings

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

Table 1. Credentials to access the Client 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 single-area OSPF.
4. Section 4: Setting interface and reference bandwidth.
5. Section 5: Overriding OSPF costs.

## 1 Introduction

## 1.1 OSPF path calculation

The OSPF protocol calculates the shortest path from a source to a destination network by running the Shortest Path First (SPF) algorithm, also known as the Dijkstra algorithm. Each router participating in the OSPF process has the topology table (i.e., Link-State Database (LSDB)), which contains all the information about the topology in an OSPF area. The information in the topology table contains the costs to reach the directly attached network.

The SPF algorithm computes these costs and the OSPF protocol then chooses the route with the lowest cost. Notice that the SPF algorithm will be executed if there is any change in the network topology within an OSPF area (i.e., intra-area changes). The SPF algorithm does not rerun if there are changes in the topology corresponding to another area (i.e., inter-area changes).

Consider Figure 1. The topology comprises routers participating in the OSPF process including the networks and the costs associated with each path. For example, to reach the network 192.168.2.0/24 from the network 192.168.1.0/24, there are two paths. The first path is r1-> r2 -> r3 and the other one is r1 -> r4 -> r3. The OSPF protocol runs the SPF algorithm to select the best path by creating an SPF tree with the information extracted from the topology table. The best path will be present in the routing table which in this case is r1 -> r4 -> r3 and has a cost of 28.

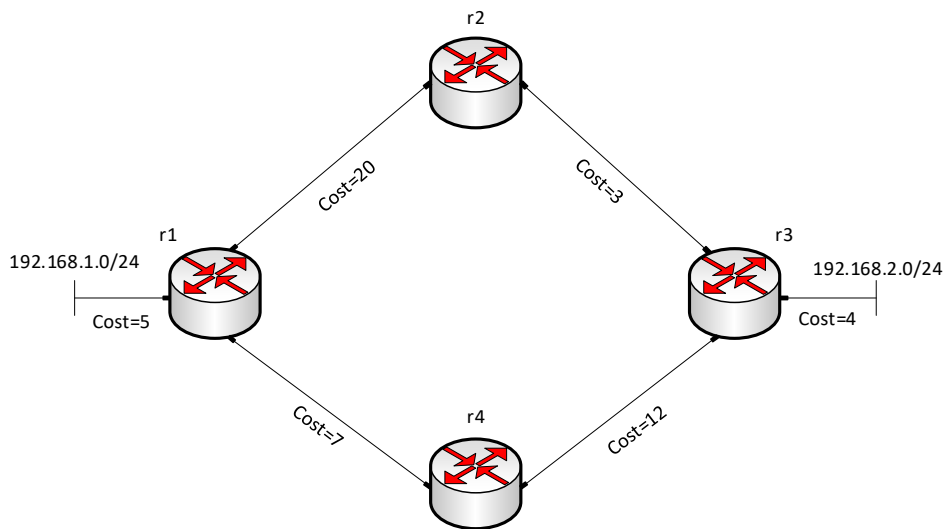


Figure 1. Path's cost calculation.

To reach the network 192.168.2.0/24 from the network 192.168.1.0/24. The OSPF protocol runs the SPF algorithm to select the best path by creating an SPF tree with the information extracted from the topology table. Then, route with the lowest cost will be selected.

## 1.2 OSPF automatic cost calculation, reference bandwidth update and cost overriding

The OSPF protocol calculates the path's cost as the inverse function of the interface bandwidth as shown in the following equation:

$$Cost = \frac{Reference\ Bandwidth}{Interface\ Bandwidth} = \frac{100\ Mbps}{Interface\ Bandwidth} \quad (1)$$

By default, the reference bandwidth is 100 Mbps (100,000,000 bps). Therefore, if the bandwidth of an interface is 10 Mbps, the cost will be 10. If the interface bandwidth is 100 Mbps or higher, the cost will be 1. Considering that current links speed can go up to 100 Gbps and beyond, the OSPF protocol allows updating the reference bandwidth to accommodate to higher interface bandwidths. Moreover, when configuring routers within an OSPF domain, it is a good practice setting the same reference bandwidth in all of them. This measure will guarantee that the cost calculation will be consistent within the OSPF domain, thus avoiding unexpected routing behaviors. Notice that the interface bandwidth value is just a descriptive and it does not set the interface's rate.

There might be also cases when the cost of a link is determined by factors other than the bandwidth. For that purpose, the OSPF protocol also allows configuring arbitrary costs in the interfaces. This value overrides the cost resulting from (1).

## 2 Lab topology

Consider Figure 2. The topology comprises four routers, two switches and two end hosts. The routers participate in the OSPF process in area 0. In this lab, you will modify the costs associated to each path and observe how the OSPF protocol always selects the path with lowest cost.

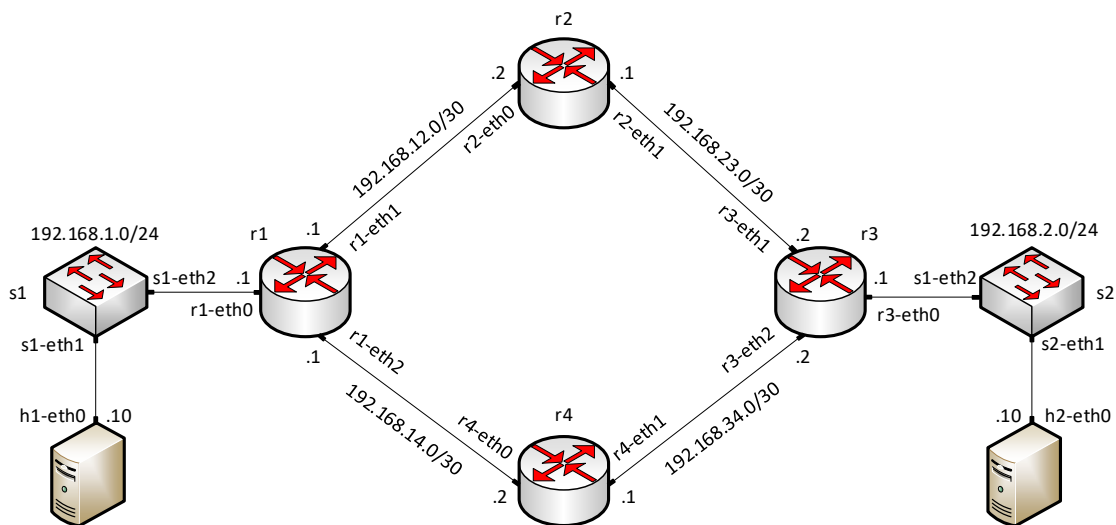


Figure 2. Lab topology.

## 2.1 Lab settings

Table 2 contains information about the devices and their corresponding interfaces, IPv4 addresses, subnet, and default gateway.

Table 2. Topology Information.

Device	Interface	IPv4 Address	Subnet	Default Gateway
Router 1 (r1)	r1-eth0	192.168.1.1	/24	N/A
	r1-eth1	192.168.12.1	/30	N/A
	r1-eth2	192.168.14.1	/30	N/A
Router 2 (r2)	r2-eth0	192.168.12.2	/30	N/A
	r2-eth1	192.168.23.1	/30	N/A
Router 3 (r3)	r3-eth0	192.168.2.1	/24	N/A
	r3-eth1	192.168.23.2	/30	N/A
	r3-eth2	192.168.34.2	/30	N/A
Router 4 (r4)	r4-eth0	192.168.14.2	/30	N/A
	r4-eth1	192.168.34.1	/30	N/A
Host 1 (h1)	h1-eth0	192.168.1.10	/24	192.168.1.1
Host 2 (h2)	h2-eth0	192.168.2.10	/24	192.168.2.1

## 2.2 Loading the topology

**Step 1.** Launch MiniEdit by double clicking on the shortcut located on the desktop. When prompted for a password, type `password`.

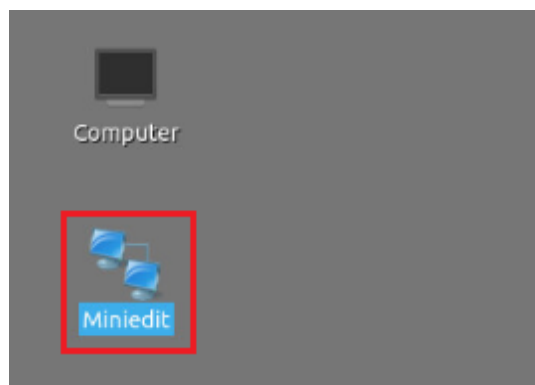


Figure 3. MiniEdit shortcut.

**Step 2.** In MiniEdit's menu bar, click on *File* then *Open* to load the lab topology. Locate the *lab8.mn* topology file in the default directory, */home/frr/OSPF\_Labs/lab8* and click on *Open*.

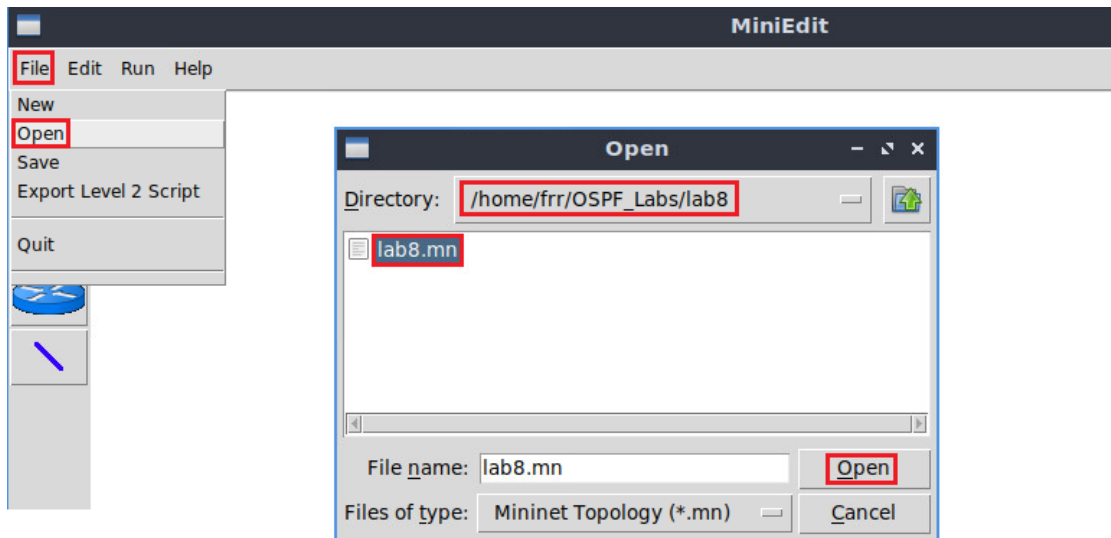


Figure 4. MiniEdit's open dialog.

**Step 3.** Click on the *Run* button located on the lower left-hand side to run the emulation.



Figure 5. Starting the emulation.

At this point, the topology is loaded with all the required network components. Next, you will execute a script that will configure the interfaces of the routers.

Automating the configuration of interfaces with the script is intended to facilitate the lab experience and to focus on the OSPF experiment.

**Step 4.** Open the Linux terminal by clicking on the icon located in the taskbar.

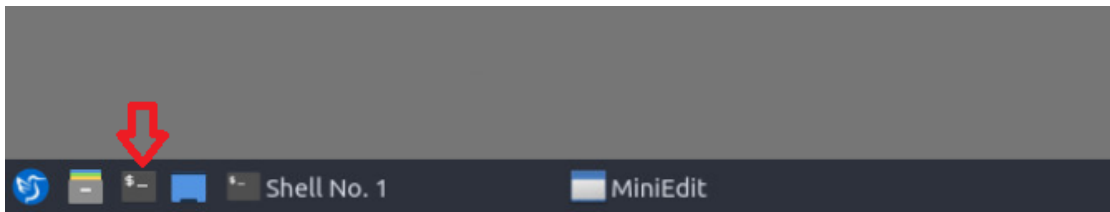


Figure 6. Opening Linux terminal.

**Step 5.** Issue the command shown below to set the lab environment. This command configures the IP addresses of the routers' interface according to Table 2. If a password is required, type `password`.

```
sudo set_env lab8
```

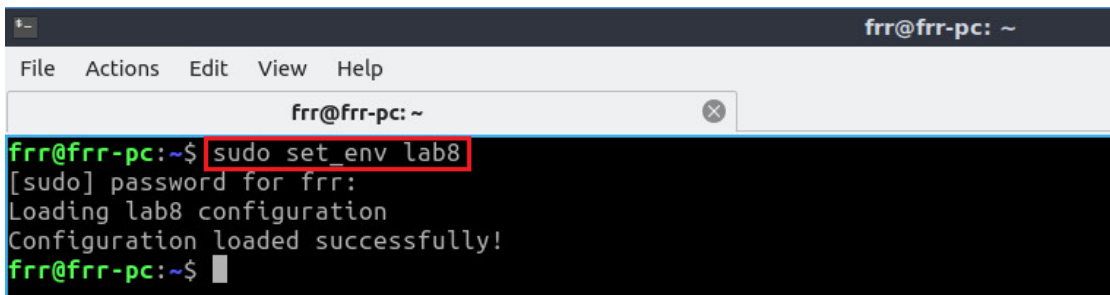


Figure 7. Setting the lab environment.

**Step 6.** Type the following command to exit the Linux terminal.

```
exit
```

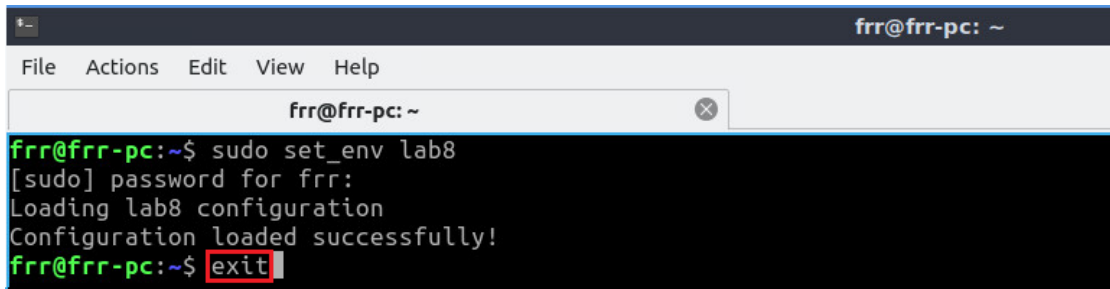


Figure 8. Exiting the Linux terminal.

**Step 7.** Click on Mininet's terminal, i.e., the one launched when MiniEdit was started.

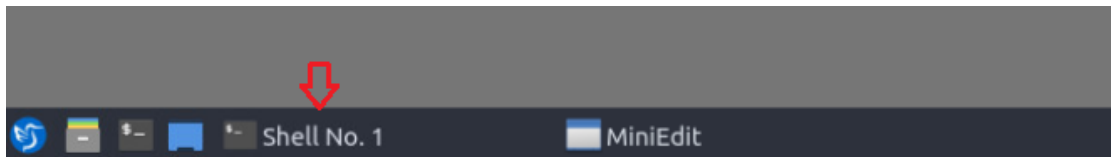


Figure 9. Opening Mininet's terminal.

**Step 8.** Issue the following command to display the interface names and connections.

```
links
```

```

mininet> links
s1-eth1<->h1-eth0 (OK OK)
s2-eth1<->h2-eth0 (OK OK)
s1-eth2<->r1-eth0 (OK OK)
r3-eth0<->s2-eth2 (OK OK)
r1-eth1<->r2-eth0 (OK OK)
r1-eth2<->r4-eth0 (OK OK)
r2-eth1<->r3-eth1 (OK OK)
r4-eth1<->r3-eth2 (OK OK)
mininet>

```

Figure 10. Displaying network interfaces.

In the figure above, the link displayed within the gray box indicates that interface *eth2* of switch *s1* connects to interface *eth0* of router *r1* (i.e., *s1-eth2<->r1-eth0*).

### 2.3 Load the zebra daemon and verify the configuration

In this section, you will verify that the IP addresses listed in Table 2 and inspect the routing table of the routers.

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

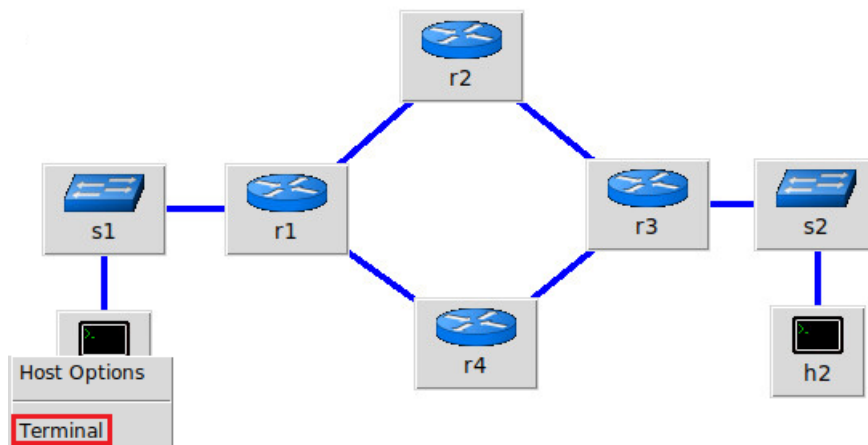


Figure 11. Opening host *h1* terminal.

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

```
ifconfig
```

```

Host: h1
root@frr-pc:~# ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
  inet 192.168.1.10 netmask 255.255.255.0 broadcast 192.168.1.255
  inet6 fe80::7c11:30ff:fea5:d022 prefixlen 64 scopeid 0x20<link>
  ether 7e:11:30:a5:d0:22 txqueuelen 1000 (Ethernet)
  RX packets 32 bytes 3781 (3.7 KB)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 12 bytes 936 (936.0 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@frr-pc:~#

```

Figure 12. Displaying host h1 interfaces and IP addresses.

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

```
route
```

```

Host: h1
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
  inet 192.168.1.10 netmask 255.255.255.0 broadcast 192.168.1.255
  inet6 fe80::7c11:30ff:fea5:d022 prefixlen 64 scopeid 0x20<link>
  ether 7e:11:30:a5:d0:22 txqueuelen 1000 (Ethernet)
  RX packets 32 bytes 3781 (3.7 KB)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 12 bytes 936 (936.0 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@frr-pc:~# 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@frr-pc:~#

```

Figure 13. Displaying host h1 default route.



**Step 4.** To verify the hosts h2 and h3 default route, proceed by following from step 1 to step 3. Similar results should be observed.

**Step 5.** You will validate that the router interfaces are configured according to Table 2. To proceed, hold right-click on router r1 and select *Terminal*.

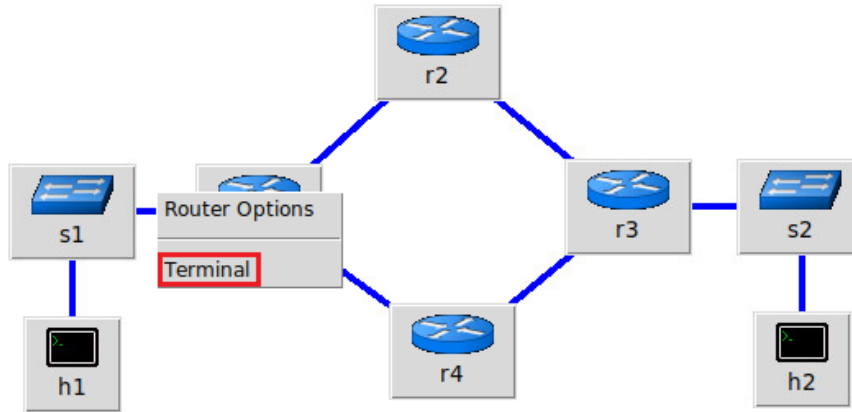


Figure 14. Opening a terminal on router r1.

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

```
zebra
```

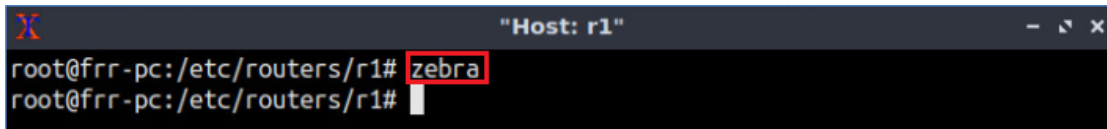


Figure 15. Starting zebra daemon.

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

```
vtysh
```

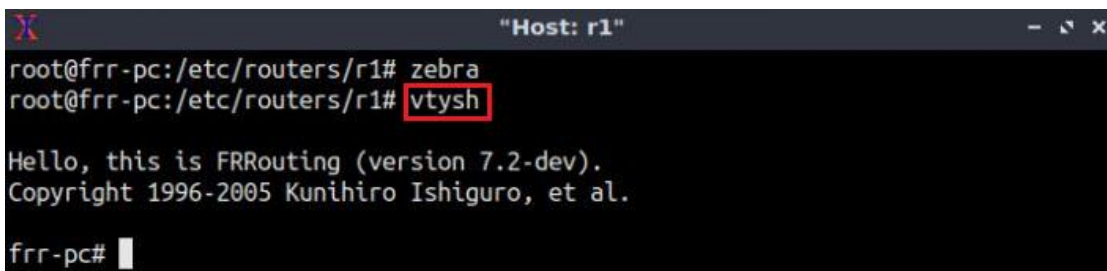


Figure 16. Starting vtysh on router r1.

**Step 8.** Type the following command on router r1 terminal to verify the routing table. It will list all the directly connected networks. Notice that the routing table of router r1 does not contain any route to the network connected to router r3 (192.168.2.0/24) as there is no routing protocol configured yet.

```
show ip route
```

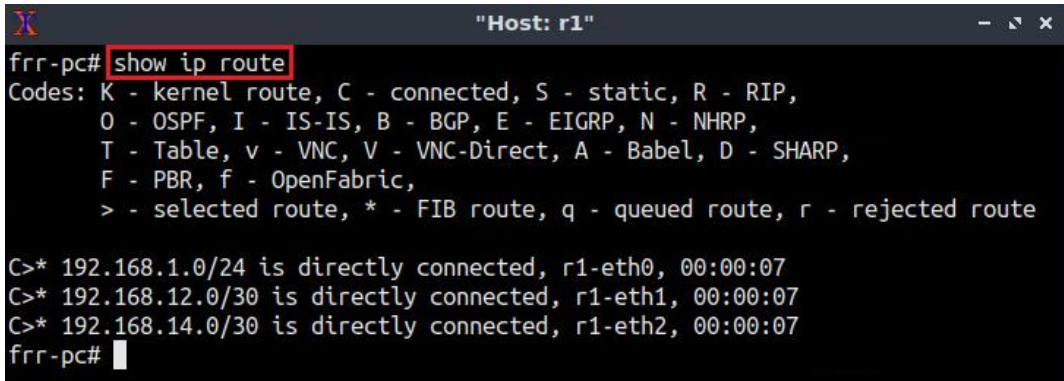


Figure 17. Displaying routing table of router r1.

**Step 9.** To verify the routing table of the other routers, proceed similarly by following from step 5 to step 8 on the corresponding router terminal.

### 3 Configuring single-area OSPF

This section shows the steps to configure single-area OSPF, meaning that all the OSPF participants will form part of the backbone area (i.e., area 0).

#### 3.1 Configuring OSPF in routers r1, r2, and r3

**Step 1.** Open router r1's terminal by selecting the icon from the taskbar as shown below.



Figure 18. Opening router r1's terminal.

**Step 2.** In router r1's terminal, type the command below to exit the vtysh session.

```
exit
```

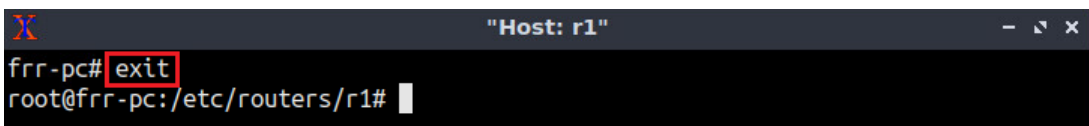
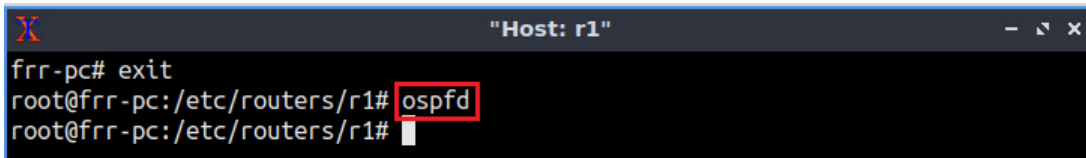


Figure 19. Exiting the vtysh session.

**Step 3.** Enable and start the OSPF routing protocol by typing the following command:

```
ospfd
```

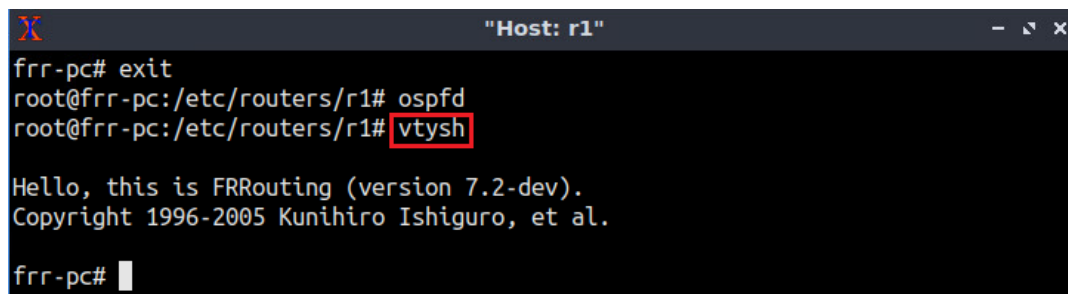


```
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1#
```

Figure 20. Starting the OSPF daemon.

**Step 4.** To enter router r1's terminal, type the following command:

```
vttysh
```



```
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vttysh

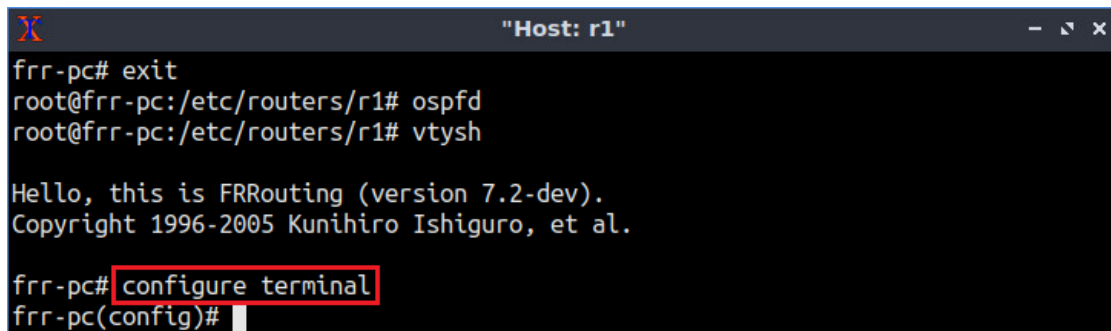
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc#
```

Figure 21. Starting vttysh in router r1.

**Step 5.** At this point, you have entered router r1's terminal, to enter configuration mode in router r1, issue the following command:

```
configure terminal
```



```
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vttysh

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

frr-pc# configure terminal
frr-pc(config)#
```

Figure 22. Entering configuration mode in router r1.

**Step 6.** To configure OSPF, type the command shown below. This command will enable OSPF configuration mode.

```
router ospf
```

```

Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)#

```

Figure 23. Configuring OSPF in router r1.

**Step 7.** To establish router r1 ID, issue the following command:

```
ospf router-id 1.1.1.1
```

```

Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 1.1.1.1
frr-pc(config-router)#

```

Figure 24. Assigning the OSPF router ID of 1.1.1.1 to router r1.

**Step 8.** Type the following command to enable interface *r1-eth1* to participate in the OSPF process of area 0.

```
network 192.168.12.0/30 area 0
```

```

Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

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

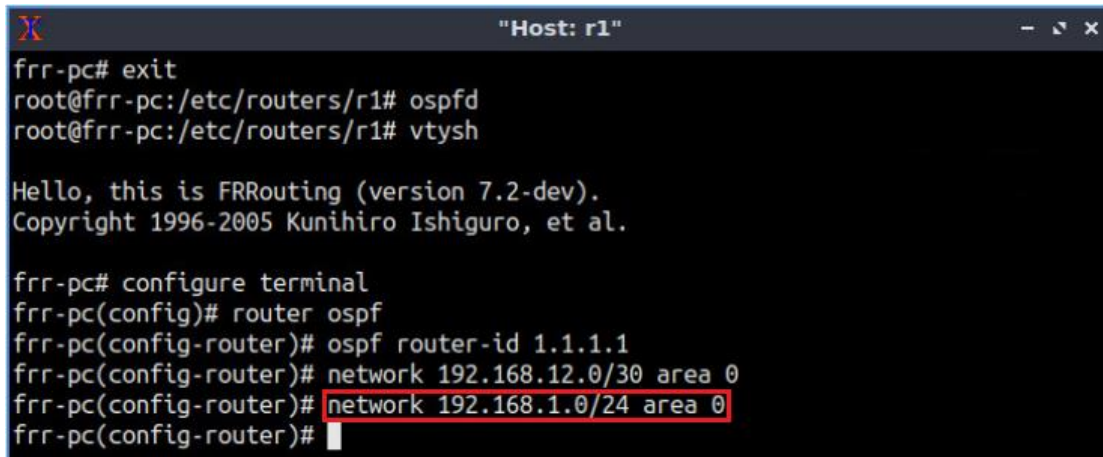
frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 1.1.1.1
frr-pc(config-router)# network 192.168.12.0/30 area 0
frr-pc(config-router)#

```

Figure 25. Enabling interface *r1-eth1* to participate in the OSPF process in area 0.

**Step 9.** Type the following command to enable interface *r1-eth0* to participate in the OSPF process of area 0.

```
network 192.168.1.0/24 area 0
```

A terminal window titled "Host: r1" showing the configuration of OSPF on interface r1-eth0. The user enters 'exit' to leave the root shell, then 'ospfd' to start the daemon, and 'vtysh' to enter configuration mode. After 'configure terminal', they enter 'router ospf', 'ospf router-id 1.1.1.1', 'network 192.168.12.0/30 area 0', and finally 'network 192.168.1.0/24 area 0', which is highlighted with a red box. The prompt returns to 'frr-pc(config-router)#'.

```
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

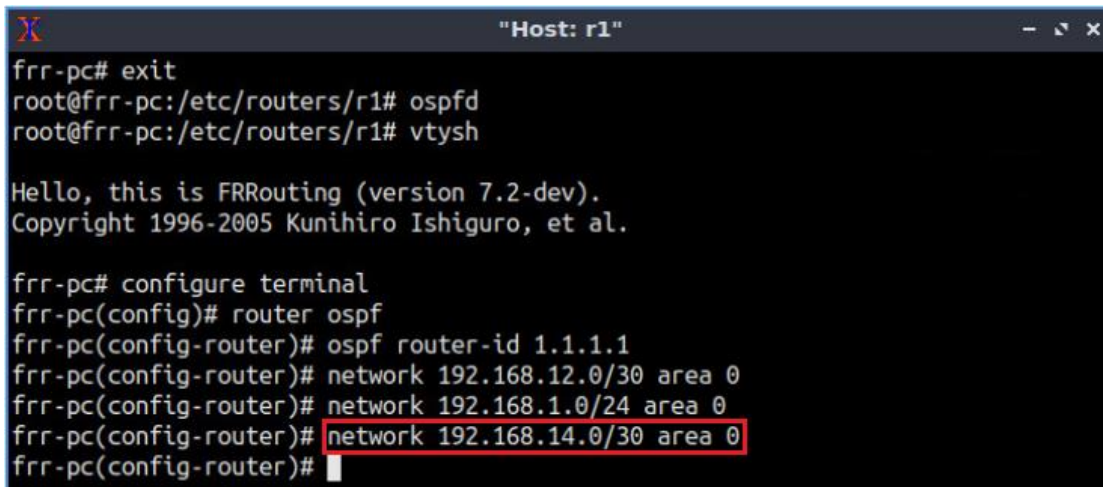
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 1.1.1.1
frr-pc(config-router)# network 192.168.12.0/30 area 0
frr-pc(config-router)# network 192.168.1.0/24 area 0
frr-pc(config-router)#
```

Figure 26. Enabling interface *r1-eth0* to participate in the OSPF process in area 0.

**Step 10.** Type the following command to enable interface *r1-eth2* to participate in the OSPF process of area 0.

```
network 192.168.14.0/30 area 0
```

A terminal window titled "Host: r1" showing the configuration of OSPF on interface r1-eth2. The user enters 'exit' to leave the root shell, then 'ospfd' to start the daemon, and 'vtysh' to enter configuration mode. After 'configure terminal', they enter 'router ospf', 'ospf router-id 1.1.1.1', 'network 192.168.12.0/30 area 0', 'network 192.168.1.0/24 area 0', and finally 'network 192.168.14.0/30 area 0', which is highlighted with a red box. The prompt returns to 'frr-pc(config-router)#'.

```
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 1.1.1.1
frr-pc(config-router)# network 192.168.12.0/30 area 0
frr-pc(config-router)# network 192.168.1.0/24 area 0
frr-pc(config-router)# network 192.168.14.0/30 area 0
frr-pc(config-router)#
```

Figure 27. Enabling interface *r1-eth2* to participate in the OSPF process in area 0.

**Step 11.** Type the following command to exit router configuration mode.

```
end
```

```

Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 1.1.1.1
frr-pc(config-router)# network 192.168.12.0/30 area 0
frr-pc(config-router)# network 192.168.1.0/24 area 0
frr-pc(config-router)# network 192.168.14.0/30 area 0
frr-pc(config-router)# end
frr-pc#

```

Figure 28. Exiting router configuration mode.

Consider the figure above. After configuring OSPF in router r1, the interfaces *r1-eth0*, *r1-eth1*, and *r1-eth2* will participate in the routing process in area 0.

**Step 12.** 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 terminal, issue the commands summarized below.

```

Host: r2
frr-pc# exit
root@frr-pc:/etc/routers/r2# ospfd
root@frr-pc:/etc/routers/r2# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 2.2.2.2
frr-pc(config-router)# network 192.168.12.0/30 area 0
frr-pc(config-router)# network 192.168.23.0/30 area 0
frr-pc(config-router)# end
frr-pc#

```

Figure 29. Router r2 configuration summary.

Consider the above figure. After configuring OSPF in router r2, the interfaces *r2-eth0*, and *r2-eth1* will participate in the routing process of area 0.

**Step 13.** Router r3 is configured similarly to router r2 but with different IP addresses (see Table 2). Those steps are summarized in the following figure. To proceed, in router r3 terminal, issue the commands depicted below.

```

Host: r3
frr-pc# exit
root@frr-pc:/etc/routers/r3# ospfd
root@frr-pc:/etc/routers/r3# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 3.3.3.3
frr-pc(config-router)# network 192.168.23.0/30 area 0
frr-pc(config-router)# network 192.168.2.0/24 area 0
frr-pc(config-router)# network 192.168.34.0/30 area 0
frr-pc(config-router)# end
frr-pc#

```

Figure 30. Router r3 configuration summary.

Consider the figure above. After configuring OSPF in router r3, the interfaces *r3-eth0*, *r3-eth1*, and *r3-eth2* will participate in the routing process of area 0.

**Step 14.** Router r4 is configured similarly to router r3 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.

```

Host: r4
frr-pc# exit
root@frr-pc:/etc/routers/r4# ospfd
root@frr-pc:/etc/routers/r4# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 4.4.4.4
frr-pc(config-router)# network 192.168.14.0/30 area 0
frr-pc(config-router)# network 192.168.34.0/30 area 0
frr-pc(config-router)# end
frr-pc#

```

Figure 31. Router r4 configuration summary.

Consider the figure above. After configuring OSPF in router r4, the interfaces *r4-eth0* and *r4-eth1* will participate in the routing process of area 0.

### 3.2 Verifying routing information and connectivity

**Step 1.** Issue the following command to display the routing table of router r2.

```
show ip route
```

```

Host: r1
frr-pc# 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

O 192.168.1.0/24 [110/10] is directly connected, r1-eth0, 00:03:12
C>* 192.168.1.0/24 is directly connected, r1-eth0, 00:07:03
O>* 192.168.2.0/24 [110/30] via 192.168.12.2, r1-eth1, 00:00:12
   * via 192.168.14.2, r1-eth2, 00:00:12
O 192.168.12.0/30 [110/10] is directly connected, r1-eth1, 00:02:24
C>* 192.168.12.0/30 is directly connected, r1-eth1, 00:07:03
O 192.168.14.0/30 [110/10] is directly connected, r1-eth2, 00:02:17
C>* 192.168.14.0/30 is directly connected, r1-eth2, 00:07:03
O>* 192.168.23.0/30 [110/20] via 192.168.12.2, r1-eth1, 00:01:35
O>* 192.168.34.0/30 [110/20] via 192.168.14.2, r1-eth2, 00:00:24
frr-pc#

```

Figure 32. Displaying router r1's routing table.

Consider the figure above. The output displays the networks learned along with additional attributes, such as the routing protocol that discovered the network. For example, networks with the symbol `O` on the left are learned from OSPF, whereas those with symbol `C` are directly connected to the device. Furthermore, the symbol `O>` on the left of the displayed network indicates that this entry is selected to reach the network. As for the symbol `*`, this indicates that the route is inserted in Forwarding Information Base (FIB) of the router.

Notice that there are two routes to reach the network 192.168.2.0/24. The selected route (i.e., denoted by the `O>` symbol) is the one via 192.168.12.2.

**Step 2.** In the host h1 terminal, issue the following command to test the connectivity between host h1 and host h2. You will observe a successful connectivity test.

```
ping 192.168.2.10 -c 4
```

```

Host: h1
root@frr-pc:~# ping 192.168.2.10 -c 4
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=61 time=1.05 ms
64 bytes from 192.168.2.10: icmp_seq=2 ttl=61 time=0.098 ms
64 bytes from 192.168.2.10: icmp_seq=3 ttl=61 time=0.108 ms
64 bytes from 192.168.2.10: icmp_seq=4 ttl=61 time=0.109 ms

--- 192.168.2.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 59ms
rtt min/avg/max/mdev = 0.098/0.340/1.048/0.408 ms
root@frr-pc:~#

```

Figure 33. Testing connectivity from host h1 to host h2.



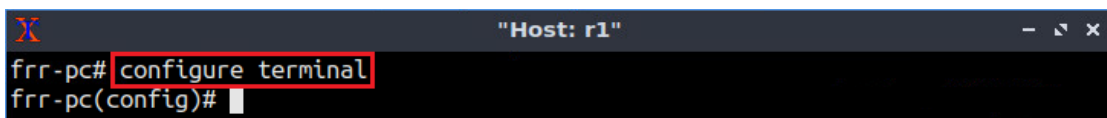
## 4 Setting interface and reference bandwidth

The OSPF protocol derives the cost of a path from the interface bandwidth. The higher the bandwidth the lower the cost. Additionally, OSPF allows the user updating the reference bandwidth which is a good practice considering that networks operate at higher speeds than default reference bandwidth (i.e., 100 Mbps).

### 4.1 Setting OSPF reference bandwidth

**Step 1.** In router r1, issue the following command to enter configuration mode.

```
configure terminal
```

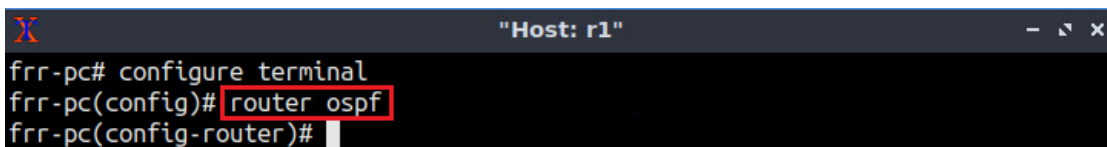


```
X "Host: r1"
frr-pc# configure terminal
frr-pc(config)#
```

Figure 34. Entering configuration mode in router r1.

**Step 2.** To configure OSPF, type the command shown below. This command will enable OSPF configuration mode.

```
router ospf
```

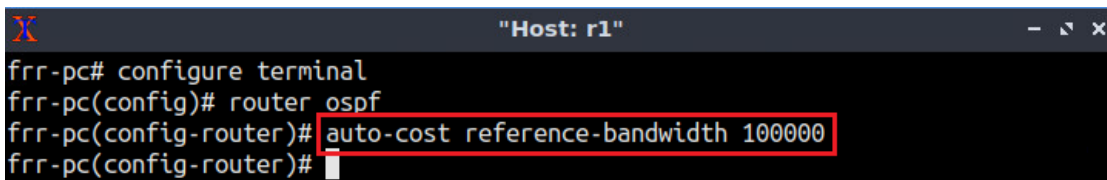


```
X "Host: r1"
frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)#
```

Figure 35. Entering OSPF configuration.

**Step 3.** Issue the following command to establish the reference bandwidth. In this example, the reference bandwidth is set to 100,000 which corresponds to 100 Gbps.

```
auto-cost reference-bandwidth 100000
```



```
X "Host: r1"
frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# auto-cost reference-bandwidth 100000
frr-pc(config-router)#
```

Figure 36. Establishing the reference bandwidth.

By default, the reference bandwidth is 100 Mbps, which means the protocol will not correctly compute costs over this value. Nowadays, links operating at 1 Gbps, 10 Gbps, and even 100 Gbps are more common. Therefore, updating the default reference bandwidth is a good practice to ensure that the protocol is computing the costs correctly.

**Step 4.** Type the following command to exit configuration mode.

```
end
```

```

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# auto-cost reference-bandwidth 100000
frr-pc(config-router)# end
frr-pc#
  
```

Figure 37. Exiting configuration mode.

## 4.2 Setting bandwidth information to the interfaces of router r1

**Step 1.** In router r1 issue the following command to enter configuration mode.

```
configure terminal
```

```

frr-pc# configure terminal
frr-pc(config)#
  
```

Figure 38. Entering configuration mode in router r1.

**Step 2.** Type the following command to enter to interface *r1-eth0* configuration.

```
interface r1-eth1
```

```

frr-pc# configure terminal
frr-pc(config)# interface r1-eth1
frr-pc(config-if)#
  
```

Figure 39. Entering interface *r1-eth0* configuration.

**Step 3.** Type the following command to set the bandwidth to 200 Mbps to interface *r1-eth0*.

```
bandwidth 200
```

```

frr-pc# configure terminal
frr-pc(config)# interface r1-eth1
frr-pc(config-if)# bandwidth 200
frr-pc(config-if)#
  
```

Figure 40. Assigning the bandwidth information to interface *r1-eth0*.

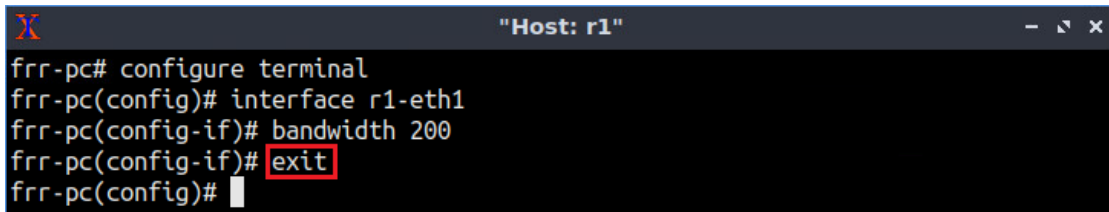
Notice that the bandwidth value assigned to the interface is merely informational and does not actually establish the interface's bandwidth. The bandwidth value is used to calculate the cost that the OSPF protocol will use to determine the best route.

Considering the cost formula explained in the section 1, the cost associated with interface *r1-eth0* will be:

$$Cost_{r1-eth0} = \frac{\text{Reference Bandwidth}}{\text{Interface Bandwidth}} = \frac{100 \text{ Gbps}}{200 \text{ Mbps}} = \frac{100,000 \text{ Mbps}}{200 \text{ Mbps}} = 500$$

**Step 4.** Type the following command to exit from the interface configuration.

```
exit
```



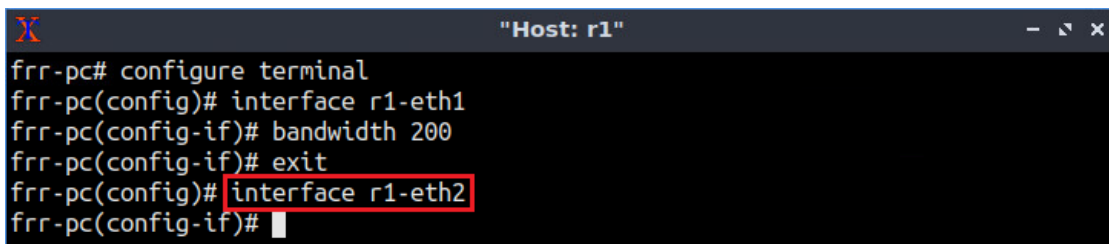
```

frr-pc# configure terminal
frr-pc(config)# interface r1-eth1
frr-pc(config-if)# bandwidth 200
frr-pc(config-if)# exit
frr-pc(config)#
  
```

Figure 41. Exiting from interface configuration.

**Step 5.** Type the following command to enter interface *r1-eth2* configuration.

```
interface r1-eth2
```



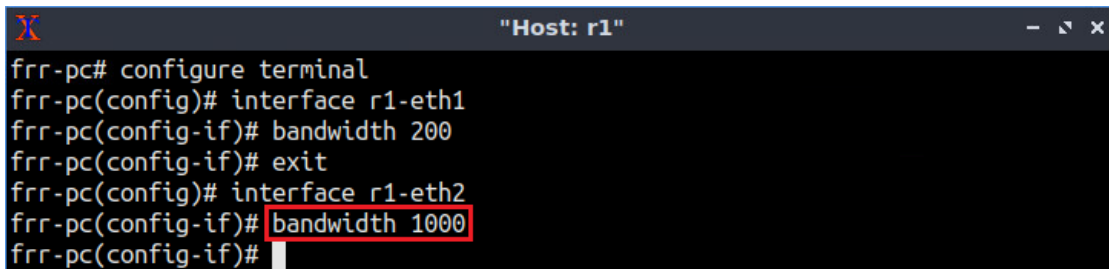
```

frr-pc# configure terminal
frr-pc(config)# interface r1-eth1
frr-pc(config-if)# bandwidth 200
frr-pc(config-if)# exit
frr-pc(config)# interface r1-eth2
frr-pc(config-if)#
  
```

Figure 42. Entering interface *r1-eth2* configuration.

**Step 6.** Type the following command to set the bandwidth to 1 Gbps (i.e., 1000 Mbps) to interface *r1-eth2*.

```
bandwidth 1000
```



```

frr-pc# configure terminal
frr-pc(config)# interface r1-eth1
frr-pc(config-if)# bandwidth 200
frr-pc(config-if)# exit
frr-pc(config)# interface r1-eth2
frr-pc(config-if)# bandwidth 1000
frr-pc(config-if)#
  
```

Figure 43. Assigning the bandwidth information to interface *r1-eth2*.

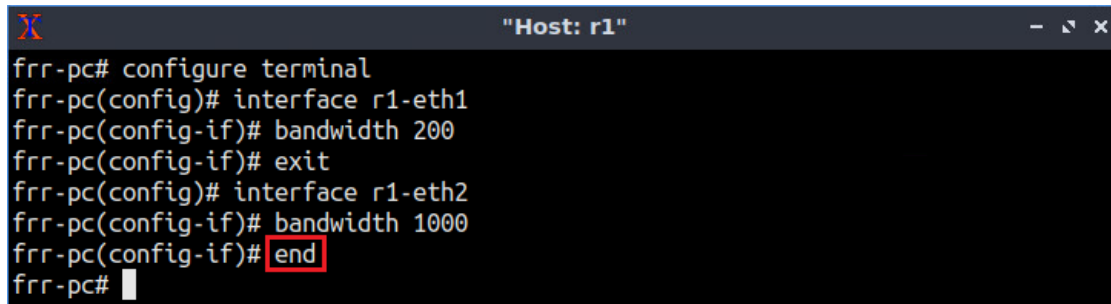
Notice that the bandwidth value assigned to the interface is merely informational and does not actually establish the interface's bandwidth. The bandwidth value is used to calculate the cost that the OSPF protocol will use to determine the best route.

Considering the cost formula explained in the section 1, the cost associated with interface *r1-eth2* will be:

$$Cost_{r1-eth2} = \frac{Reference\ Bandwidth}{Interface\ Bandwidth} = \frac{100\ Gbps}{1\ Gbps} = 100$$

**Step 7.** Type the following command to exit interface configuration.

```
end
```



```

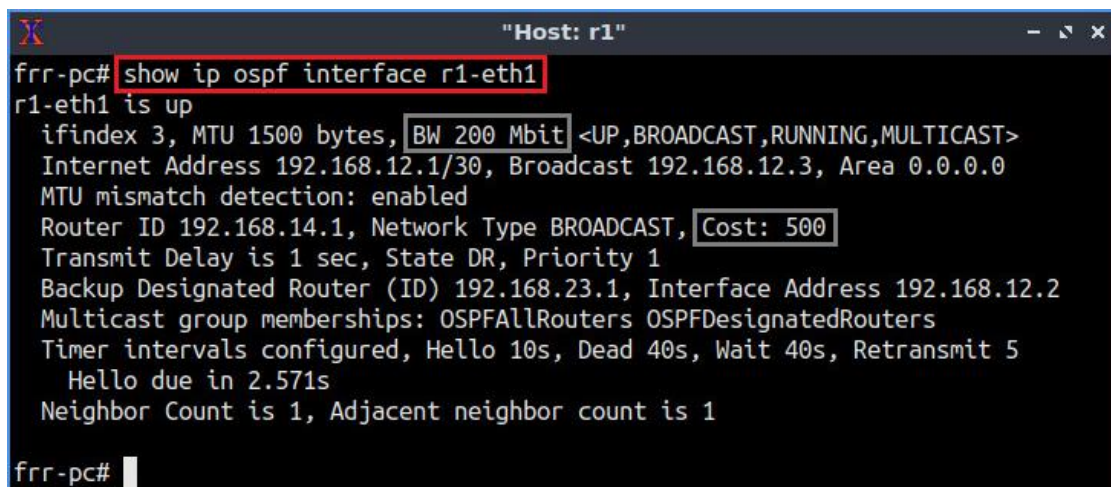
frr-pc# configure terminal
frr-pc(config)# interface r1-eth1
frr-pc(config-if)# bandwidth 200
frr-pc(config-if)# exit
frr-pc(config)# interface r1-eth2
frr-pc(config-if)# bandwidth 1000
frr-pc(config-if)# end
frr-pc#
  
```

Figure 44. Exiting interface configuration mode.

### 4.3 Verifying interfaces costs and bandwidth

**Step 1.** In router *r1*, issue the following command to see interface *r1-eth1* configuration. Notice that the bandwidth is 200 Mbps and the related cost 500.

```
show ip ospf interface r1-eth1
```



```

frr-pc# show ip ospf interface r1-eth1
r1-eth1 is up
  ifindex 3, MTU 1500 bytes, BW 200 Mbit <UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 192.168.12.1/30, Broadcast 192.168.12.3, Area 0.0.0.0
  MTU mismatch detection: enabled
  Router ID 192.168.14.1, Network Type BROADCAST, Cost: 500
  Transmit Delay is 1 sec, State DR, Priority 1
  Backup Designated Router (ID) 192.168.23.1, Interface Address 192.168.12.2
  Multicast group memberships: OSPFAllRouters OSPFDesignatedRouters
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
  Hello due in 2.571s
  Neighbor Count is 1, Adjacent neighbor count is 1

frr-pc#
  
```

Figure 45. Verifying OSPF path cost.

**Step 2.** In router *r1*, issue the following command to see interface *r1-eth2* configuration. Notice that the bandwidth is 1000 Mbps (i.e., 1 Gbps) and the cost 100.

```
show ip ospf interface r1-eth2
```

```

Host: r1
frr-pc# show ip ospf interface r1-eth2
r1-eth2 is up
  ifindex 4, MTU 1500 bytes, BW 1000 Mbit <UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 192.168.14.1/30, Broadcast 192.168.14.3, Area 0.0.0.0
  MTU mismatch detection: enabled
  Router ID 192.168.14.1, Network Type BROADCAST, Cost: 100
  Transmit Delay is 1 sec, State Backup, Priority 1
  Backup Designated Router (ID) 192.168.14.1, Interface Address 192.168.14.1
  Multicast group memberships: OSPFAllRouters OSPFDesignatedRouters
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
    Hello due in 9.716s
  Neighbor Count is 1, Adjacent neighbor count is 1

frr-pc#

```

Figure 46. Verifying OSPF path cost.

**Step 3.** Type the following command to see the routing table of router r1.

```
show ip route
```

```

Host: r1
frr-pc# 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

O 192.168.1.0/24 [110/10] is directly connected, r1-eth0, 00:26:27
C>* 192.168.1.0/24 is directly connected, r1-eth0, 00:30:18
O>* 192.168.2.0/24 [110/120] via 192.168.14.2, r1-eth2, 00:05:26
O 192.168.12.0/30 [110/130] via 192.168.14.2, r1-eth2, 00:05:26
C>* 192.168.12.0/30 is directly connected, r1-eth1, 00:30:18
O 192.168.14.0/30 [110/100] is directly connected, r1-eth2, 00:05:26
C>* 192.168.14.0/30 is directly connected, r1-eth2, 00:30:18
O>* 192.168.23.0/30 [110/120] via 192.168.14.2, r1-eth2, 00:05:26
O>* 192.168.34.0/30 [110/110] via 192.168.14.2, r1-eth2, 00:05:26
frr-pc#

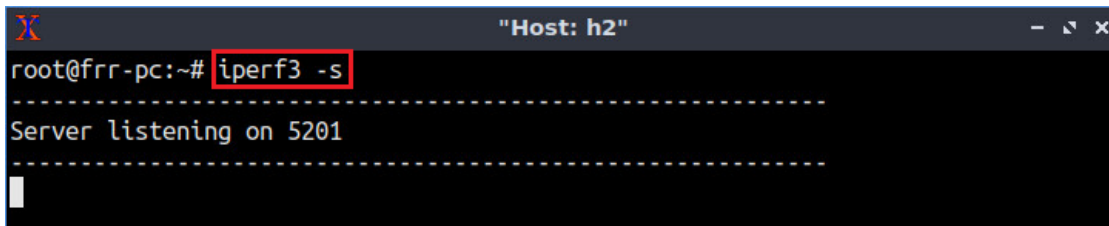
```

Figure 47. Verifying the routing table of router r1.

Notice that the route selected to reach the network 192.168.2.0/24 is via the interface *r1-eth2*, which has the lower cost.

**Step 4.** In host h2 terminal, type the following command to start an iperf3 server. Iperf3 is a tool to measure the maximum achievable bandwidth on IP networks. The argument `-s` specifies that host h2 will run in server mode.

```
iperf3 -s
```



```

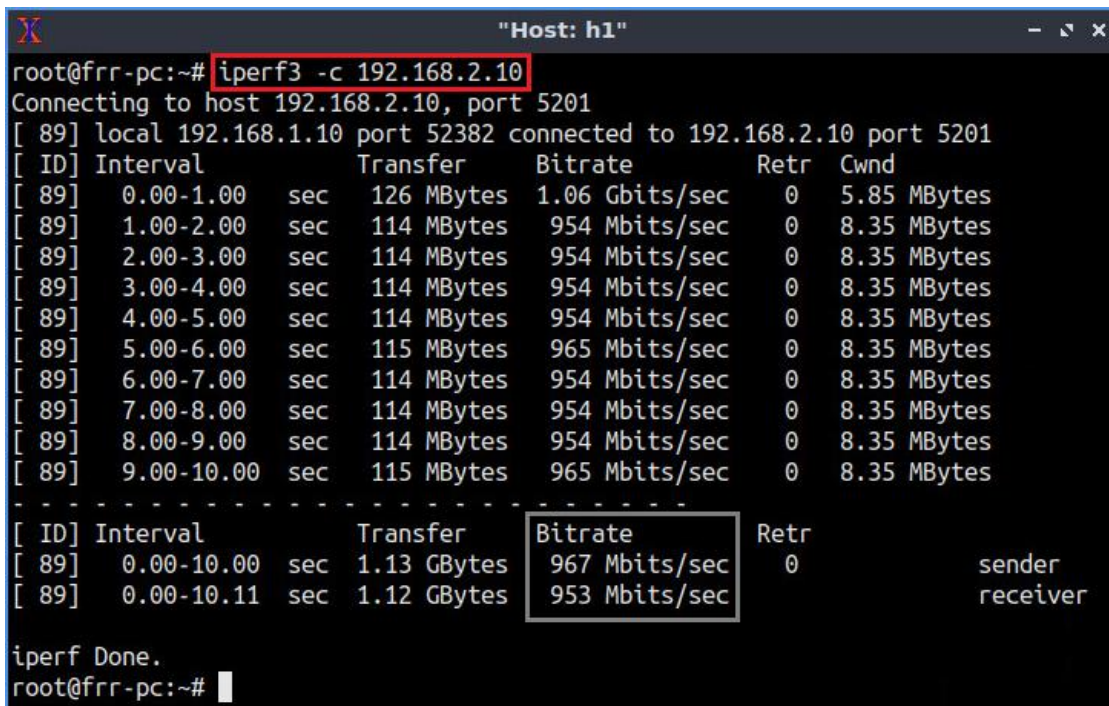
Host: h2
root@frr-pc:~# iperf3 -s
-----
Server listening on 5201
-----

```

Figure 48. Starting an iperf3 server in host h2.

**Step 5.** In host h1 terminal issue the command below to start an iperf3 client. The iperf3 client will start transferring data to host h2 (i.e., the iperf3 server). The argument `-c` specifies that host h1 will be the client followed by host h2 IP address (i.e., 192.168.2.10).

```
iperf3 -c 192.168.2.10
```



```

Host: h1
root@frr-pc:~# iperf3 -c 192.168.2.10
Connecting to host 192.168.2.10, port 5201
[ 89] local 192.168.1.10 port 52382 connected to 192.168.2.10 port 5201
[ ID] Interval          Transfer    Bitrate    Retr  Cwnd
[ 89] 0.00-1.00 sec      126 MBytes  1.06 Gbits/sec    0   5.85 MBytes
[ 89] 1.00-2.00 sec      114 MBytes  954 Mbits/sec    0   8.35 MBytes
[ 89] 2.00-3.00 sec      114 MBytes  954 Mbits/sec    0   8.35 MBytes
[ 89] 3.00-4.00 sec      114 MBytes  954 Mbits/sec    0   8.35 MBytes
[ 89] 4.00-5.00 sec      114 MBytes  954 Mbits/sec    0   8.35 MBytes
[ 89] 5.00-6.00 sec      115 MBytes  965 Mbits/sec    0   8.35 MBytes
[ 89] 6.00-7.00 sec      114 MBytes  954 Mbits/sec    0   8.35 MBytes
[ 89] 7.00-8.00 sec      114 MBytes  954 Mbits/sec    0   8.35 MBytes
[ 89] 8.00-9.00 sec      114 MBytes  954 Mbits/sec    0   8.35 MBytes
[ 89] 9.00-10.00 sec     115 MBytes  965 Mbits/sec    0   8.35 MBytes
-----
[ ID] Interval          Transfer    Bitrate    Retr  Cwnd
[ 89] 0.00-10.00 sec     1.13 GBytes  967 Mbits/sec    0
[ 89] 0.00-10.11 sec     1.12 GBytes  953 Mbits/sec    0
-----
iperf Done.
root@frr-pc:~#

```

Figure 49. Starting an iperf3 client in host h1.

The result observed in the figure above shows that throughput is around 960 Mbps which is close the link bandwidth (i.e., 1 Gbps).

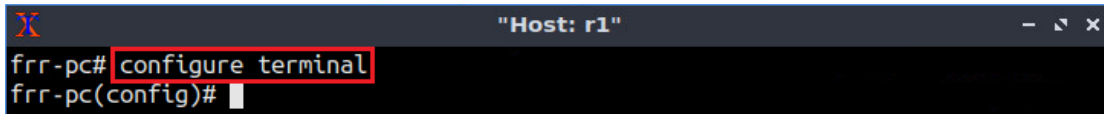
## 5 Overriding OSPF costs

The OSPF protocol provides the option of manually setting the cost associated with a path. This option can be used in cases when the metrics that determine the path's cost area other than the bandwidth.

### 5.1 Setting the path cost to router r1 interfaces

**Step 1.** In router r1, issue the following command to enter configuration mode.

```
configure terminal
```



```

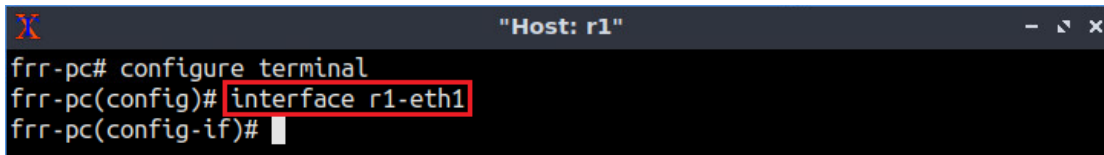
frr-pc# configure terminal
frr-pc(config)#

```

Figure 50. Entering configuration mode in router r1.

**Step 2.** Type the following command to enter to interface *r1-eth1* configuration.

```
interface r1-eth1
```



```

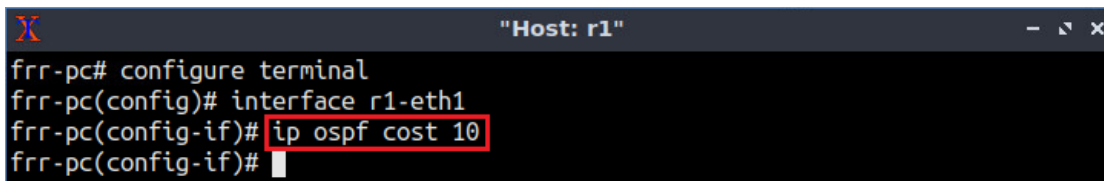
frr-pc# configure terminal
frr-pc(config)# interface r1-eth1
frr-pc(config-if)#

```

Figure 51. Entering interface *r1-eth1* configuration.

**Step 3.** Override the current OSPF calculated cost on the interface *r1-eth1* by issuing the following command.

```
ip ospf cost 10
```



```

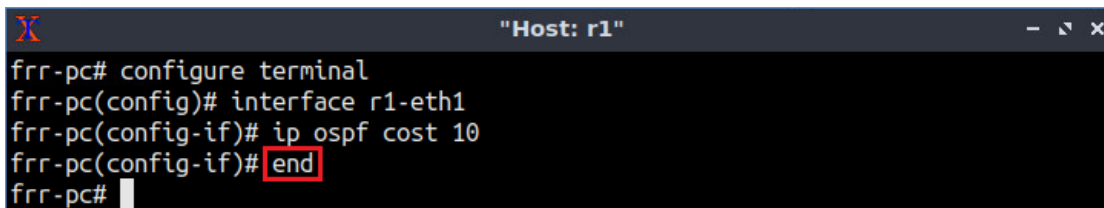
frr-pc# configure terminal
frr-pc(config)# interface r1-eth1
frr-pc(config-if)# ip ospf cost 10
frr-pc(config-if)#

```

Figure 52. Overriding interface *r1-eth0* calculated cost.

**Step 4.** Type the following command to exit from the interface configuration.

```
end
```



```

frr-pc# configure terminal
frr-pc(config)# interface r1-eth1
frr-pc(config-if)# ip ospf cost 10
frr-pc(config-if)# end
frr-pc#

```

Figure 53. Exiting configuration mode.

## 5.2 Verifying interfaces costs and bandwidth

**Step 1.** In router r1, issue the following command to see interface *r1-eth1* configuration.

```
show ip ospf interface r1-eth1
```

```

Host: r1
frr-pc# show ip ospf interface r1-eth1
r1-eth1 is up
  ifindex 3, MTU 1500 bytes, BW 200 Mbit <UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 192.168.12.1/30, Broadcast 192.168.12.3, Area 0.0.0.0
  MTU mismatch detection: enabled
  Router ID 192.168.14.1, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State DR, Priority 1
  Backup Designated Router (ID) 192.168.23.1, Interface Address 192.168.12.2
  Saved Network-LSA sequence number 0x80000003
  Multicast group memberships: OSPFAllRouters OSPFDesignatedRouters
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
    Hello due in 7.136s
  Neighbor Count is 1, Adjacent neighbor count is 1

frr-pc#

```

Figure 54. Verifying OSPF path cost.

Notice that although the bandwidth is 200 Mbps, the cost is 10, which diverges from the cost resulting from the formula presented in section 1.

**Step 2.** In router r1, issue the following to see interface *r1-eth2* configuration. Notice that the bandwidth is 1000 Mbps (i.e., 1 Gbps) and the cost 100.

```
show ip ospf interface r1-eth2
```

```

Host: r1
frr-pc# show ip ospf interface r1-eth2
r1-eth2 is up
  ifindex 4, MTU 1500 bytes, BW 1000 Mbit <UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 192.168.14.1/30, Broadcast 192.168.14.3, Area 0.0.0.0
  MTU mismatch detection: enabled
  Router ID 192.168.14.1, Network Type BROADCAST, Cost: 100
  Transmit Delay is 1 sec, State Backup, Priority 1
  Backup Designated Router (ID) 192.168.14.1, Interface Address 192.168.14.1
  Multicast group memberships: OSPFAllRouters OSPFDesignatedRouters
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
    Hello due in 9.716s
  Neighbor Count is 1, Adjacent neighbor count is 1

frr-pc#

```

Figure 55. Verifying OSPF path cost.

**Step 3.** Type the following command to see the routing table of router r1.

```
show ip route
```



```

Host: r1
frr-pc# 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

O 192.168.1.0/24 [110/10] is directly connected, r1-eth0, 01:03:37
C>* 192.168.1.0/24 is directly connected, r1-eth0, 01:07:28
O>* 192.168.2.0/24 [110/30] via 192.168.12.2, r1-eth1, 00:04:13
O 192.168.12.0/30 [110/10] is directly connected, r1-eth1, 00:04:13
C>* 192.168.12.0/30 is directly connected, r1-eth1, 01:07:28
O 192.168.14.0/30 [110/40] via 192.168.12.2, r1-eth1, 00:04:13
C>* 192.168.14.0/30 is directly connected, r1-eth2, 01:07:28
O>* 192.168.23.0/30 [110/20] via 192.168.12.2, r1-eth1, 00:04:13
O>* 192.168.34.0/30 [110/30] via 192.168.12.2, r1-eth1, 00:04:13
frr-pc#

```

Figure 56. Verifying the routing table of router r1.

Notice that the route selected to reach the network 192.168.2.0/24 is via the interface *r1-eth2*, which has the lower cost.

**Step 4.** In host h1 terminal issue the command below to start an iperf3 client. The iperf3 client will start transferring data to host h2 (i.e., the iperf3 server). The argument `-c` specifies that host h1 will be the client followed by host h2 IP address (i.e., 192.168.2.10).

```
iperf3 -c 192.168.2.10
```

```

Host: h1
root@frr-pc:~# iperf3 -c 192.168.2.10
Connecting to host 192.168.2.10, port 5201
[ 89] local 192.168.1.10 port 52386 connected to 192.168.2.10 port 5201
[ ID] Interval          Transfer    Bitrate    Retr  Cwnd
[ 89] 0.00-1.00        sec  23.9 MBytes  200 Mbits/sec    1   212 KBytes
[ 89] 1.00-2.00        sec  23.3 MBytes  195 Mbits/sec    0   284 KBytes
[ 89] 2.00-3.00        sec  22.6 MBytes  190 Mbits/sec    0   341 KBytes
[ 89] 3.00-4.00        sec  23.0 MBytes  193 Mbits/sec    0   389 KBytes
[ 89] 4.00-5.00        sec  23.1 MBytes  194 Mbits/sec    0   431 KBytes
[ 89] 5.00-6.00        sec  23.6 MBytes  198 Mbits/sec    0   471 KBytes
[ 89] 6.00-7.00        sec  22.6 MBytes  189 Mbits/sec    0   506 KBytes
[ 89] 7.00-8.00        sec  23.0 MBytes  193 Mbits/sec    0   540 KBytes
[ 89] 8.00-9.00        sec  23.4 MBytes  196 Mbits/sec    0   573 KBytes
[ 89] 9.00-10.00       sec  22.6 MBytes  190 Mbits/sec    0   660 KBytes
-----
[ ID] Interval          Transfer    Bitrate    Retr
[ 89] 0.00-10.00       sec  231 MBytes  194 Mbits/sec    1      sender
[ 89] 0.00-10.06       sec  229 MBytes  191 Mbits/sec           receiver

iperf Done.
root@frr-pc:~#

```

Figure 57. Starting an iperf3 client in host h1.

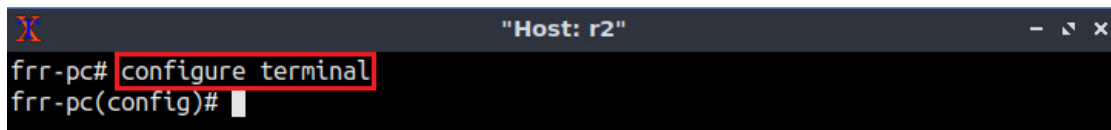
The result observed in the figure above shows that throughput is around 192 Mbps which is close the link bandwidth (i.e., 200 Mbps).

### 5.3 Changing route cost in an intermediate router

In this subsection, you will change the cost of the interface *r2-eth1*, aiming at impacting the aggregate path's cost and modifying the route to reach the network 192.168.2.0/24.

**Step 1.** Enter router r2 configuration mode issuing the following command.

```
configure terminal
```

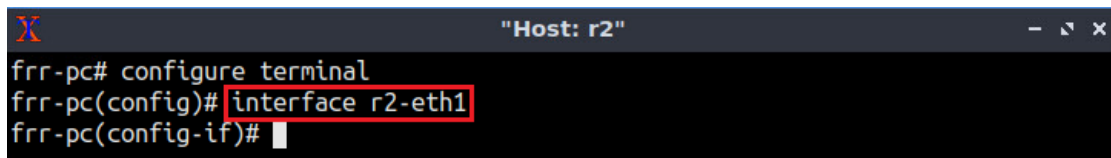


```
frr-pc# configure terminal
frr-pc(config)#
```

Figure 58. Entering configuration mode in router r2.

**Step 2.** Enter interface *r2-eth1* configuration by typing the command below.

```
interface r2-eth1
```

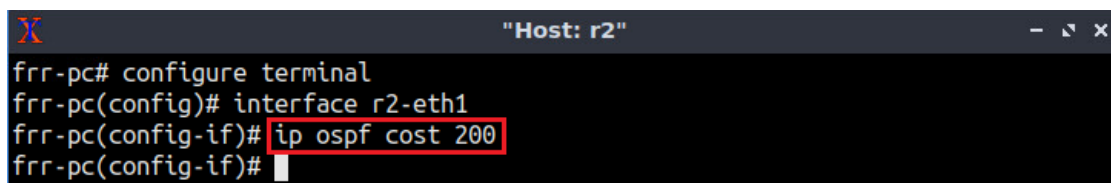


```
frr-pc# configure terminal
frr-pc(config)# interface r2-eth1
frr-pc(config-if)#
```

Figure 59. Entering interface *r2-eth1* configuration.

**Step 3.** Modify the OSPF cost in the interface *r2-eth1* issuing the following command.

```
ip ospf cost 200
```



```
frr-pc# configure terminal
frr-pc(config)# interface r2-eth1
frr-pc(config-if)# ip ospf cost 200
frr-pc(config-if)#
```

Figure 60. Modifying OSPF cost in the interface *r2-eth1*.

**Step 4.** Type the following command to exit configuration mode.

```
end
```

```

Host: r2
frr-pc# configure terminal
frr-pc(config)# interface r2-eth1
frr-pc(config-if)# ip ospf cost 200
frr-pc(config-if)# end
frr-pc#

```

Figure 61. Exiting from configuration mode.

**Step 5.** To verify the route selected to reach the network 192.168.2.0/24, type the following command in router r1 terminal.

```
show ip route
```

```

Host: r1
frr-pc# 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

O 192.168.1.0/24 [110/10] is directly connected, r1-eth0, 01:17:20
C>* 192.168.1.0/24 is directly connected, r1-eth0, 01:21:11
O>* 192.168.2.0/24 [110/120] via 192.168.14.2, r1-eth2, 00:05:53
O 192.168.12.0/30 [110/10] is directly connected, r1-eth1, 00:17:56
C>* 192.168.12.0/30 is directly connected, r1-eth1, 01:21:11
O 192.168.14.0/30 [110/100] is directly connected, r1-eth2, 00:05:53
C>* 192.168.14.0/30 is directly connected, r1-eth2, 01:21:11
O>* 192.168.23.0/30 [110/120] via 192.168.14.2, r1-eth2, 00:05:53
O>* 192.168.34.0/30 [110/110] via 192.168.14.2, r1-eth2, 00:05:53
frr-pc#

```

Figure 62. Showing the routing table of router r1.

Notice that the route selected to reach the network 192.168.2.0/24 is via the interface *r1-eth2*, which has the lower cost.

This concludes lab 8. Stop the emulation and then exit out of MiniEdit.

## References

1. Moy, J. "RFC 1247: OSPF version 2." 1991.
2. Moy, J. "RFC 2328: OSPF version 2." 1998.
3. Coltun, R., Ferguson, D., Moy, J., Lindem, A. "RFC 5340: OSPF for IPv6." 2008.
4. Peterson, L. "Computer networks: a systems approach." 6th Edition, 2021.
5. Tanenbaum, D. Wetherall. "Computer networks." 5th Edition. 2012.
6. J. Kurose, K. Ross. "Computer networking, a top-down approach." 7th Edition. 2017.
7. Teare, D., Vachon, B., Graziani, R. "Implementing Cisco IP routing (ROUTE) foundation learning guide: (CCNP ROUTE 300-101)." 2014.

8. Linux foundation collaborative projects. "*FRR routing documentation.*" 2017  
[Online] Available: <http://docs.frrouting.org/en/latest/>.



UNIVERSITY OF  
**SOUTH CAROLINA**

## OPEN SHORTEST PATH FIRST

### Lab 9: Configuring Multi-Area OSPFv3

Document Version: **10-03-2021**



Award 1829698

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

## Contents

Overview .....	3
Objectives.....	3
Lab settings .....	3
Lab roadmap .....	3
1 Introduction .....	3
1.1 Differences between OSPv2 and OPSFv3.....	3
1.2 OSPFv3 packets .....	4
1.3 OSPFv3 LSDB exchange and synchronization.....	4
1.4 OSPFv3 passive interface .....	6
2 Lab topology.....	6
2.1 Lab settings.....	6
2.2 Loading the topology.....	7
2.3 Load the zebra daemon and verify the configuration .....	10
3 Configuring multi-area OSPFv3.....	13
1.1 Configuring OSPFv3 in routers r1, r2, and r3 .....	13
3.2 Configuring passive interfaces .....	18
4 Verifying routing information and connectivity .....	19
References .....	21

## Overview

This lab introduces Open Shortest Path First version 3 (OSPFv3), which supports IPv6 routing. The lab explains the behavior and main features of OSPFv3 highlighting the differences with its previous version. Additionally, it guides the user on how to configure and verify multi-area OSPFv3.

## Objectives

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

1. Understand the features of OSPFv3.
2. Identify the difference between OSPFv2 and OSPFv3.
3. Configure multi-area OSPFv3.
4. Verify the configuration.

## Lab settings

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

Table 1. Credentials to access the Client 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 multi-area OSPFv3.
4. Section 4: Verifying routing information and connectivity.

### 1 Introduction

#### 1.1 Differences between OSPv2 and OPSFv3

The OSPFv3 protocol supports IPv6 routes and uses the same principles of OSPFv2. Understanding the differences between OSPFv2 and OSPFv3 is essential to operate and

configure IPv6 networks that uses OSPF as their routing protocol. The changes introduced in OSPFv3 were necessary to support new addressing semantics and the size of IPv6 addresses. The OSPFv3 header is simpler than the previous version having the authentication fields removed from the OSPFv3 header. Instead, OSPFv3 relies on IPv6 Authentication Header (AH) and Encapsulating Security Packets (ESP). Additionally, OSPFv3 supports IPv6 address families, which supports both IPv4 and IPv6 unicast traffic.

Notice that the OSPF version implemented in Free Range Routing (FRR), is based in the RFC 2740<sup>3</sup>, which only supports IPv6 routing.

## 1.2 OSPFv3 packets

The OSPFv3 packets are classified into the following types:

- Hello packets: They are used to discover, build, and keep OSPF neighbor adjacencies. In OSPFv3, hello packets do not contain address information. Instead, it includes the interface ID that the originating router has included to identify its interface to the link uniquely.
- Database Description Packets (DBD): It is used to describe the Link-State Data Base (LSDB) of a router. The LSDB is used by routers participating in the OSPF process to verify that their databases are synchronized.
- Link-State Request (LSR): A router uses it to request the latest Link State Advertisement (LSA).
- Link-State Update (LSU): They are used to flood LSAs and sending LSA responses to LSR packets.
- Link-State Acknowledgement (LSAck): It is used to acknowledge LSA packets.

OSPFv3 packets are encapsulated into IPv6 packets, and the protocol number is 89.

## 1.3 OSPFv3 LSDB exchange and synchronization

The routers participating in the OSPF process pass through multiple states before forming full adjacency, as shown in Figure 1. Router r1 and r2 will start exchanging OSPF packets following the states summarized below.

- Down: No information is sent or received by routers r1 and r2.
- Init: Router r2 interface has detected a Hello packet broadcasted by router r1. The IPv6 broadcast address used by OSPFv3 is FF02::5. The hello packet also contains the Router ID. However, bidirectional communication has not been established.
- Two-way: Router r2 responded by sending a hello packet to the unicast address of router r1 (i.e., 2001:10::1). At the end of this state, the Designated Router (DR) and the Backup Designated Router (BDR) are chosen. The DR and the BDR are used to reduce the amount of packet exchanged in the OSPF process, meaning that non-DR and non-BDR will only exchange routes with the former and the latter. The router with the highest router ID will be elected as the selected as the DR (i.e.,



router r2). The router with the second router ID will become the BDR (i.e., router r1).

- **ExStart:** In this state, routers r1 and r2 establish the initial sequence number used to ensure that routers get the latest update when they exchange OSPF packets.
- **Exchange:** During this state, routers exchange LSDB by sending DBD packets. DBD packet contain information about the LSA entry header included in the router's LSDB. In OSPFv3, this information is about the link. Additionally, the router uses a sequence number to determine the freshness of the OSPF packet. Each DBD packet is acknowledged with LSAck packet.
- **Loading:** In this state, router information exchange finalizes. Routers build two lists that comprise the link-state request and link-state retransmission. The link-state request list includes any information that might be incomplete or outdated, whereas any update that is sent will be registered in the retransmission list until it gets acknowledged.
- **Full:** At this point, neighbor adjacencies are fully established, meaning that adjacent routers will have the same LSDB.

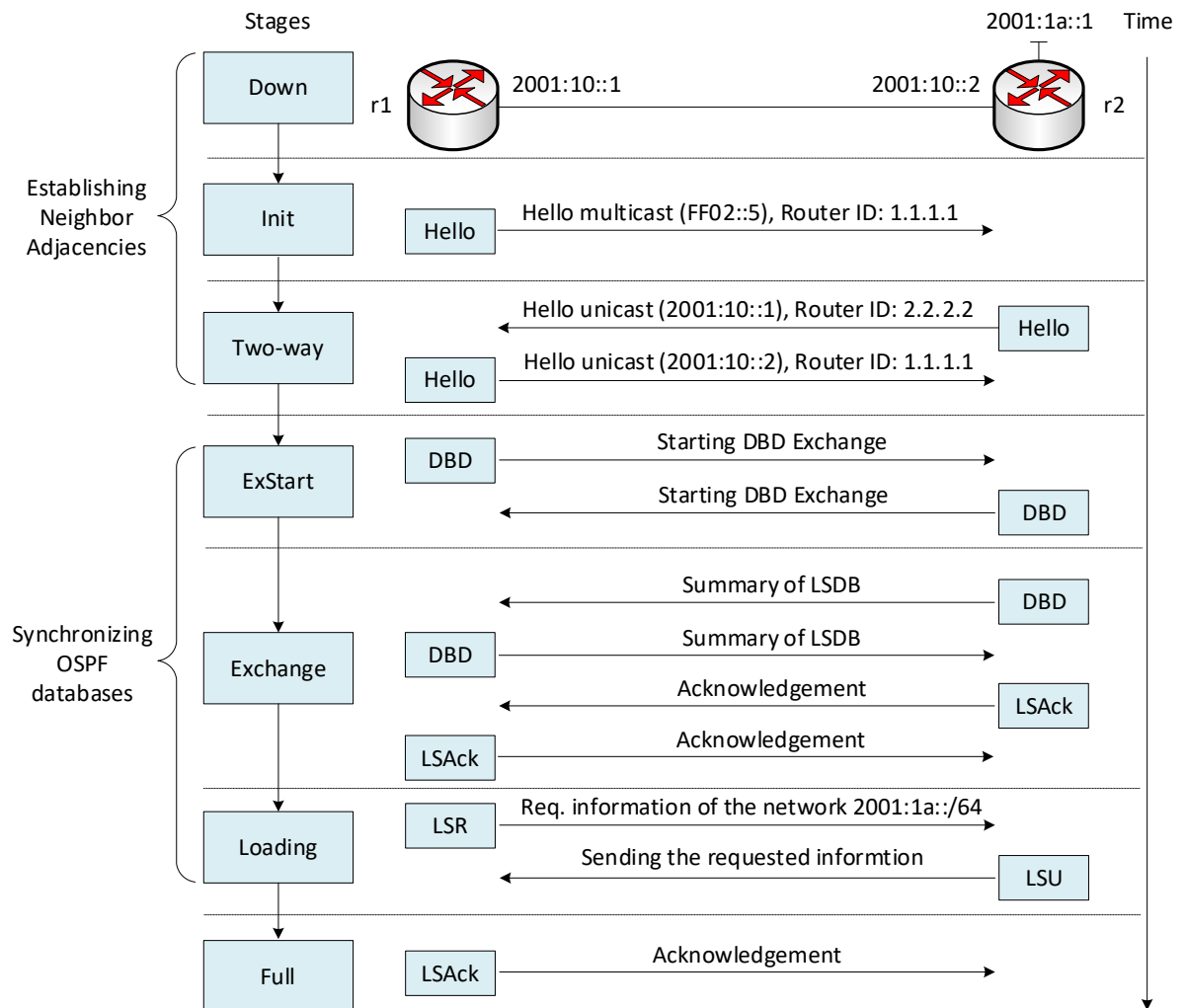


Figure 1. OSPFv3 states and packet exchange timeline.

## 1.4 OSPFv3 passive interface

Configuring a passive interface in a router participating in the OSPFv3 process is a common method for reducing the use of resources. When a passive interface is configured, the router stops sending and receiving OSPF hello packets on the selected interface. Therefore, the passive interface must be configured only on interfaces where the router will not establish any OSPF neighbor adjacency.

## 2 Lab topology

Consider Figure 4. The topology consists of four networks connected to its respective routers. The communication between networks is established via multi-area OSPFv3 configured in all routers.

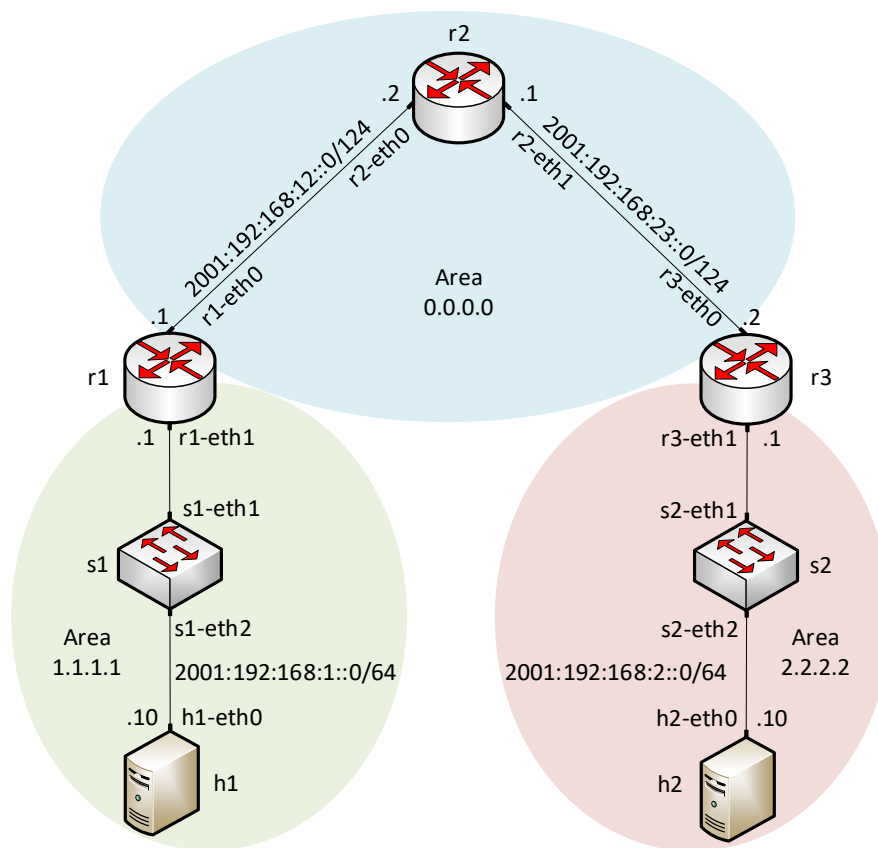


Figure 2. Lab topology.

### 2.1 Lab settings

Table 2 contains information about the devices and their corresponding interfaces, IPv4 addresses, subnet, and default gateway.

Table 2. Topology information.

Device	Interface	IPv6 Address	Subnet	Default Gateway
Router 1 (r1)	r1-eth0	2001:192:168:12::1	/124	N/A
	r1-eth1	2001:192:168:1::1	/64	N/A
Router 2 (r2)	r2-eth0	2001:192:168:12::2	/124	N/A
	r2-eth1	2001:192:168:23::1	/124	N/A
Router 3 (r3)	r3-eth0	2001:192:168:23::2	/124	N/A
	r3-eth1	2001:192:168:2::1	/64	N/A
Host 1 (h1)	h1-eth0	2001:192:168:1::10	/64	2001:192:168:1::1
Host 2 (h2)	h2-eth0	2001:192:168:2::10	/64	2001:192:168:2::1

## 2.2 Loading the topology

**Step 1.** Launch MiniEdit by double clicking on the shortcut located on the desktop. When prompted for a password, type `password`.



Figure 3. MiniEdit shortcut.

**Step 2.** In MiniEdit's menu bar, click on *File* then *Open* to load the lab topology. Locate the *lab9.mn* topology file in the default directory, `/home/frr/OSPF_Labs/lab9` and click on *Open*.

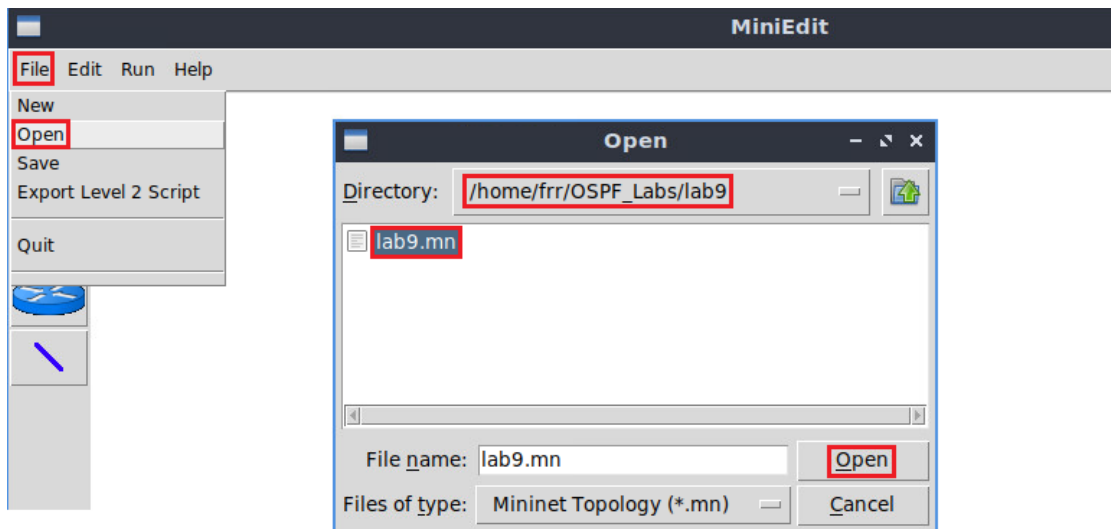


Figure 4. MiniEdit's open dialog.

**Step 3.** Click on the *Run* button located on the lower left-hand side to run the emulation.

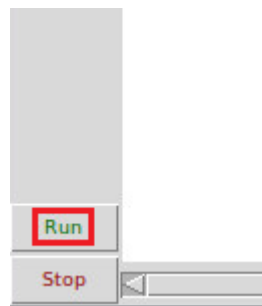


Figure 5. Starting the emulation.

At this point, the topology is loaded with all the required network components. Next, you will execute a script that will configure the interfaces of the routers.

Automating the configuration of interfaces with the script is intended to facilitate the lab experience and to focus on the OSPF experiment.

**Step 4.** Open the Linux terminal by clicking on the icon located in the taskbar.

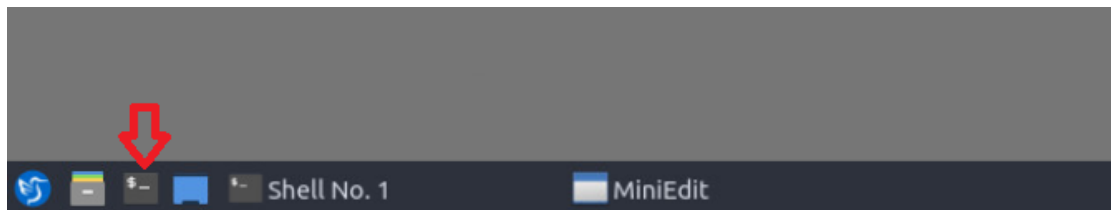


Figure 6. Opening Linux terminal.

**Step 5.** Issue the command shown below to set the lab environment. This command configures the IP addresses of the routers' interface according to Table 2. If a password is required, type `password`.

```
sudo set_env lab9
```

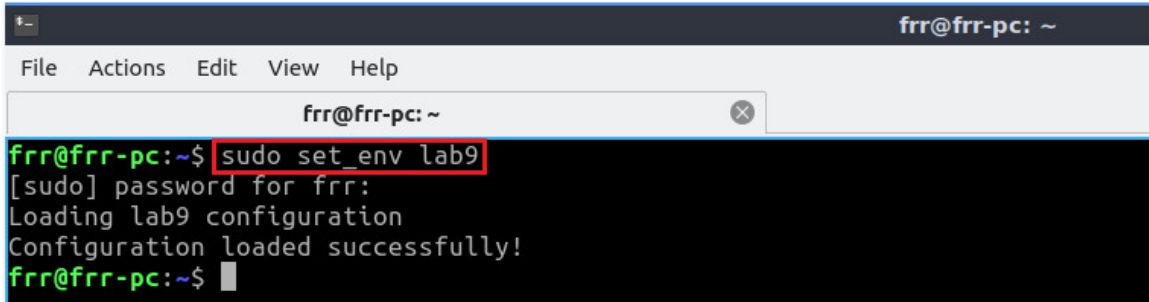


Figure 7. Setting the lab environment.

**Step 6.** Type the following command to exit the Linux terminal.

```
exit
```

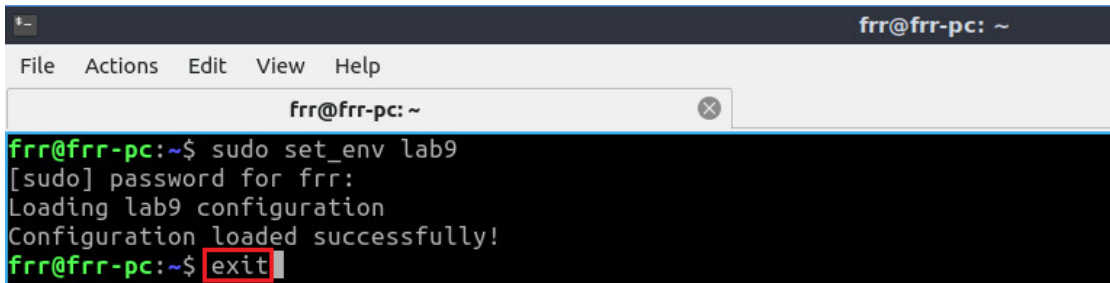


Figure 8. Exiting the Linux terminal.

**Step 7.** Click on Mininet's terminal, i.e., the one launched when MiniEdit was started.

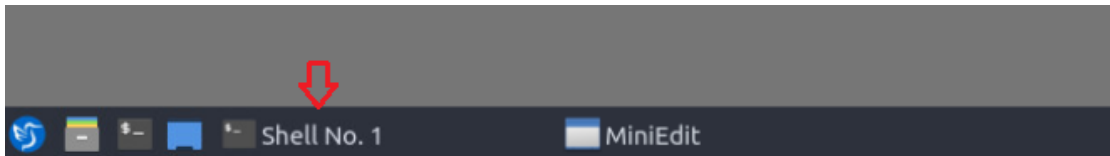
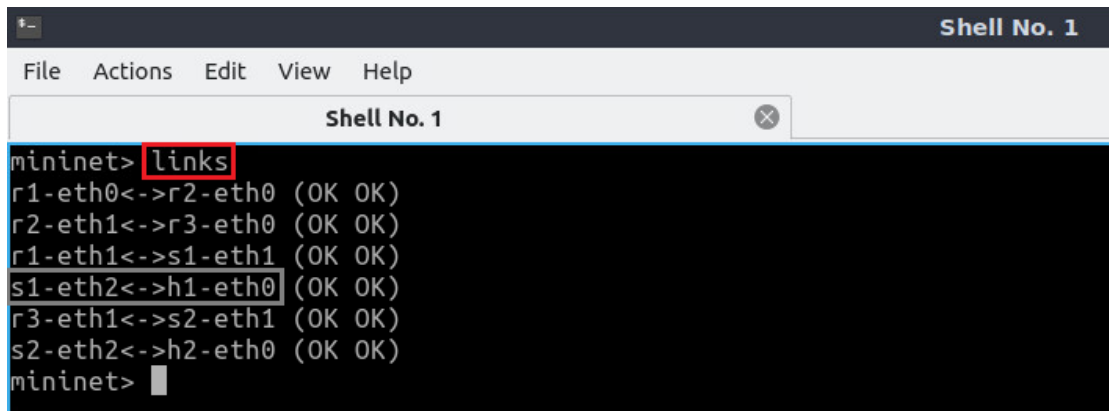


Figure 9. Opening Mininet's terminal.

**Step 8.** Issue the following command to display the interface names and connections.

```
links
```



```

mininet> links
r1-eth0<->r2-eth0 (OK OK)
r2-eth1<->r3-eth0 (OK OK)
r1-eth1<->s1-eth1 (OK OK)
s1-eth2<->h1-eth0 (OK OK)
r3-eth1<->s2-eth1 (OK OK)
s2-eth2<->h2-eth0 (OK OK)
mininet>

```

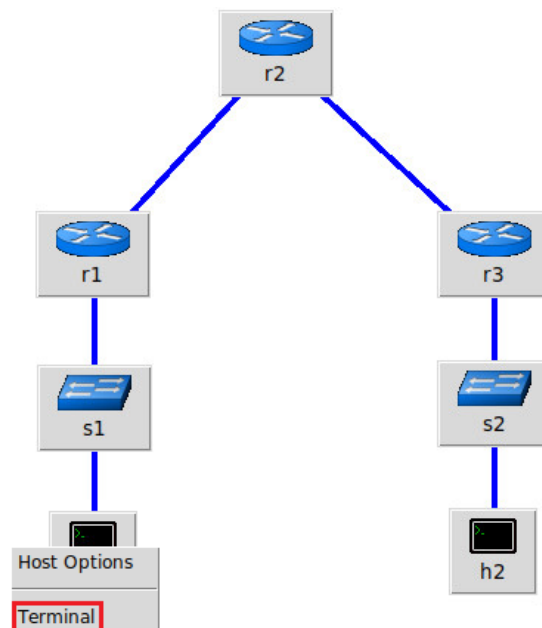
Figure 10. Displaying network interfaces.

In the figure above, the link displayed within the gray box indicates that interface *eth2* of switch *s1* connects to interface *eth0* of host *h1* (i.e., *s1-eth2*<->*h1-eth0*).

### 2.3 Load the zebra daemon and verify the configuration

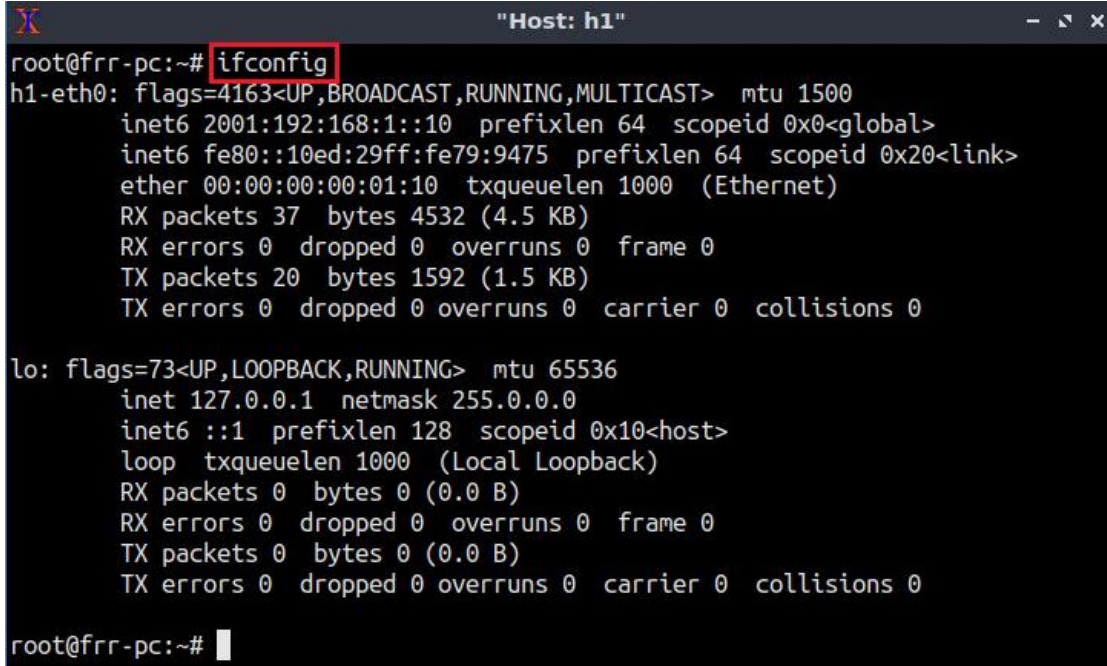
In this section, you will verify that the IP addresses listed in Table 2 and inspect the routing table of the routers.

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

Figure 11. Opening host *h1* terminal.

**Step 2.** In the host *h1* terminal, type the command shown below to verify that the IP address was assigned successfully. You will verify that host *h1* has two interfaces. Interface *h1-eth0* is configured with the IPv6 address 2001:192:168:1::10 and the link-local address is fe80:10ed:29ff:fe79:9475. The prefix length for IPv6 addresses is 64.

```
ifconfig
```



```

root@frr-pc:~# ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 2001:192:168:1::10 prefixlen 64 scopeid 0x0<global>
    inet6 fe80::10ed:29ff:fe79:9475 prefixlen 64 scopeid 0x20<link>
    ether 00:00:00:00:01:10 txqueuelen 1000 (Ethernet)
    RX packets 37 bytes 4532 (4.5 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 20 bytes 1592 (1.5 KB)
    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@frr-pc:~#

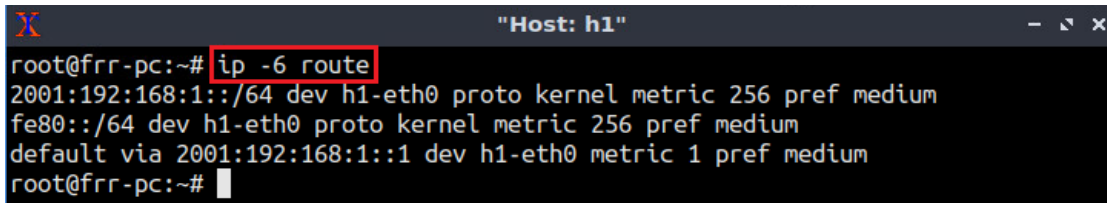
```

Figure 12. Displaying host h1 interfaces and IP addresses.

A link-local address is an IPv6 unicast address that can be configured on any interface. It is used to reach the neighboring nodes attached to the same link. The IPv6 link-local addresses start with the FE80 prefix.

**Step 3.** In the host h1 terminal, type the command shown below to verify that the default gateway IPv6 address is 2001:192:168:1::1.

```
ip -6 route
```



```

root@frr-pc:~# ip -6 route
2001:192:168:1::/64 dev h1-eth0 proto kernel metric 256 pref medium
fe80::/64 dev h1-eth0 proto kernel metric 256 pref medium
default via 2001:192:168:1::1 dev h1-eth0 metric 1 pref medium
root@frr-pc:~#

```

Figure 13. Displaying host h1 default route.

**Step 4.** To verify the hosts h2 and h3 default route, proceed by following from step 1 to step 3. Similar results should be observed.

**Step 5.** You will validate that the router interfaces are configured according to Table 2. To proceed, hold right-click on router r1 and select *Terminal*.

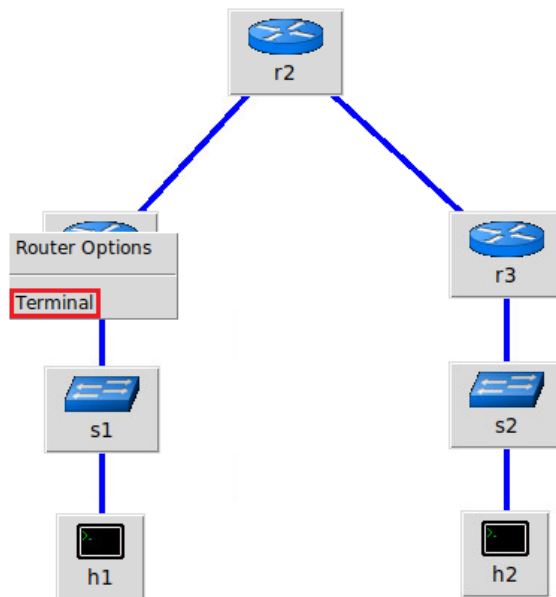


Figure 14. Opening a terminal on router r1.

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

```
zebra
```

Figure 15. Starting zebra daemon.

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

```
vtysh
```

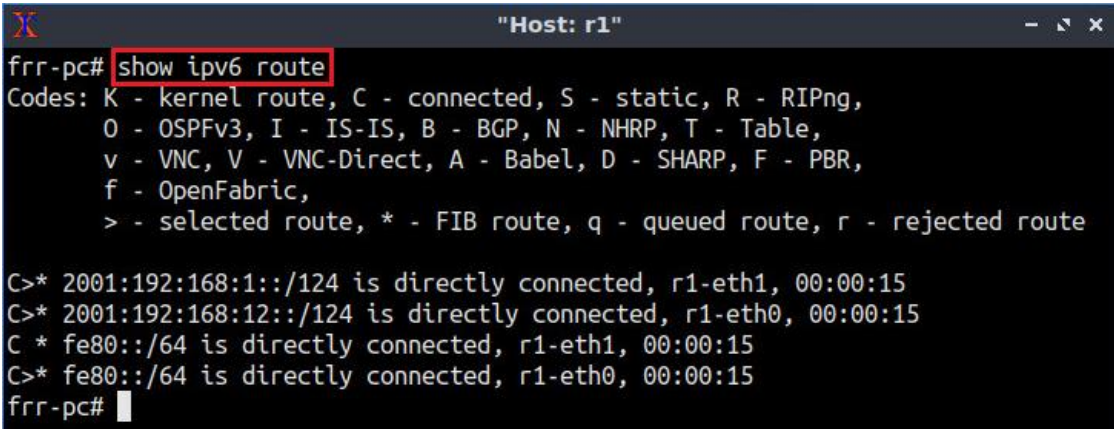
Figure 16. Starting vtysh on router r1.

**Step 8.** Type the following command on router r1 terminal to verify the routing table. It will list all the directly connected networks. Notice that the routing table of router r1 does not contain any route to the network connected to router r3 (2001:192:168:2::0/64) as there is no routing protocol configured yet. The first two entries correspond to the directly



connected networks. The last two entries correspond to link-local network connected to each interface.

```
show ipv6 route
```



```
frr-pc# show ipv6 route
Codes: K - kernel route, C - connected, S - static, R - RIPng,
       O - OSPFv3, I - IS-IS, B - BGP, 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>* 2001:192:168:1::/124 is directly connected, r1-eth1, 00:00:15
C>* 2001:192:168:12::/124 is directly connected, r1-eth0, 00:00:15
C * fe80::/64 is directly connected, r1-eth1, 00:00:15
C>* fe80::/64 is directly connected, r1-eth0, 00:00:15
frr-pc#
```

Figure 17. Displaying routing table of router r1.

**Step 9.** To verify the routing table of the other routers, proceed similarly by following from step 5 to step 8 on the corresponding router terminal.

### 3 Configuring multi-area OSPFv3

In this section, you will configure r1, r2, and r3 to participate in the OSPFv3 process. The topology in this lab is divided into three areas as shown in Figure 4. By dissecting a network into several areas, routers will not have a detailed information about the network topology outside of their area. This segregation improves scalability and control over the network.

#### 1.1 Configuring OSPFv3 in routers r1, r2, and r3

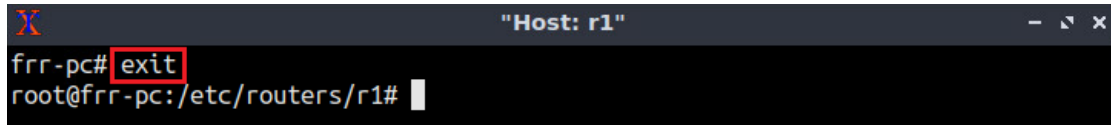
**Step 1.** Open router r1's terminal by selecting the icon from the taskbar as shown below.



Figure 18. Opening router r1's terminal.

**Step 2.** In router r1's terminal, type the command below to exit the vtysh session.

```
exit
```



```

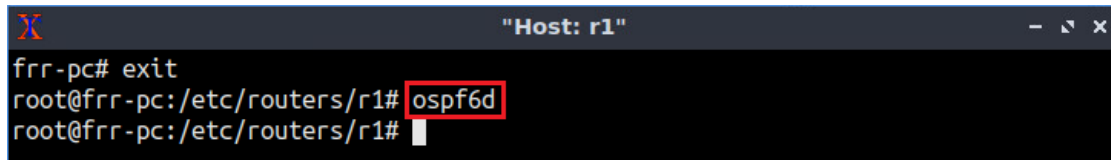
frr-pc# exit
root@frr-pc:/etc/routers/r1#

```

Figure 19. Exiting the vtysh session.

**Step 3.** Enable and start the OSPF routing protocol by typing the following command:

```
ospf6d
```



```

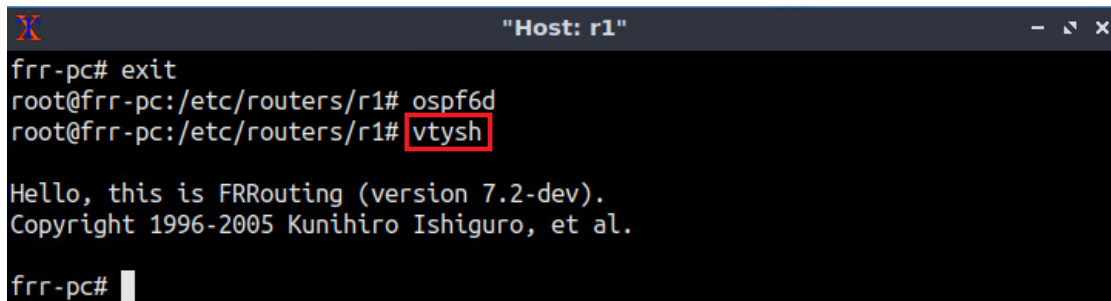
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospf6d
root@frr-pc:/etc/routers/r1#

```

Figure 20. Starting the OSPF daemon.

**Step 4.** To enter router r1's terminal, type the following command:

```
vttysh
```



```

frr-pc# exit
root@frr-pc:/etc/routers/r1# ospf6d
root@frr-pc:/etc/routers/r1# vtysh

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

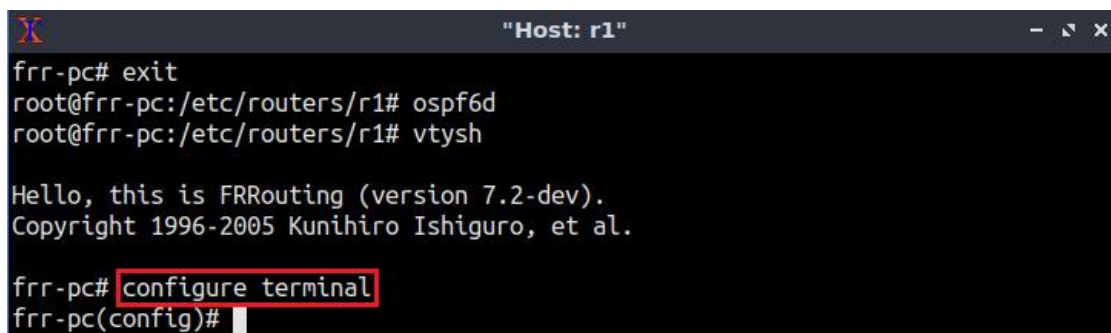
frr-pc#

```

Figure 21. Starting vtysh in router r1.

**Step 5.** At this point, you have entered router r1's terminal, to enter configuration mode in router r1, issue the following command:

```
configure terminal
```



```

frr-pc# exit
root@frr-pc:/etc/routers/r1# ospf6d
root@frr-pc:/etc/routers/r1# vtysh

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

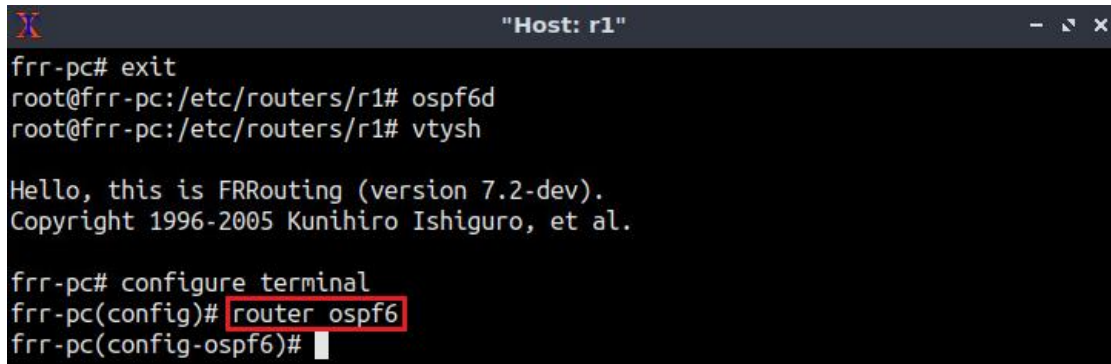
frr-pc# configure terminal
frr-pc(config)#

```

Figure 22. Entering configuration mode in router r1.

**Step 6.** To configure OSPF, type the command shown below. This command will enable OSPF configuration mode.

```
router ospf6
```

A terminal window titled "Host: r1" showing the configuration process. The user starts at the root prompt, exits the shell, and enters 'ospf6' to enter OSPF configuration mode. The prompt changes to 'frr-pc(config-ospf6)#'. The command 'router ospf6' is entered and highlighted with a red box. The prompt then changes to 'frr-pc(config-ospf6)#'.

```
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospf6
root@frr-pc:/etc/routers/r1# vtysh

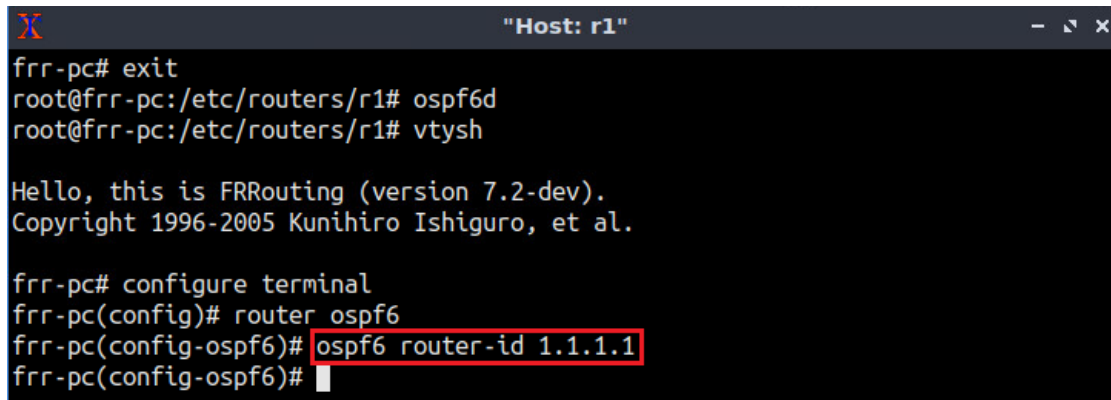
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# router ospf6
frr-pc(config-ospf6)#
```

Figure 23. Configuring OSPF in router r1.

**Step 7.** Type the following command to set an ID to router r1.

```
ospf6 router-id 1.1.1.1
```

A terminal window titled "Host: r1" showing the configuration process. The user enters 'ospf6 router-id 1.1.1.1' in the 'frr-pc(config-ospf6)#' prompt, which is highlighted with a red box. The prompt remains 'frr-pc(config-ospf6)#'.

```
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospf6
root@frr-pc:/etc/routers/r1# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf6
frr-pc(config-ospf6)# ospf6 router-id 1.1.1.1
frr-pc(config-ospf6)#
```

Figure 24. Setting an ID to router r1.

**Step 8.** Type the following command to enable interface *r1-eth0* to participate in the OSPF process corresponding to the area 0.0.0.0.

```
interface r1-eth0 area 0.0.0.0
```

```

Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospf6d
root@frr-pc:/etc/routers/r1# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf6
frr-pc(config-ospf6)# ospf6 router-id 1.1.1.1
frr-pc(config-ospf6)# interface r1-eth0 area 0.0.0.0
frr-pc(config-ospf6)#

```

Figure 25. Enabling interface *r1-eth0* to participate in the OSPF process.

**Step 9.** Type the following command to enable interface *r1-eth1* to participate in the OSPF process corresponding to the area 1.1.1.1.

```
interface r1-eth1 area 1.1.1.1
```

```

Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospf6d
root@frr-pc:/etc/routers/r1# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf6
frr-pc(config-ospf6)# ospf6 router-id 1.1.1.1
frr-pc(config-ospf6)# interface r1-eth0 area 0.0.0.0
frr-pc(config-ospf6)# interface r1-eth1 area 1.1.1.1
frr-pc(config-ospf6)#

```

Figure 26. Enabling interface *r1-eth1* to participate in the OSPF process.

**Step 10.** Type the following command to exit router configuration mode.

```
end
```

```

Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospf6d
root@frr-pc:/etc/routers/r1# vtysh

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

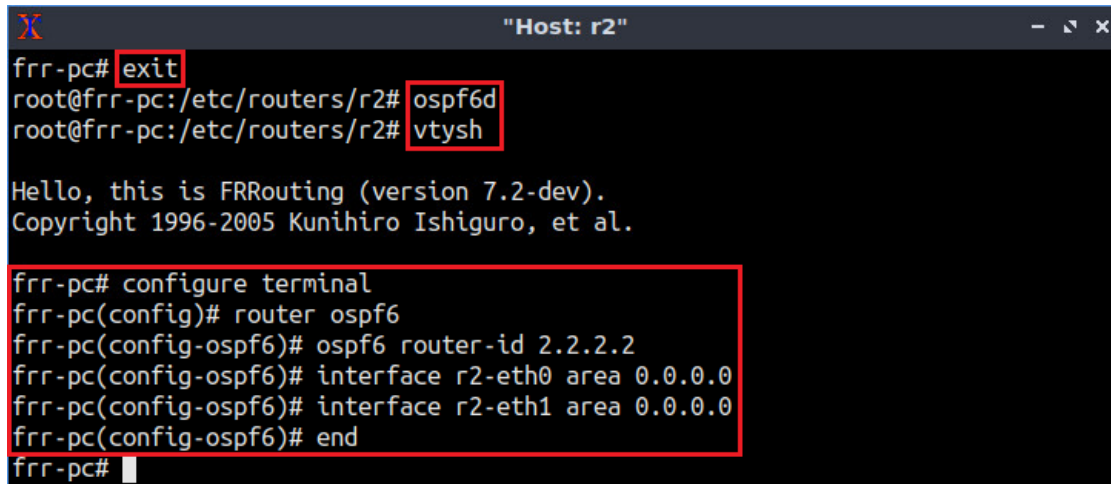
frr-pc# configure terminal
frr-pc(config)# router ospf6
frr-pc(config-ospf6)# ospf6 router-id 1.1.1.1
frr-pc(config-ospf6)# interface r1-eth0 area 0.0.0.0
frr-pc(config-ospf6)# interface r1-eth1 area 1.1.1.1
frr-pc(config-ospf6)# end
frr-pc#

```

Figure 27. Exiting router configuration mode.

Consider the above figure. After configuring OSPF in router r1, the interfaces *r1-eth0*, and *r1-eth1* will participate in the routing process of area 0.0.0.0 and area 1.1.1.1 respectively.

**Step 11.** 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 terminal, issue the commands summarized below.



```

frr-pc# exit
root@frr-pc:/etc/routers/r2# ospf6d
root@frr-pc:/etc/routers/r2# vtysh

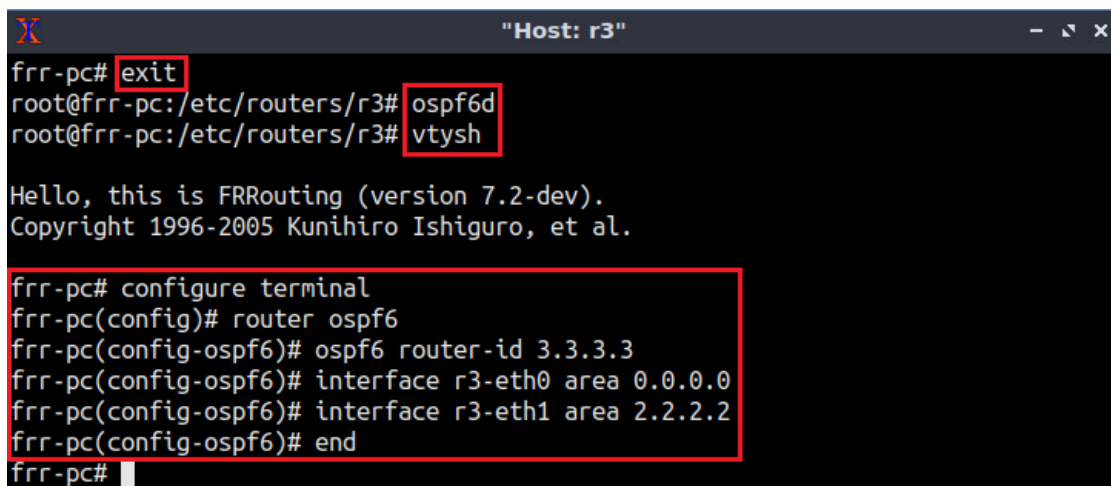
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# router ospf6
frr-pc(config-ospf6)# ospf6 router-id 2.2.2.2
frr-pc(config-ospf6)# interface r2-eth0 area 0.0.0.0
frr-pc(config-ospf6)# interface r2-eth1 area 0.0.0.0
frr-pc(config-ospf6)# end
frr-pc#
  
```

Figure 28. Router r2 configuration summary.

Consider the above figure. After configuring OSPF in router r2, the interfaces *r2-eth0*, and *r2-eth1* will participate in the routing process of area 0.0.0.0.

**Step 12.** Router r3 is configured similarly to router r2 but with different IP addresses (see Table 2). Those steps are summarized in the following figure. To proceed, in router r3 terminal, issue the commands depicted below.



```

frr-pc# exit
root@frr-pc:/etc/routers/r3# ospf6d
root@frr-pc:/etc/routers/r3# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf6
frr-pc(config-ospf6)# ospf6 router-id 3.3.3.3
frr-pc(config-ospf6)# interface r3-eth0 area 0.0.0.0
frr-pc(config-ospf6)# interface r3-eth1 area 2.2.2.2
frr-pc(config-ospf6)# end
frr-pc#
  
```

Figure 29. Router r3 configuration summary.

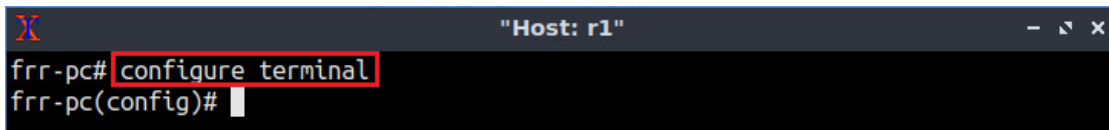
Consider the above figure. After configuring OSPF in router r3, the interfaces *r3-eth0* and *r3-eth1* will participate in the routing process of area 0.0.0.0 and area 2.2.2.2 respectively.

### 3.2 Configuring passive interfaces

Configuring a passive interface in a router participating in the OSPFv3 process is a common method for reducing the use of resources. When a passive interface is configured, the router stops sending and receiving OSPF hello packets on the selected interface. The following steps show how to establish the interfaces *r1-eth1* and *r3-eth1* as passive.

**Step 1.** Issue the following command in router *r1* to enter configuration mode.

```
configure terminal
```

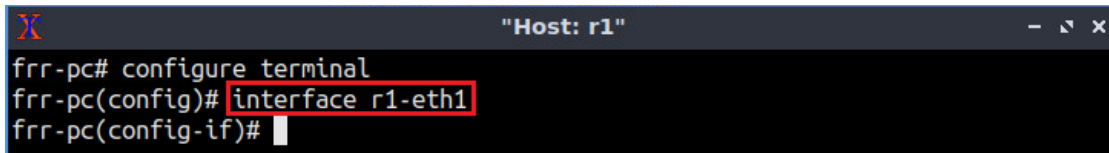
A terminal window titled "Host: r1" showing the command prompt "frr-pc#" followed by the command "configure terminal" which is highlighted with a red box. The prompt then changes to "frr-pc(config)#".

```
frr-pc# configure terminal
frr-pc(config)#
```

Figure 30. Entering configuration mode.

**Step 2.** Enter interface *r1-eth1* by typing the following command.

```
interface r1-eth1
```

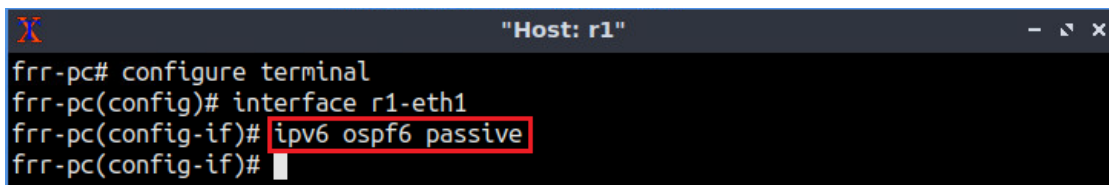
A terminal window titled "Host: r1" showing the command prompt "frr-pc#" followed by "configure terminal", then "frr-pc(config)#" followed by the command "interface r1-eth1" which is highlighted with a red box. The prompt then changes to "frr-pc(config-if)#".

```
frr-pc# configure terminal
frr-pc(config)# interface r1-eth1
frr-pc(config-if)#
```

Figure 31. Entering interface *r1-eth1* configuration.

**Step 3.** Type the following command to set the interface *r1-eth1* as passive.

```
ipv6 ospf6 passive
```

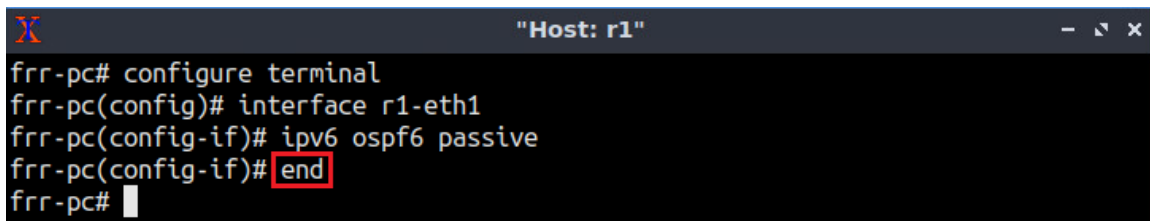
A terminal window titled "Host: r1" showing the command prompt "frr-pc#" followed by "configure terminal", then "frr-pc(config)#" followed by "interface r1-eth1", and finally "frr-pc(config-if)#" followed by the command "ipv6 ospf6 passive" which is highlighted with a red box. The prompt remains "frr-pc(config-if)#".

```
frr-pc# configure terminal
frr-pc(config)# interface r1-eth1
frr-pc(config-if)# ipv6 ospf6 passive
frr-pc(config-if)#
```

Figure 32. Configuring a passive interface.

**Step 4.** Issue the following command to exit configuration mode.

```
end
```



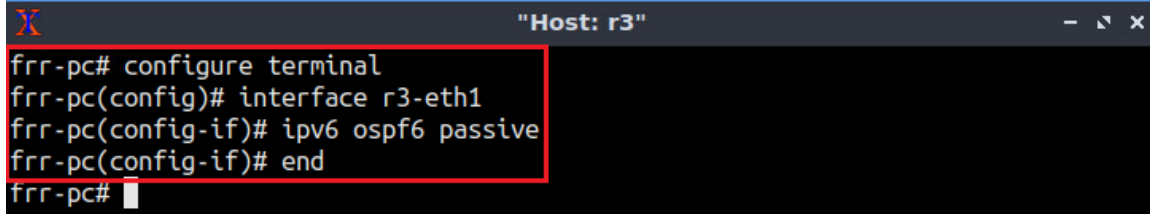
```

frr-pc# configure terminal
frr-pc(config)# interface r1-eth1
frr-pc(config-if)# ipv6 ospf6 passive
frr-pc(config-if)# end
frr-pc#

```

Figure 33. Exiting configuration mode.

**Step 5.** Similarly, configure router r3 by following from step 1 to step 3. The commands are summarized in the following figure.



```

frr-pc# configure terminal
frr-pc(config)# interface r3-eth1
frr-pc(config-if)# ipv6 ospf6 passive
frr-pc(config-if)# end
frr-pc#

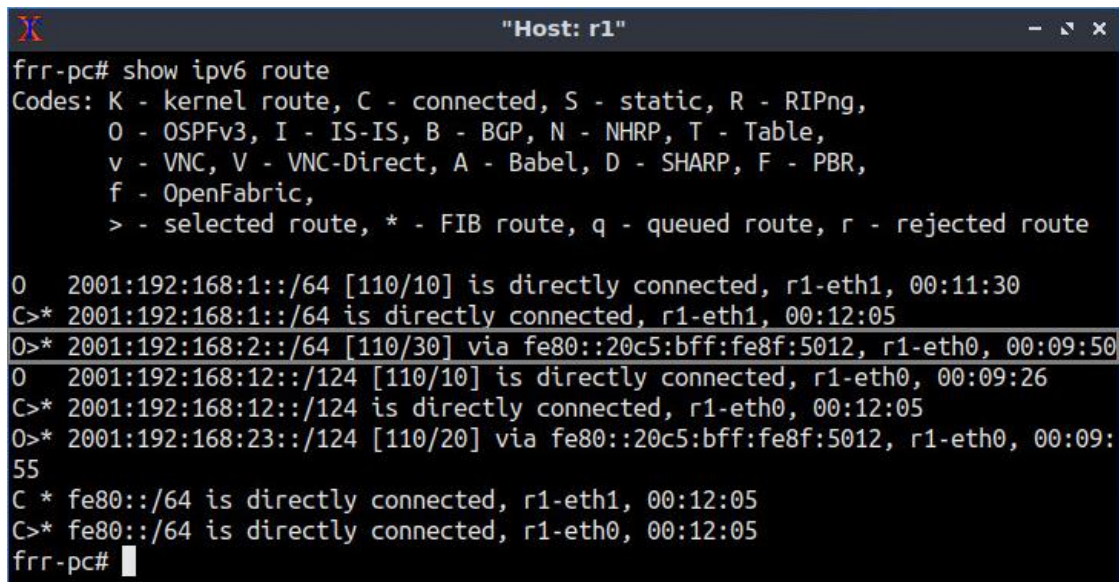
```

Figure 34. Configuring a passive interface in router r3.

## 4 Verifying routing information and connectivity

**Step 1.** Issue the following command to display the routing table of router r1.

```
show ipv6 route
```



```


frr-pc# show ipv6 route
Codes: K - kernel route, C - connected, S - static, R - RIPng,
       O - OSPFv3, I - IS-IS, B - BGP, 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

O  2001:192:168:1::/64 [110/10] is directly connected, r1-eth1, 00:11:30
C>* 2001:192:168:1::/64 is directly connected, r1-eth1, 00:12:05
O>* 2001:192:168:2::/64 [110/30] via fe80::20c5:bff:fe8f:5012, r1-eth0, 00:09:50
O  2001:192:168:12::/124 [110/10] is directly connected, r1-eth0, 00:09:26
C>* 2001:192:168:12::/124 is directly connected, r1-eth0, 00:12:05
O>* 2001:192:168:23::/124 [110/20] via fe80::20c5:bff:fe8f:5012, r1-eth0, 00:09:55
C * fe80::/64 is directly connected, r1-eth1, 00:12:05
C>* fe80::/64 is directly connected, r1-eth0, 00:12:05
frr-pc#

```

Figure 35. Displaying router r1's routing table.

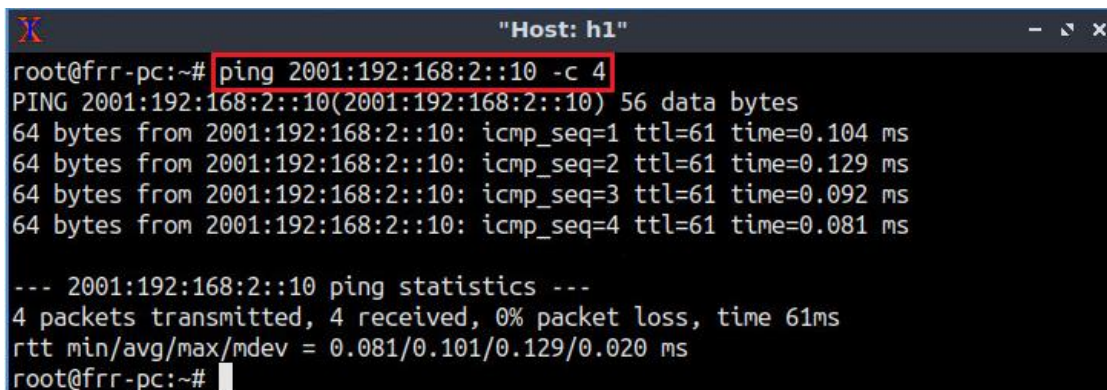
Consider the above figure. The output displays the networks learned along with additional attributes, such as the routing protocol that discovered the network. For example, networks with the symbol **O** on the left are learned from OSPF, whereas those with symbol **C** are directly connected to the device. Furthermore, the symbol **>** on the left of the displayed network indicates that this entry is selected to reach the network. As for the

symbol , this indicates that the route is inserted in Forwarding Information Base (FIB) of the router.

Notice that reaching the network 2001:192:168:2::/64 is performed via fe80:24a5:19ff:fe90:59d0, which is the link-local IPv6 address of the interface *r2-eth0* of router r2.

**Step 2.** In the host h1 terminal, issue the following command to test the connectivity between host h1 and host h2. You will observe a successful connectivity test.

```
ping 2001:192:168:2::10 -c 4
```



```

Host: h1
root@frr-pc:~# ping 2001:192:168:2::10 -c 4
PING 2001:192:168:2::10(2001:192:168:2::10) 56 data bytes
64 bytes from 2001:192:168:2::10: icmp_seq=1 ttl=61 time=0.104 ms
64 bytes from 2001:192:168:2::10: icmp_seq=2 ttl=61 time=0.129 ms
64 bytes from 2001:192:168:2::10: icmp_seq=3 ttl=61 time=0.092 ms
64 bytes from 2001:192:168:2::10: icmp_seq=4 ttl=61 time=0.081 ms

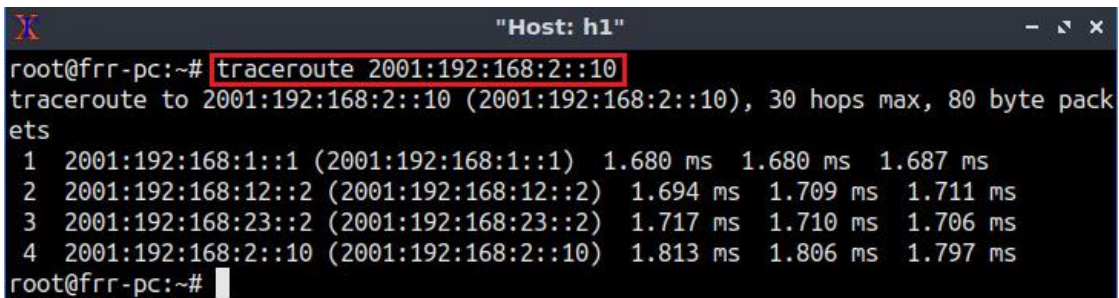
--- 2001:192:168:2::10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 61ms
rtt min/avg/max/mdev = 0.081/0.101/0.129/0.020 ms
root@frr-pc:~#

```

Figure 36. Testing connectivity from host h1 to host h2.

**Step 3.** In the host h1 terminal, issue the following command to track the route that packets take on the way to their destination, in this case host h2 (2001:192:168:2::10).

```
traceroute 2001:192:168:2::10
```



```

Host: h1
root@frr-pc:~# traceroute 2001:192:168:2::10
traceroute to 2001:192:168:2::10 (2001:192:168:2::10), 30 hops max, 80 byte packets
 1  2001:192:168:1::1 (2001:192:168:1::1)  1.680 ms  1.680 ms  1.687 ms
 2  2001:192:168:12::2 (2001:192:168:12::2)  1.694 ms  1.709 ms  1.711 ms
 3  2001:192:168:23::2 (2001:192:168:23::2)  1.717 ms  1.710 ms  1.706 ms
 4  2001:192:168:2::10 (2001:192:168:2::10)  1.813 ms  1.806 ms  1.797 ms
root@frr-pc:~#

```

Figure 37. Performing a traceroute test.

Notice that it takes four hops to reach host h2 from host h1. The type of the IPv6 addresses is global unicast, which perform the same function of public IPv4 addresses and are also routable to the Internet.

**Step 4.** In router r2 terminal, issue the following command to see the OSPF neighbors.

```
show ipv6 ospf6 neighbor
```



```

frr-pc# show ipv6 ospf6 neighbor
Neighbor ID      Pri   DeadTime   State/IfState   Duration I/F[State]
1.1.1.1          1     00:00:30   Full/BDR        15:51:44 r2-eth0[DR]
3.3.3.3          1     00:00:30   Full/DR         15:51:13 r2-eth1[BDR]
frr-pc#

```

Figure 38. Testing connectivity from host h1 to host h2.

This concludes lab 9. Stop the emulation and then exit out of MiniEdit.

## References

1. Moy, J. "RFC 1247: OSPF version 2." 1991.
2. Moy, J. "RFC 2328: OSPF version 2." 1998.
3. Moy, J., Coltun, R., Ferguson, D. "RFC 2740: OSPF for IPv6." 1999.
4. Coltun, R., Ferguson, D., Moy, J., Lindem, A. "RFC 5340: OSPF for IPv6." 2008.
5. Peterson, L. "Computer networks: a systems approach." 6th Edition, 2021.
6. Tanenbaum, D. Wetherall. "Computer networks." 5th Edition. 2012.
7. J. Kurose, K. Ross. "Computer networking, a top-down approach." 7th Edition. 2017.
8. Teare, D., Vachon, B., Graziani, R. "Implementing Cisco IP routing (ROUTE) foundation learning guide: (CCNP ROUTE 300-101)." 2014.
9. Linux foundation collaborative projects. "FRR routing documentation." 2017 [Online] Available: <http://docs.frrouting.org/en/latest/>.



UNIVERSITY OF  
**SOUTH CAROLINA**

## OPEN SHORTEST PATH FIRST

### Exercise 3: Configuring Multi-Area OSPFv3

Document Version: **10-24-2021**



Award 1829698

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

## Contents

1	Exercise description .....	3
1.1	Credentials .....	3
2	Setting the environment.....	3
3	Deliverables.....	6

## 1 Exercise description

This exercise is a practice to the Open Shortest Path First version 3 (OSPFv3). The exercise requires configuring multi-area OSPFv3 to provide connectivity between two hosts located in different areas. The topology in the figure below provides the information such as the devices' names, IPv6 addresses, interface names and the OSPF areas where each device is located.

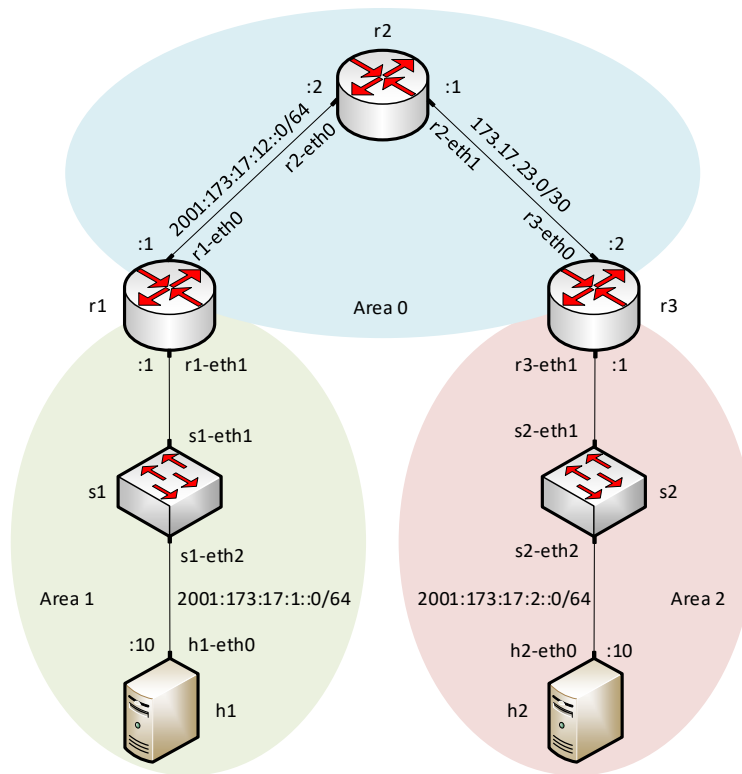


Figure 1. Exercise topology.

### 1.1 Credentials

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

Table 1. Credentials to access the Client machine.

Device	Account	Password
Client	admin	password

## 2 Setting the environment

Follow the steps below to set the environment.

**Step 1.** Open MiniEdit by double-clicking the shortcut on the desktop. If a password is required type `password`.



Figure 2. MiniEdit shortcut.

**Step 2.** Load the topology located at `/home/frr/OSPF_Labs/Exercise3/`.

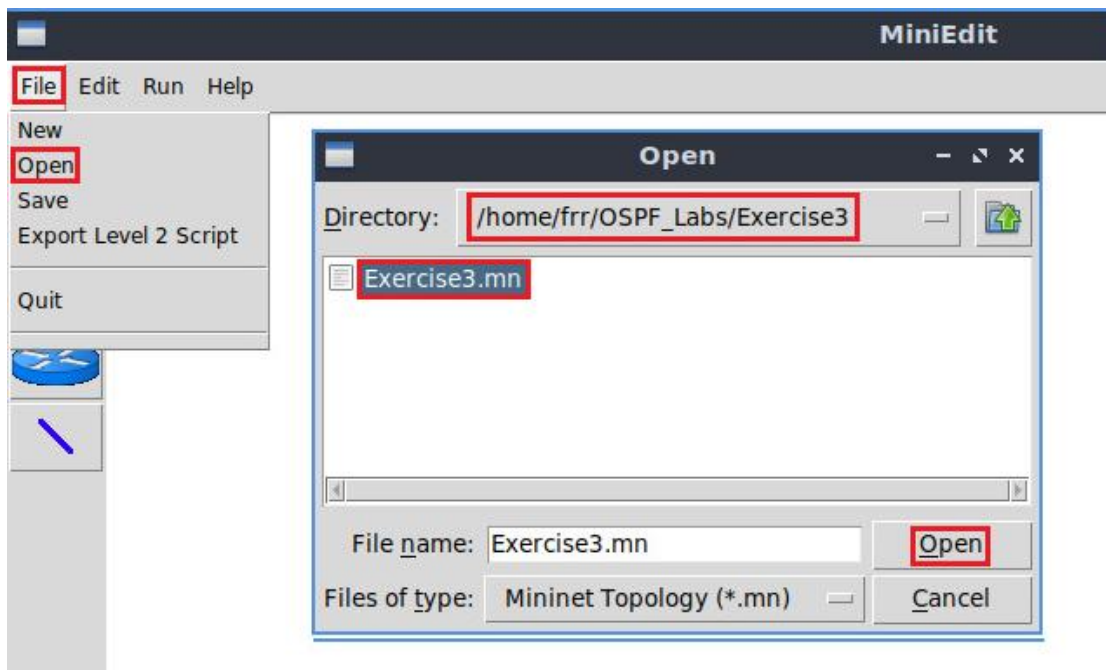


Figure 3. MiniEdit's open dialog.

**Step 3.** Run the emulation by clicking on the button located on the lower left-hand side.



Figure 4. Starting the emulation.

**Step 4.** Open the Linux terminal by clicking on the icon located in the taskbar.

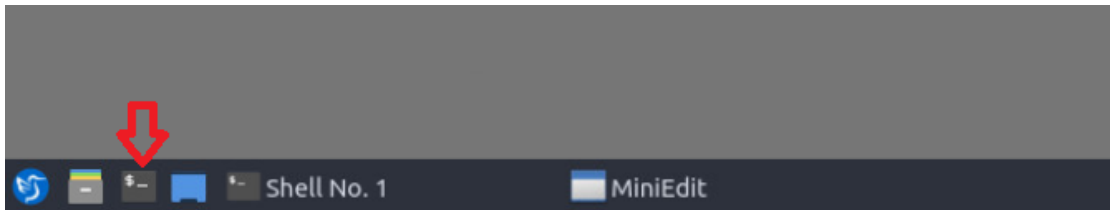


Figure 5. Opening Linux terminal.

**Step 5.** Issue the command shown below to set the lab environment. This command configures the IP addresses in all routers. If a password is required, type `password`.

```
sudo set_env exercise3
```

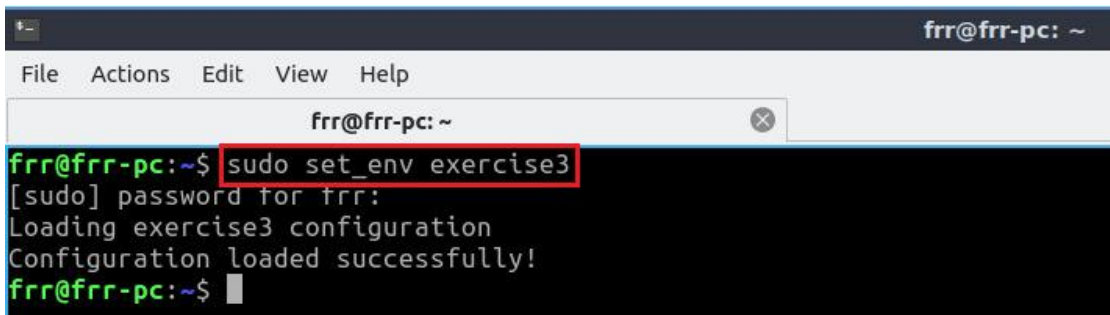


Figure 6. Setting the lab environment.

**Step 6.** Type the following command to exit the Linux terminal.

```
exit
```

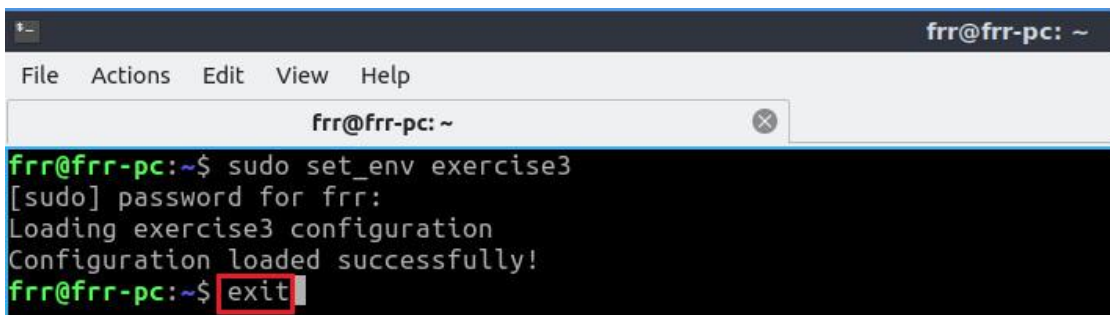


Figure 7. Exiting the Linux terminal.

### 3 Deliverables

Follow the steps below to complete the exercise.

- a) Open a terminal in each router, then, start zebra and the OSPFv3 daemons.
- b) Access each router CLI using the `vttysh` command.
- c) Verify the directly connected routes in each router.
- d) Assign an OSPF router ID to each router.
- e) Configure router r1 to participate in the OSPF process of areas 0 and 1.
- f) Configure router r2 to participate in the OSPF process of area 0.
- g) Configure router r3 to participate in the OSPF process of areas 0 and 2.
- h) Set the passive interfaces in routers r1 and r3.
- i) Verify the routing and OSPF neighbor tables of each router.
- j) Perform a connectivity test from host h1 to host h2.



UNIVERSITY OF  
**SOUTH CAROLINA**

## OPEN SHORTEST PATH FIRST

### Lab 10: Configuring Dual Stack OSPF Routing

Document Version: **10-11-2021**



Award 1829698

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



## Contents

Overview .....	3
Objectives.....	3
Lab settings .....	3
Lab roadmap .....	3
1 Introduction .....	3
1.1 Advantages of dual stack routing.....	4
1.2 Free range routing (FRR) and OSPFv3 .....	4
2 Lab topology.....	4
2.1 Lab settings.....	5
2.2 Loading the topology.....	6
2.3 Load the zebra daemon and verify the configuration .....	9
3 Configuring single-area OSPFv2.....	13
3.1 Configuring OSPFv2 in routers r1, r2, and r3 .....	13
3.2 Verifying IPv4 routing information and connectivity.....	17
4 Configuring single-area OSPFv3.....	19
4.1 Configuring OSPFv3 in routers r1, r2, and r3 .....	19
4.2 Verifying IPv6 routing information and connectivity.....	23
References .....	25

## Overview

This lab introduces dual stack routing using OSPF, where the routers involved in the OSPF process forward IPv4 and IPv6 traffic. Dual stack routing is a method that can facilitate the transition from IPv4 to IPv6. The user will configure the routers participating in the OSPF process to handle IPv4 and IPv6 packets in this lab.

## Objectives

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

1. Understand the concept of dual stack routing.
2. Configure two OSPF process per instance to handle IPv4 and IPv6 traffic.
3. Understand the scope and limitations of dual stack routing.
4. Verify both OSPv2 and OSPv3 configurations.

## Lab settings

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

Table 1. Credentials to access the Client 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 single-area OSPFv2.
4. Section 4: Configuring single-area OSPFv3.

### 1 Introduction

The Internet Protocol version 6 (IPv6) has been developed to solve the architectural limitations of IPv4. One of the main characteristics of IPv6 is the provision of a more significant number of addresses compared to IPv4. Although the IPv6 adoption over the Internet remains small, major network operators and content providers are increasingly using IPv6 in their deployments<sup>10</sup>.

IPv4 is the protocol that the Internet has used since its very early years. Therefore, it is the most important and most widely deployed networking protocol in existence. Although it has been said that IPv4 addresses are scarce and already exhausted, the Network Address Translation (NAT) technology has extended IPv4's life<sup>11</sup>. Using the dual stack approach is a method to achieve the coexistence between IPv4 and IPv6. Utilizing dual stack implies that the network hardware can handle IPv4 and IPv6 traffic simultaneously. Therefore, this approach can facilitate the transition from IPv4 to IPv6<sup>11,12,13</sup>.

## 1.1 Advantages of dual stack routing

The dual stack approach can be a cost-effective and straightforward method for migrating to IPv6. It enables high processing efficiency and reduces information loss by eliminating the need for translation between two protocol stacks. The translation refers to a mechanism that introduces operational complexity and lower performance. Moreover, implementing services over IPv6 can increasingly reduce the usage of IPv4, paving the way to fully migrate to IPv6.

Nevertheless, the dual stack approach must not be considered a long-term solution as it requires the implementation of routing protocol for IPv4 and IPv6. This limitation increases the time that it takes to implement and configure devices that handle both protocols.

## 1.2 Free range routing (FRR) and OSPFv3

The OSPF version 3 implementation available in FRR is based on the RFC 2740<sup>3</sup>, which only supports IPv6 routing. However, FRR supports running and configuring multiple routing daemons in one instance (i.e., a router). This characteristic allows implementing dual stack routing.

## 2 Lab topology

Consider Figure 1. The topology consists of four networks connected to their respective routers. The communication between networks is established via single-area OSPFv2, and OSPFv3 configured in all routers. Note that end hosts interfaces have both IPv4 and IPv6 addresses.

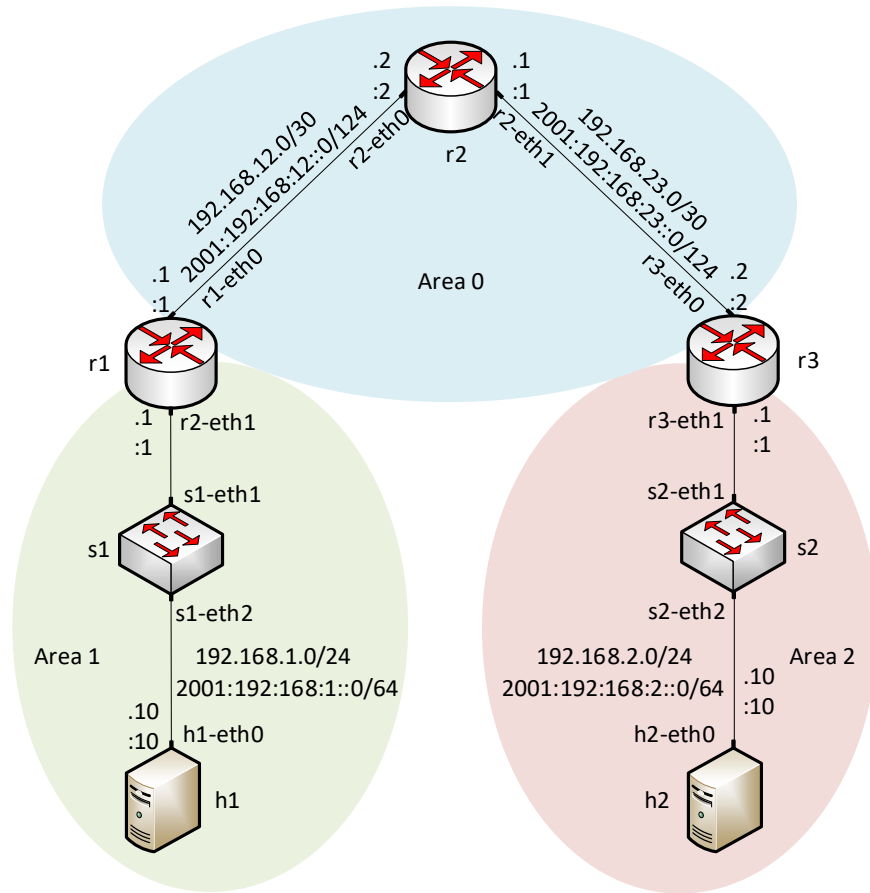


Figure 1. Lab topology.

## 2.1 Lab settings

Table 2 and Table 3 contain information related to the interfaces of all devices in the topology. Additionally, the tables have the IPv4, IPv6 addresses, subnet, and default gateways.

Table 2. Topology information.

Device	Interface	IPv4 Address	Subnet	Default Gateway
Router 1 (r1)	r1-eth0	192.168.12.1	/30	N/A
	r1-eth1	192.168.1.0	/24	N/A
Router 2 (r2)	r2-eth0	192.168.12.2	/30	N/A
	r2-eth1	192.168.23.1	/30	N/A
Router 3 (r3)	r3-eth0	192.168.23.2	/30	N/A
	r3-eth1	192.168.2.1	/24	N/A

Host 1 (h1)	h1-eth0	192.168.1.10	/24	192.168.1.1
Host 2 (h2)	h2-eth0	192.168.2.10	/24	192.168.2.1

Table 3. Topology information.

Device	Interface	IPv6 Address	Subnet	Default Gateway
Router 1 (r1)	r1-eth0	2001:192:168:12::1	/124	N/A
	r1-eth1	2001:192:168:1::1	/64	N/A
Router 2 (r2)	r2-eth0	2001:192:168:12::2	/124	N/A
	r2-eth1	2001:192:168:23::1	/124	N/A
Router 3 (r3)	r3-eth0	2001:192:168:23::2	/124	N/A
	r3-eth1	2001:192:168:2::1	/64	N/A
Host 1 (h1)	h1-eth0	2001:192:168:1::10	/64	2001:192:168:1::1
Host 2 (h2)	h2-eth0	2001:192:168:2::10	/64	2001:192:168:2::1

## 2.2 Loading the topology

**Step 1.** Launch MiniEdit by double clicking on the shortcut located on the desktop. When prompted for a password, type `password`.



Figure 2. MiniEdit shortcut.

**Step 2.** In MiniEdit's menu bar, click on *File* then *Open* to load the lab topology. Locate the `lab10.mn` topology file in the default directory, `/home/fr/OSPF_Labs/lab10` and click on *Open*.

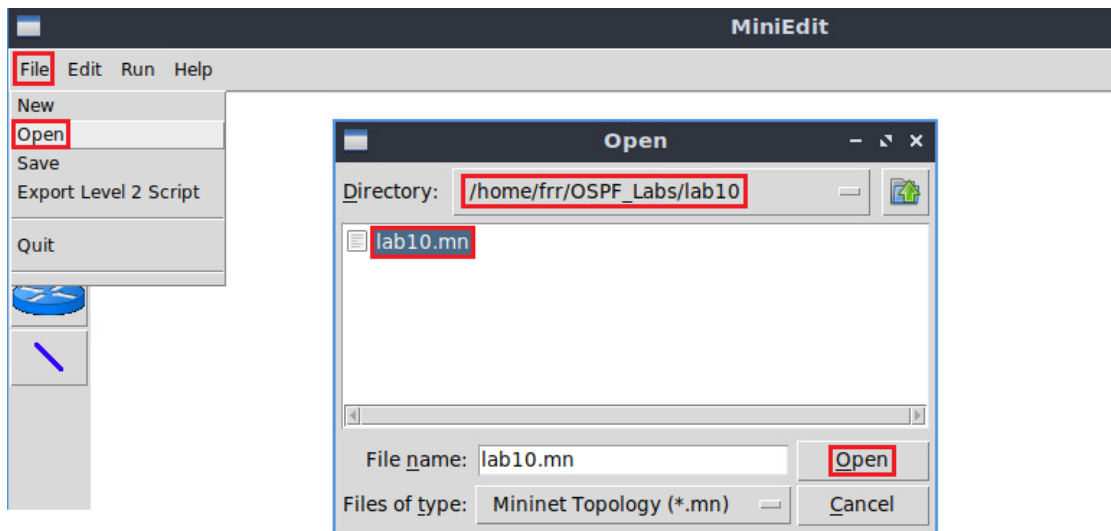


Figure 3. MiniEdit's open dialog.

At this point, the topology is loaded with all the required network components. Next, you will execute a script that will load the configuration of the routers.

**Step 3.** Click on the *Run* button located on the lower left-hand side to run the emulation.



Figure 4. Starting the emulation.

At this point, the topology is loaded with all the required network components. Next, you will execute a script that will configure the interfaces of the routers.

Automating the configuration of interfaces with the script is intended to facilitate the lab experience and to focus on the OSPF experiment.

**Step 4.** Open the Linux terminal by clicking on the icon located in the taskbar.

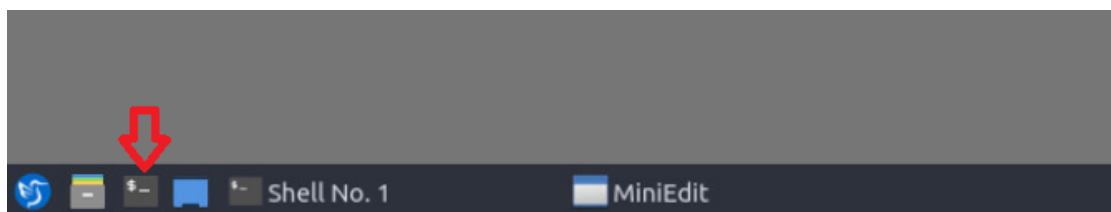


Figure 5. Opening Linux terminal.

**Step 5.** Issue the command shown below to set the lab environment. This command configures the IP addresses of the routers' interface according to Table 2 and Table 3. If a password is required, type `password`.

```
sudo set_env lab10
```

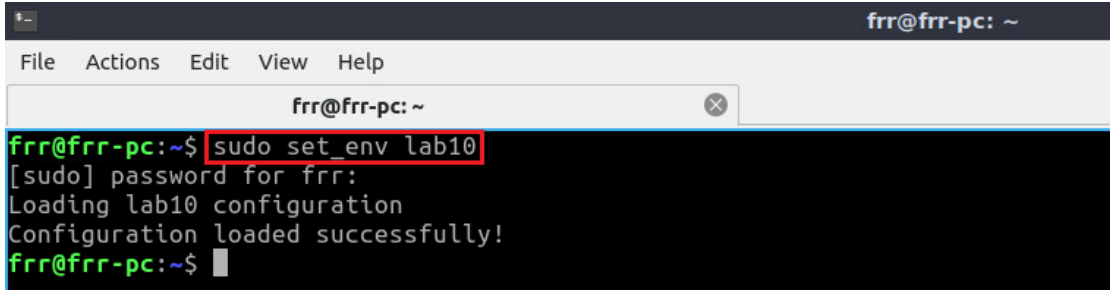


Figure 6. Setting the lab environment.

**Step 6.** Type the following command to exit the Linux terminal.

```
exit
```

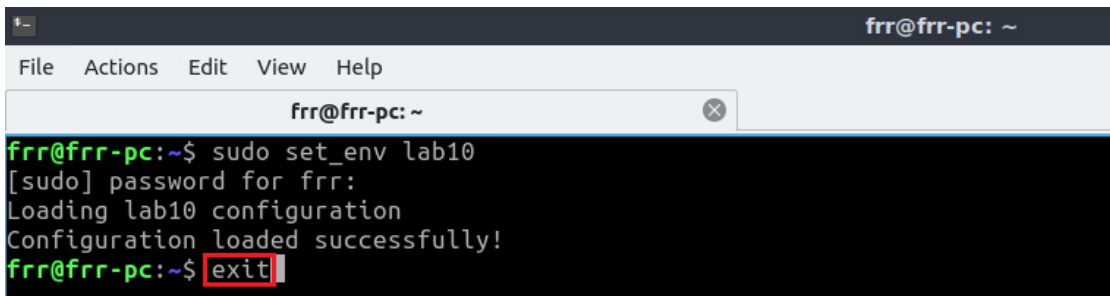


Figure 7. Exiting the Linux terminal.

**Step 7.** Click on Mininet's terminal, i.e., the one launched when MiniEdit was started.



Figure 8. Opening Mininet's terminal.

**Step 8.** Issue the following command to display the interface names and connections.

```
links
```

```

mininet> links
r1-eth0<->r2-eth0 (OK OK)
r2-eth1<->r3-eth0 (OK OK)
r1-eth1<->s1-eth1 (OK OK)
s1-eth2<->h1-eth0 (OK OK)
r3-eth1<->s2-eth1 (OK OK)
s2-eth2<->h2-eth0 (OK OK)
mininet>
  
```

Figure 9. Displaying network interfaces.

In the figure above, the link displayed within the gray box indicates that interface *eth2* of switch *s1* connects to interface *eth0* of host *h1* (i.e., *s1-eth2<->h1-eth0*).

### 2.3 Load the zebra daemon and verify the configuration

In this section, you will verify that the IP addresses listed in Table 2 and inspect the routing table of the routers.

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

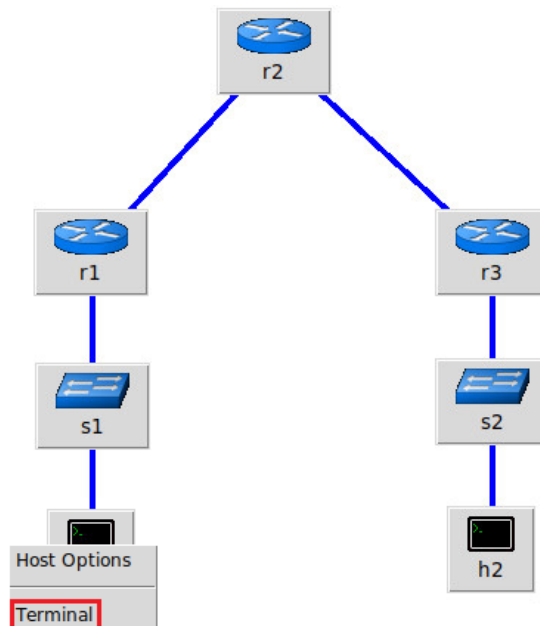


Figure 10. Opening host *h1* terminal.

**Step 2.** In the host *h1* terminal, type the command shown below to verify that the IP address was assigned successfully.

```
ifconfig
```



```

Host: h1
root@frr-pc:~# ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
  inet 192.168.1.10 netmask 255.255.255.0 broadcast 192.168.1.255
  inet6 fe80::d858:83ff:fe72:7576 prefixlen 64 scopeid 0x20<link>
  inet6 2001:192:168:1::10 prefixlen 64 scopeid 0x0<global>
  ether 00:00:00:00:01:10 txqueuelen 1000 (Ethernet)
  RX packets 33 bytes 4104 (4.1 KB)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 16 bytes 1312 (1.3 KB)
  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@frr-pc:~#

```

Figure 11. Displaying host h1 interfaces and IP addresses.

You will verify that host h1 has two interfaces. Interface *h1-eth0* is configured with the IPv4 address 192.168.1.10, the link-local IPv6 address fe80::d858:83ff:fe72:7576, and the IPv6 global unicast address 2001:192:168:1::10. The prefix length for IPv6 addresses is 64. The other interface is the loopback *lo*.

A link-local address is an IPv6 unicast address that can be configured on any interface. It is used to reach the neighboring nodes attached to the same link. The IPv6 link-local addresses start with the FE80 prefix.

**Step 3.** In the host h1 terminal, type the following command to verify that the default IPv4 gateway address is 192.168.1.1.

```

Host: h1
root@frr-pc:~# 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@frr-pc:~#

```

Figure 12. Displaying host h1 default IPv4 route.

**Step 4.** In the host h1 terminal, type the command shown below to verify that the default IPv6 gateway address is 2001:192:168:1::1.

```
ip -6 route
```

```

Host: h1
root@frr-pc:~# ip -6 route
2001:192:168:1::/64 dev h1-eth0 proto kernel metric 256 pref medium
fe80::/64 dev h1-eth0 proto kernel metric 256 pref medium
default via 2001:192:168:1::1 dev h1-eth0 metric 1 pref medium
root@frr-pc:~#
    
```

Figure 13. Displaying host h1 default IPv6 route.

**Step 5.** To verify the hosts h2 and h3 default route, proceed by following from step 1 to step 4. Similar results should be observed.

**Step 6.** You will validate that the router interfaces are configured according to Table 2. To proceed, hold right-click on router r1 and select *Terminal*.

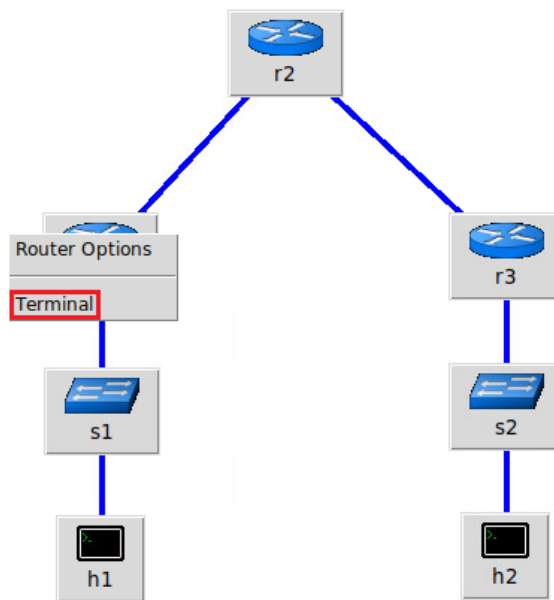


Figure 14. Opening a terminal on router r1.

**Step 7.** In this step, you will start the zebra daemon, a multi-server routing software that provides TCP/IP-based routing protocols. The configuration will not be working if you do not enable zebra daemon initially. To start the zebra, type the following command:

```
zebra
```

```

Host: r1
root@frr-pc:/etc/routers/r1# zebra
root@frr-pc:/etc/routers/r1#
    
```

Figure 15. Starting zebra daemon.

**Step 8.** After initializing zebra, vtysh should be started to provide all the CLI commands defined by the daemons. To proceed, issue the following command:

```
vtysh
```

```

Host: r1
root@frr-pc:/etc/routers/r1# zebra
root@frr-pc:/etc/routers/r1# vtysh
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.
frr-pc#

```

Figure 16. Starting vtysh on router r1.

**Step 9.** Type the following command on router r1 terminal to verify the IPv4 routing table. It will list all the directly connected networks. Notice that the routing table of router r1 does not contain any route to the network connected to router r3 (192.168.2.0/24) as there is no routing protocol configured yet.

```
show ip route
```

```

Host: r1
frr-pc# 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-eth1, 00:00:09
C>* 192.168.12.0/30 is directly connected, r1-eth0, 00:00:09
frr-pc#

```

Figure 17. Displaying the IPv4 routing table of router r1.

**Step 10.** Type the following command on router r1 terminal to verify the IPv6 routing table. It will list all the directly connected networks. Notice that the routing table of router r1 does not contain any route to the network connected to router r3 (2001:192:168:2::0/64) as there is no routing protocol configured yet. The first two entries correspond to the directly connected networks. The last two entries correspond to link-local network connected to each interface.

```
show ipv6 route
```

```

Host: r1
frr-pc# show ipv6 route
Codes: K - kernel route, C - connected, S - static, R - RIPng,
       O - OSPFv3, I - IS-IS, B - BGP, 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>* 2001:192:168:1::/124 is directly connected, r1-eth1, 00:00:15
C>* 2001:192:168:12::/124 is directly connected, r1-eth0, 00:00:15
C * fe80::/64 is directly connected, r1-eth1, 00:00:15
C>* fe80::/64 is directly connected, r1-eth0, 00:00:15
frr-pc#
    
```

Figure 18. Displaying the IPv6 routing table of router r1.

**Step 11.** To verify the routing table of the other routers, proceed similarly by following from step 6 to step 10 on the corresponding router terminal.

### 3 Configuring single-area OSPFv2

In this section, you will configure r1, r2, and r3 to participate in the OSPFv2 process of area 0.

#### 3.1 Configuring OSPFv2 in routers r1, r2, and r3

**Step 1.** Open router r1's terminal by selecting the icon from the taskbar as shown below.



Figure 19. Opening router r1's terminal.

**Step 2.** In router r1's terminal, type the command below to exit the vtysh session.

```

exit

```

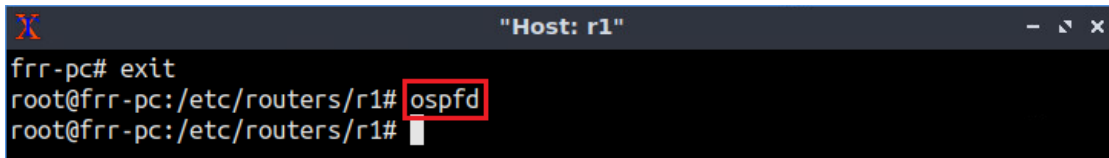
```

Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1#
    
```

Figure 20. Exiting the vtysh session.

**Step 3.** Enable and start the OSPF routing protocol by typing the following command:

```
ospfd
```

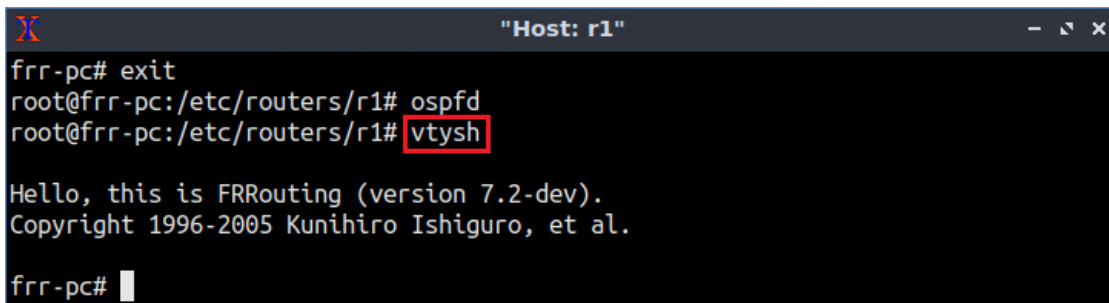


```
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1#
```

Figure 21. Starting the OSPF daemon.

**Step 4.** To enter router r1's terminal, type the following command:

```
vttysh
```



```
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vttysh

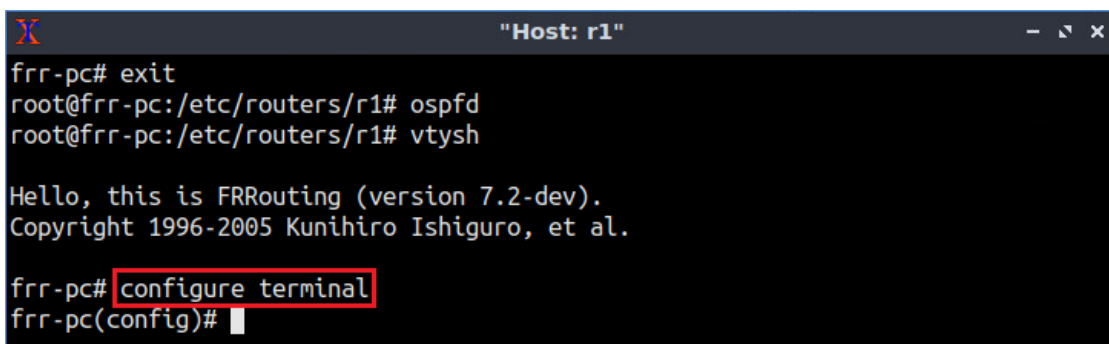
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc#
```

Figure 22. Starting vtysh in router r1.

**Step 5.** At this point, you have entered router r1's terminal, to enter configuration mode in router r1, issue the following command:

```
configure terminal
```



```
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vttysh

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

frr-pc# configure terminal
frr-pc(config)#
```

Figure 23. Entering configuration mode in router r1.

**Step 6.** To configure OSPF, type the command shown below. This command will enable OSPF configuration mode.

```
router ospf
```

```

Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)#

```

Figure 24. Configuring OSPF in router r1.

**Step 7.** Type the following command to set an ID to router r1.

```
ospf router-id 1.1.1.1
```

```

Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 1.1.1.1
frr-pc(config-router)#

```

Figure 25. Setting an ID to router r1.

**Step 8.** Type the following command to enable interface *r1-eth0* to participate in the OSPF process corresponding to area 0.

```
network 192.168.12.0/30 area 0
```

```

Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

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

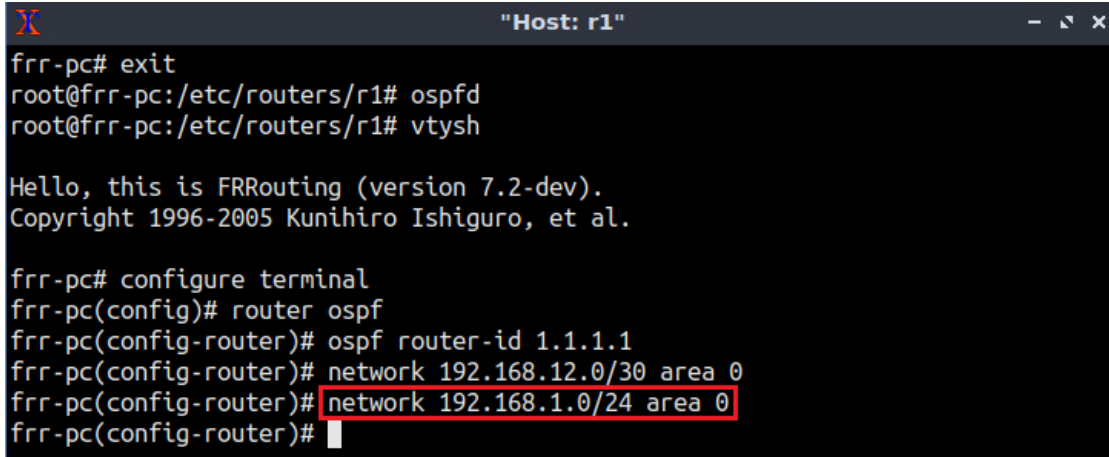
frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 1.1.1.1
frr-pc(config-router)# network 192.168.12.0/30 area 0
frr-pc(config-router)#

```

Figure 26. Enabling interface *r1-eth0* to participate in the OSPF process.

**Step 9.** Type the following command to enable interface *r1-eth1* to participate in the OSPF process corresponding to area 0.

```
network 192.168.2.0/24 area 0
```



```

Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 1.1.1.1
frr-pc(config-router)# network 192.168.12.0/30 area 0
frr-pc(config-router)# network 192.168.1.0/24 area 0
frr-pc(config-router)#

```

Figure 27. Enabling interface *r1-eth1* to participate in the OSPF process.

**Step 10.** Type the following command to exit router configuration mode.

```
end
```



```

Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospfd
root@frr-pc:/etc/routers/r1# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 1.1.1.1
frr-pc(config-router)# network 192.168.12.0/30 area 0
frr-pc(config-router)# network 192.168.1.0/24 area 0
frr-pc(config-router)# end
frr-pc#

```

Figure 28. Exiting router configuration mode.

Consider the figure above. After configuring OSPF in router *r1*, the interfaces *r1-eth0*, and *r1-eth1* will participate in the routing process of area 0 and area 1 respectively.

**Step 11.** 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* terminal, issue the commands summarized below.

```
Host: r2
frr-pc# exit
root@frr-pc:/etc/routers/r2# ospfd
root@frr-pc:/etc/routers/r2# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 2.2.2.2
frr-pc(config-router)# network 192.168.12.0/30 area 0
frr-pc(config-router)# network 192.168.23.0/30 area 0
frr-pc(config-router)# end
frr-pc#
```

Figure 29. Router r2 configuration summary.

Consider the figure above. After configuring OSPF in router r2, the interfaces *r2-eth0*, and *r2-eth1* will participate in the routing process of area 0.

**Step 12.** Router r3 is configured similarly to router r2 but with different IP addresses (see Table 2). Those steps are summarized in the following figure. To proceed, in router r3 terminal, issue the commands depicted below.

```
Host: r3
frr-pc# exit
root@frr-pc:/etc/routers/r3# ospfd
root@frr-pc:/etc/routers/r3# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# ospf router-id 3.3.3.3
frr-pc(config-router)# network 192.168.23.0/30 area 0
frr-pc(config-router)# network 192.168.2.0/24 area 0
frr-pc(config-router)# end
frr-pc#
```

Figure 30. Router r3 configuration summary.

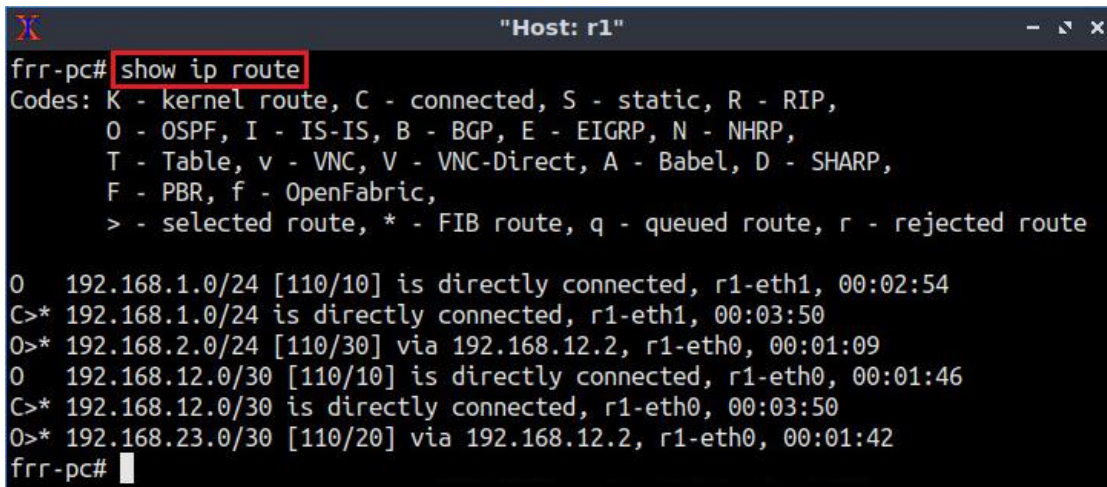
Consider the figure above. After configuring OSPF in router r3, the interfaces *r3-eth0* and *r3-eth1* will participate in the routing process of area.

### 3.2 Verifying IPv4 routing information and connectivity

**Step 1.** Issue the following command to display the IPv4 routing table of router r1.

```
show ip route
```





```

Host: r1
frr-pc# 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

O 192.168.1.0/24 [110/10] is directly connected, r1-eth1, 00:02:54
C>* 192.168.1.0/24 is directly connected, r1-eth1, 00:03:50
O>* 192.168.2.0/24 [110/30] via 192.168.12.2, r1-eth0, 00:01:09
O 192.168.12.0/30 [110/10] is directly connected, r1-eth0, 00:01:46
C>* 192.168.12.0/30 is directly connected, r1-eth0, 00:03:50
O>* 192.168.23.0/30 [110/20] via 192.168.12.2, r1-eth0, 00:01:42
frr-pc#

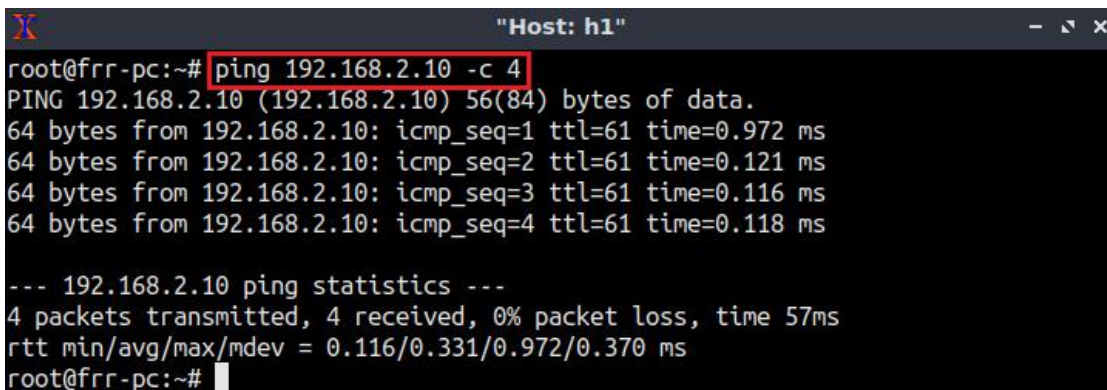
```

Figure 31. Displaying router r1's IPv4 routing table.

Consider the figure above. The output displays the networks learned along with additional attributes, such as the routing protocol that discovered the network. For example, networks with the symbol `O` on the left are learned from OSPF, whereas those with symbol `C` are directly connected to the device. Furthermore, the symbol `>` on the left of the displayed network indicates that this entry is selected to reach the network. As for the symbol `*`, this indicates that the route is inserted in Forwarding Information Base (FIB) of the router.

**Step 2.** In the host h1 terminal, issue the following command to test the connectivity between host h1 and host h2. You will observe a successful connectivity test.

```
ping 192.168.2.10 -c 4
```



```

Host: h1
root@frr-pc:~# ping 192.168.2.10 -c 4
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=61 time=0.972 ms
64 bytes from 192.168.2.10: icmp_seq=2 ttl=61 time=0.121 ms
64 bytes from 192.168.2.10: icmp_seq=3 ttl=61 time=0.116 ms
64 bytes from 192.168.2.10: icmp_seq=4 ttl=61 time=0.118 ms

--- 192.168.2.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 57ms
rtt min/avg/max/mdev = 0.116/0.331/0.972/0.370 ms
root@frr-pc:~#

```

Figure 32. Testing connectivity from host h1 to host h2 over IPv4.

**Step 3.** In the host h1 terminal, issue the following command to track the route that packets take on the way to their destination, in this case host h2 (192.168.2.10). Notice that it takes four hops to reach host h2 from host h1.

```
traceroute 192.168.1.10
```

```

Host: h1
root@frr-pc:~# traceroute 192.168.2.10
traceroute to 192.168.2.10 (192.168.2.10), 30 hops max, 60 byte packets
 1 192.168.1.1 (192.168.1.1)  1.654 ms  1.642 ms  1.639 ms
 2 192.168.12.2 (192.168.12.2)  1.652 ms  1.652 ms  1.679 ms
 3 192.168.23.2 (192.168.23.2)  1.702 ms  1.698 ms  1.679 ms
 4 192.168.2.10 (192.168.2.10)  1.812 ms  1.806 ms  1.798 ms
root@frr-pc:~#
    
```

Figure 33. Performing a traceroute test over IPv4.

**Step 4.** In router r2 terminal, issue the following command to see the OSPF neighbors.

```
show ip ospf neighbor
```

```

Host: r2
frr-pc# show ip ospf neighbor
Neighbor ID    Pri  State           Dead Time Address          Interface          RXmtL  RqstL  DBsmL
1.1.1.1        1    Full/Backup     31.348s  192.168.12.1    r2-eth0:192.168.12.2  0      0      0
3.3.3.3        1    Full/DR         30.981s  192.168.23.2    r2-eth1:192.168.23.1  0      0      0
frr-pc#
    
```

Figure 34. Showing router r2 OSPFv2 neighbors.

## 4 Configuring single-area OSPFv3

In this section, you will configure r1, r2, and r3 to participate in the OSPFv3 process in area 0.

### 4.1 Configuring OSPFv3 in routers r1, r2, and r3

**Step 1.** Open router r1's terminal by selecting the icon from the taskbar as shown below.



Figure 35. Opening router r1's terminal.

**Step 2.** In router r1's terminal, type the command below to exit the vtysh session.

```
exit
```

```

Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1#
    
```

Figure 36. Exiting the vtysh session.

**Step 3.** Enable and start the OSPF routing protocol by typing the following command:

```
ospf6d
```

```

Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospf6d
root@frr-pc:/etc/routers/r1#
    
```

Figure 37. Starting the OSPF daemon.

**Step 4.** To enter router r1's terminal, type the following command:

```
vttysh
```

```

Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospf6d
root@frr-pc:/etc/routers/r1# vtysh

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

frr-pc#
    
```

Figure 38. Starting vtysh in router r1.

**Step 5.** At this point, you have entered router r1's terminal, to enter configuration mode in router r1, issue the following command:

```
configure terminal
```

```

Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospf6d
root@frr-pc:/etc/routers/r1# vtysh

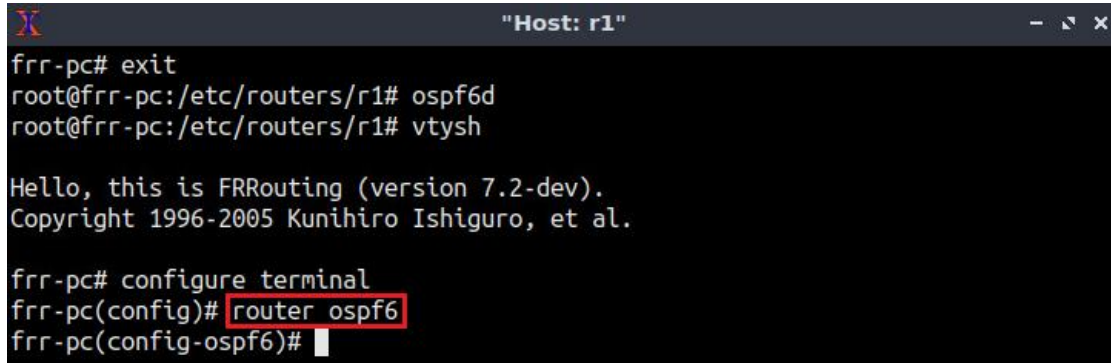
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)#
    
```

Figure 39. Entering configuration mode in router r1.

**Step 6.** To configure OSPF, type the command shown below. This command will enable OSPF configuration mode.

```
router ospf6
```



```

Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospf6
root@frr-pc:/etc/routers/r1# vtysh

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

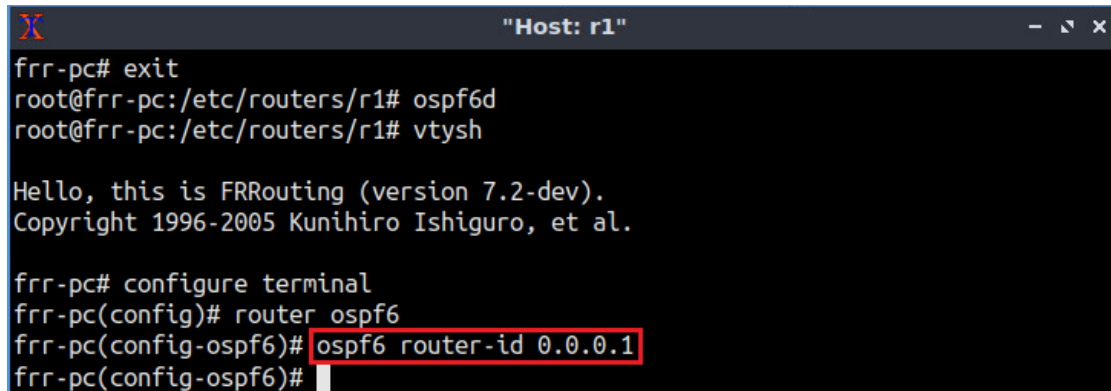
frr-pc# configure terminal
frr-pc(config)# router ospf6
frr-pc(config-ospf6)#

```

Figure 40. Configuring OSPF in router r1.

**Step 7.** Type the following command to set an ID to router r1.

```
ospf6 router-id 0.0.0.1
```



```

Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospf6
root@frr-pc:/etc/routers/r1# vtysh

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

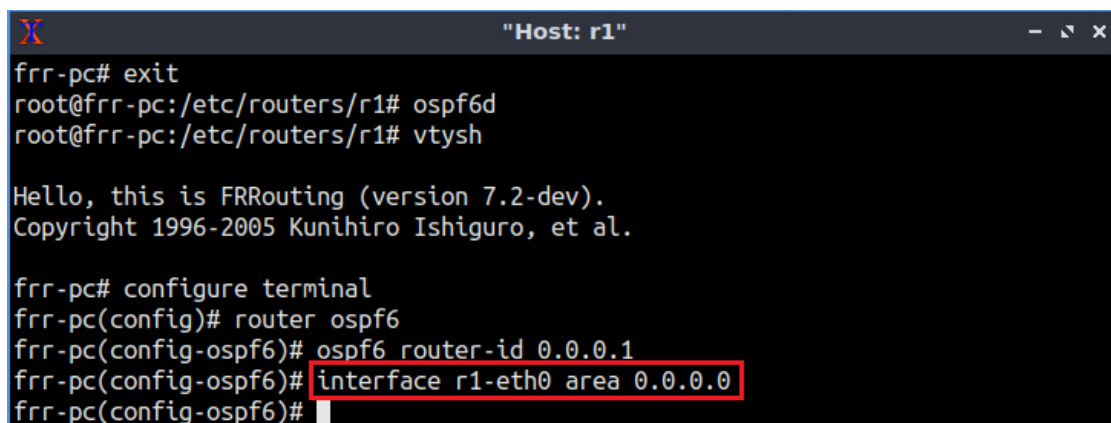
frr-pc# configure terminal
frr-pc(config)# router ospf6
frr-pc(config-ospf6)# ospf6 router-id 0.0.0.1
frr-pc(config-ospf6)#

```

Figure 41. Setting an ID to router r1.

**Step 8.** Type the following command to enable interface *r1-eth0* to participate in the OSPF process corresponding to the area 0.0.0.0.

```
interface r1-eth0 area 0.0.0.0
```



```

Host: r1
frr-pc# exit
root@frr-pc:/etc/routers/r1# ospf6
root@frr-pc:/etc/routers/r1# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf6
frr-pc(config-ospf6)# ospf6 router-id 0.0.0.1
frr-pc(config-ospf6)# interface r1-eth0 area 0.0.0.0
frr-pc(config-ospf6)#

```

Figure 42. Enabling interface *r1-eth0* to participate in the OSPF process.

**Step 9.** Type the following command to enable interface *r1-eth1* to participate in the OSPF process corresponding to the area 0.0.0.0.

```
interface r1-eth1 area 0.0.0.0
```

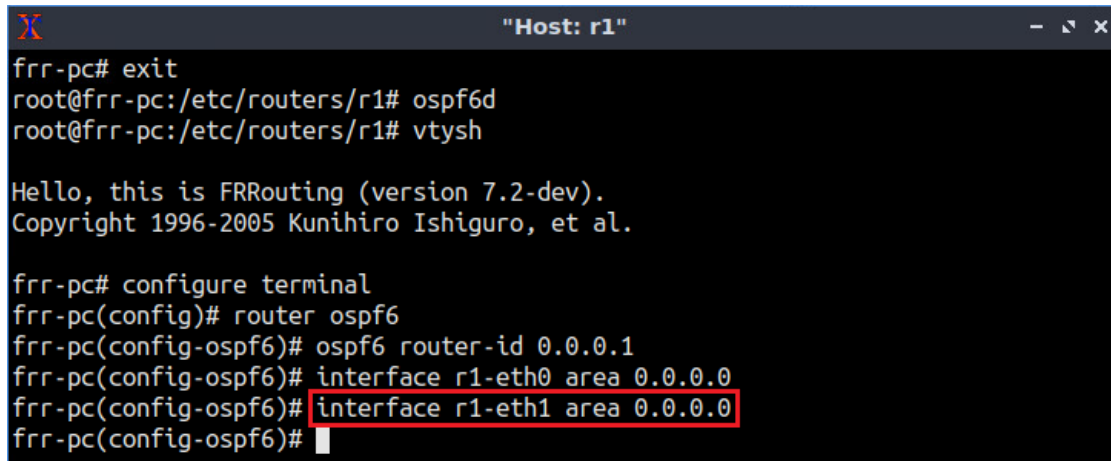
A terminal window titled "Host: r1" showing the configuration of OSPF on a router. The user enters 'exit' to leave the previous configuration, then 'ospf6' to enter the OSPF6 configuration mode, and 'vtysh' to enter the configuration terminal. The terminal displays the FRRouting version 7.2-dev and copyright information. The user then enters 'configure terminal', 'router ospf6', 'ospf6 router-id 0.0.0.1', 'interface r1-eth0 area 0.0.0.0', and finally 'interface r1-eth1 area 0.0.0.0', which is highlighted with a red box. The prompt returns to 'frr-pc(config-ospf6)#'.

Figure 43. Enabling interface *r1-eth1* to participate in the OSPF process.

**Step 10.** Type the following command to exit router configuration mode.

```
end
```

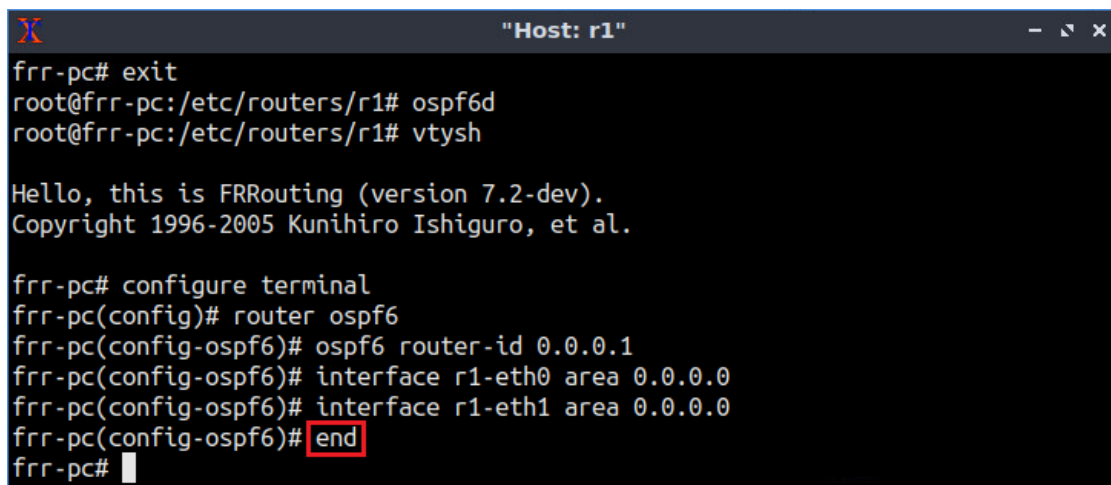
A terminal window titled "Host: r1" showing the continuation of the OSPF configuration. The user enters 'end' to exit the configuration mode, which is highlighted with a red box. The terminal returns to the 'frr-pc#' prompt.

Figure 44. Exiting router configuration mode.

Consider the figure above. After configuring OSPF in router *r1*, the interfaces *r1-eth0*, and *r1-eth1* will participate in the routing process of area 0.0.0.0.

**Step 11.** 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* terminal, issue the commands summarized below.

```
Host: r2
frr-pc# exit
root@frr-pc:/etc/routers/r2# ospf6d
root@frr-pc:/etc/routers/r2# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf6
frr-pc(config-ospf6)# ospf6 router-id 0.0.0.2
frr-pc(config-ospf6)# interface r2-eth0 area 0.0.0.0
frr-pc(config-ospf6)# interface r2-eth1 area 0.0.0.0
frr-pc(config-ospf6)# end
frr-pc#
```

Figure 45. Router r2 configuration summary.

Consider the figure above. After configuring OSPF in router r2, the interfaces *r2-eth0*, and *r2-eth1* will participate in the routing process of area 0.0.0.0.

**Step 12.** Router r3 is configured similarly to router r2 but with different IP addresses (see Table 2). Those steps are summarized in the following figure. To proceed, in router r3 terminal, issue the commands depicted below.

```
Host: r3
frr-pc# exit
root@frr-pc:/etc/routers/r3# ospf6d
root@frr-pc:/etc/routers/r3# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router ospf6
frr-pc(config-ospf6)# ospf6 router-id 0.0.0.3
frr-pc(config-ospf6)# interface r3-eth0 area 0.0.0.0
frr-pc(config-ospf6)# interface r3-eth1 area 0.0.0.0
frr-pc(config-ospf6)# end
frr-pc#
```

Figure 46. Router r3 configuration summary.

Consider the figure above. After configuring OSPF in router r3, the interfaces *r3-eth0* and *r3-eth1* will participate in the routing process of area 0.0.0.0.

## 4.2 Verifying IPv6 routing information and connectivity

**Step 1.** Issue the following command to display the routing table of router r1.

```
show ipv6 route
```

```

Host: r1
frr-pc# show ipv6 route
Codes: K - kernel route, C - connected, S - static, R - RIPng,
       O - OSPFv3, I - IS-IS, B - BGP, 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

O   2001:192:168:1::/64 [110/10] is directly connected, r1-eth1, 00:00:22
C>* 2001:192:168:1::/64 is directly connected, r1-eth1, 00:00:22
O>* 2001:192:168:2::/64 [110/30] via fe80::2c32:81ff:fe80:7aba, r1-eth0, 00:00:21
O   2001:192:168:12::/124 [110/10] is directly connected, r1-eth0, 00:00:21
C>* 2001:192:168:12::/124 is directly connected, r1-eth0, 00:00:21
O>* 2001:192:168:23::/124 [110/20] via fe80::2c32:81ff:fe80:7aba, r1-eth0, 00:00:21
C * fe80::/64 is directly connected, r1-eth1, 00:40:47
C>* fe80::/64 is directly connected, r1-eth0, 00:40:47
frr-pc#

```

Figure 47. Displaying router r1's routing table.

Notice that reaching the network 2001:192:168:2::/64 is performed via fe80::2c32:81ff:fe80:7aba, which is the link-local IPv6 address of the interface r2-eth0 of router r2.

**Step 2.** In the host h1 terminal, issue the following command to test the connectivity between host h1 and host h2. You will observe a successful connectivity test.

```
ping 2001:192:168:2::10 -c 4
```

```

Host: h1
root@frr-pc:~# ping 2001:192:168:2::10 -c 4
PING 2001:192:168:2::10(2001:192:168:2::10) 56 data bytes
64 bytes from 2001:192:168:2::10: icmp_seq=1 ttl=61 time=0.104 ms
64 bytes from 2001:192:168:2::10: icmp_seq=2 ttl=61 time=0.129 ms
64 bytes from 2001:192:168:2::10: icmp_seq=3 ttl=61 time=0.092 ms
64 bytes from 2001:192:168:2::10: icmp_seq=4 ttl=61 time=0.081 ms

--- 2001:192:168:2::10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 61ms
rtt min/avg/max/mdev = 0.081/0.101/0.129/0.020 ms
root@frr-pc:~#

```

Figure 48. Testing connectivity from host h1 to host h2 over IPv6.

**Step 3.** In the host h1 terminal, issue the following command to track the route that packets take on the way to their destination, in this case host h2 (2001:192:168:2::10).

```
traceroute 2001:192:168:2::10
```

```

Host: h1
root@frr-pc:~# traceroute 2001:192:168:2::10
traceroute to 2001:192:168:2::10 (2001:192:168:2::10), 30 hops max, 80 byte packets
 1  2001:192:168:1::1 (2001:192:168:1::1)  1.680 ms  1.680 ms  1.687 ms
 2  2001:192:168:12::2 (2001:192:168:12::2)  1.694 ms  1.709 ms  1.711 ms
 3  2001:192:168:23::2 (2001:192:168:23::2)  1.717 ms  1.710 ms  1.706 ms
 4  2001:192:168:2::10 (2001:192:168:2::10)  1.813 ms  1.806 ms  1.797 ms
root@frr-pc:~#

```

Figure 49. Performing a traceroute test over IPv6.

Notice that it takes four hops to reach host h2 from host h1. The type of the IPv6 addresses is global unicast, which perform the same function of public IPv4 addresses and are also routable to the Internet.

**Step 4.** In router r2 terminal, issue the following command to see the OSPF neighbors.

```
show ipv6 ospf6 neighbor
```

```

Host: r2
frr-pc# show ipv6 ospf6 neighbor
Neighbor ID      Pri   DeadTime   State/IfState   Duration I/F[State]
0.0.0.1          1     00:00:37   Full/DR         00:20:12 r2-eth0[BDR]
0.0.0.3          1     00:00:31   Full/BDR        00:16:44 r2-eth1[DR]
frr-pc#

```

Figure 50. Testing connectivity from host h1 to host h2.

The figure above shows that router r1 and router r3 (Router IDs 0.0.0.1 and 0.0.0.3 respectively) are neighbors of router r2.

This concludes lab 10. Stop the emulation and then exit out of MiniEdit.

## References

1. Moy, J. "RFC 1247: OSPF version 2." 1991.
2. Moy, J. "RFC 2328: OSPF version 2." 1998.
3. Coltun, R., Ferguson, D., Moy, J., Lindem, A. "RFC 2740: OSPF for IPv6." 1999.
4. Coltun, R., Ferguson, D., Moy, J., Lindem, A. "RFC 5340: OSPF for IPv6." 2008.
5. Peterson, L. "Computer networks: a systems approach." 6th Edition, 2021.
6. Tanenbaum, D. Wetherall. "Computer networks." 5th Edition. 2012.
7. J. Kurose, K. Ross. "Computer networking, a top-down approach." 7th Edition. 2017.
8. Teare, D., Vachon, B., Graziani, R. "Implementing Cisco IP routing (ROUTE) foundation learning guide: (CCNP ROUTE 300-101)." 2014.
9. Linux foundation collaborative projects. "FRR routing documentation." 2017. [Online], Available: <http://docs.frrouting.org/en/latest/>.
10. Google, "Google ipv6 implementors conference," 2010. [Online]. Available: <https://sites.google.com/site/ipv6implementors/2010/agenda>.



11. Karpilovsky, E., Gerber A., Pei D., Rexford J., Shaikh, A. "*Quantifying the extent of IPv6 deployment.*" In International Conference on Passive and Active Network Measurement. 2009.
12. Ram, M., "*Will U.S. government directives spur IPv6 adoption?*" 2010. [Online]. Available: [http://www.circleid.com/posts/20100929\\_will\\_us\\_government\\_directives\\_spur\\_ipv6\\_adoption/](http://www.circleid.com/posts/20100929_will_us_government_directives_spur_ipv6_adoption/).
13. Jakub C., Allman M., Zhang J., Lelak-Johnson S., Osterweil E., Bailey, M. "*Measuring IPv6 adoption.*" In Proceedings of the 2014 ACM Conference on SIGCOMM, 2014.