



BORDER GATEWAY PROTOCOL

Lab 1: Introduction to Mininet

Document Version: **02-17-2020**



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	17
4 Configure router r1	18
4.1 Verify end-hosts configuration.....	19
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, students 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 machine containing Mininet.

Table 1. Credentials to access Client1 machine.

Device	Account	Password
Client1	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 networks¹. 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.

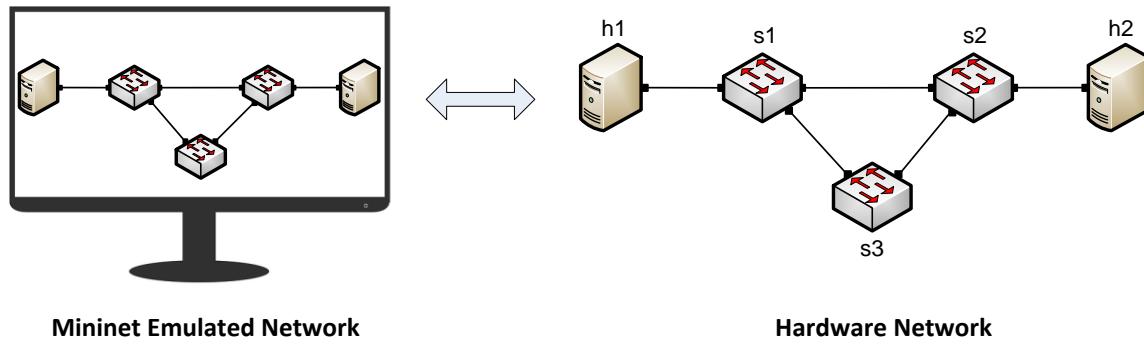


Figure 1. Hardware network vs. Mininet emulated network.

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*² and *Software-Defined Networking (SDN)*³. 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 that 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 switch⁴. 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.

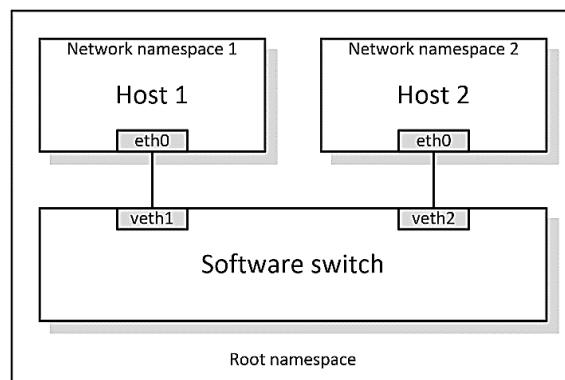


Figure 2. Network namespaces and virtual Ethernet links.

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.

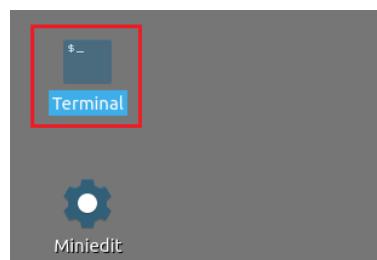
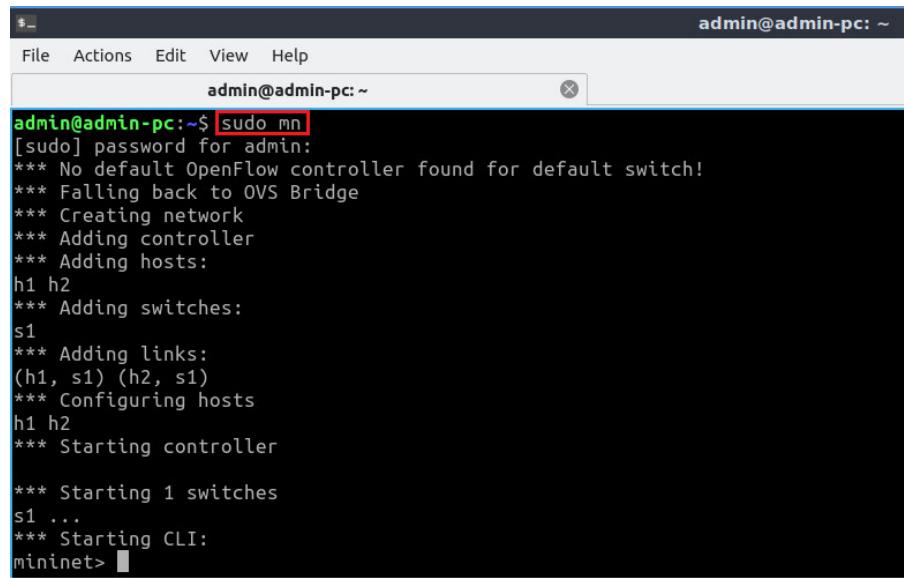


Figure 3. Shortcut to open a Linux terminal.

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
```



```
admin@admin-pc:~$ sudo mn
[sudo] password for admin:
*** 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.

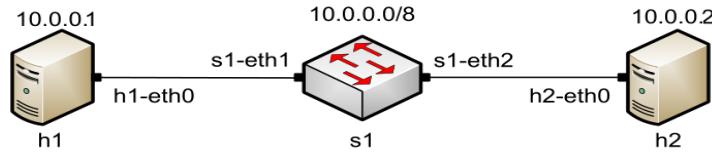


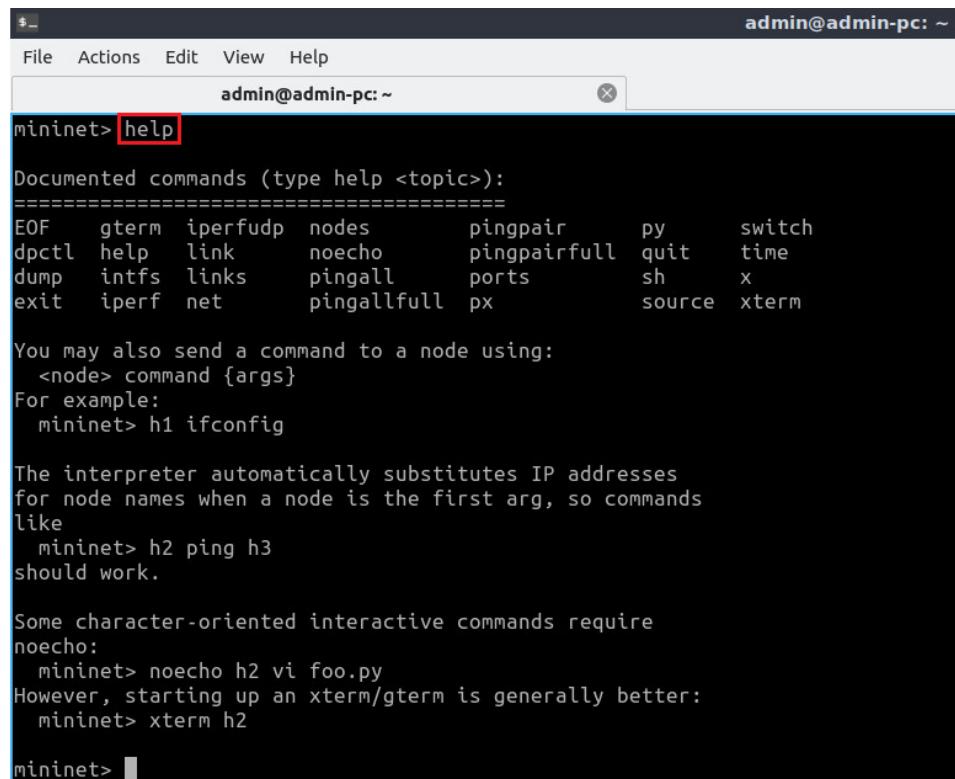
Figure 5. Mininet's default minimal topology.

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
```



```
admin@admin-pc: ~
File Actions Edit View Help
admin@admin-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.

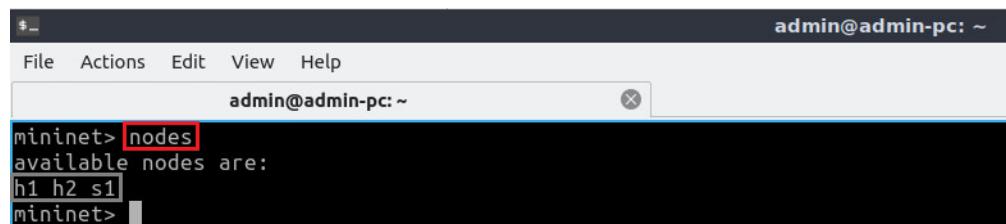
Some character-oriented interactive commands require
noecho:
  mininet> noecho h2 vi foo.py
However, starting up an xterm/gterm is generally better:
  mininet> xterm h2

mininet> 
```

Figure 6. Mininet's `help` command.

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

```
nodes
```



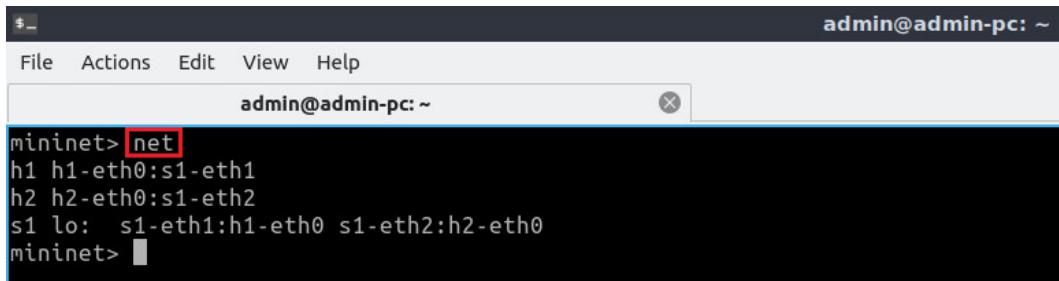
```
admin@admin-pc: ~
File Actions Edit View Help
admin@admin-pc: ~
mininet> nodes
available nodes are:
h1 h2 s1
mininet> 
```

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
```



```
admin@admin-pc: ~
File Actions Edit View Help
admin@admin-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.

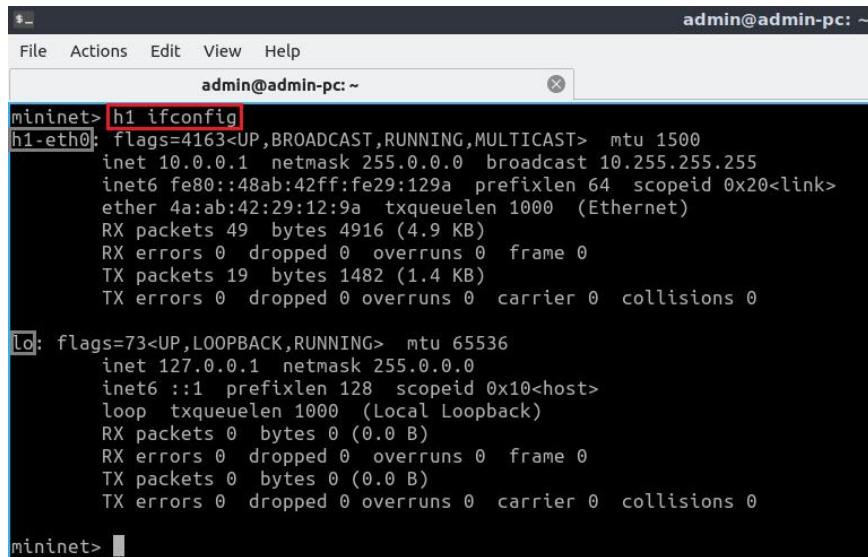
The output of this command shows that:

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
```



```
admin@admin-pc: ~
File Actions Edit View Help
admin@admin-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::48ab:42ff:fe29:129a prefixlen 64 scopeid 0x20<link>
          ether 4a:ab:42:29:12:9a txqueuelen 1000 (Ethernet)
            RX packets 49 bytes 4916 (4.9 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 19 bytes 1482 (1.4 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. 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
```

```
admin@admin-pc: ~
File Actions Edit View Help
admin@admin-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=1.15 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.073 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.072 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.074 ms
^C
--- 10.0.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 57ms
rtt min/avg/max/mdev = 0.072/0.342/1.150/0.466 ms
mininet>
```

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

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

```
exit
```

```
admin@admin-pc: ~
File Actions Edit View Help
admin@admin-pc: ~
mininet> exit
*** Stopping 0 controllers
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 922.271 seconds
admin@admin-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 MiniEdit⁵ to deploy the topology illustrated below. MiniEdit is a simple GUI network editor for Mininet.

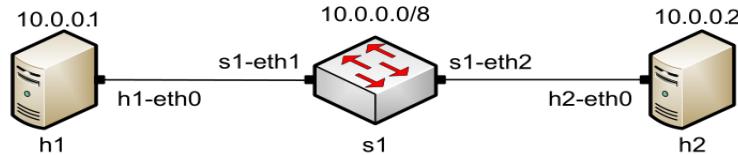


Figure 12. Lab topology.

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.

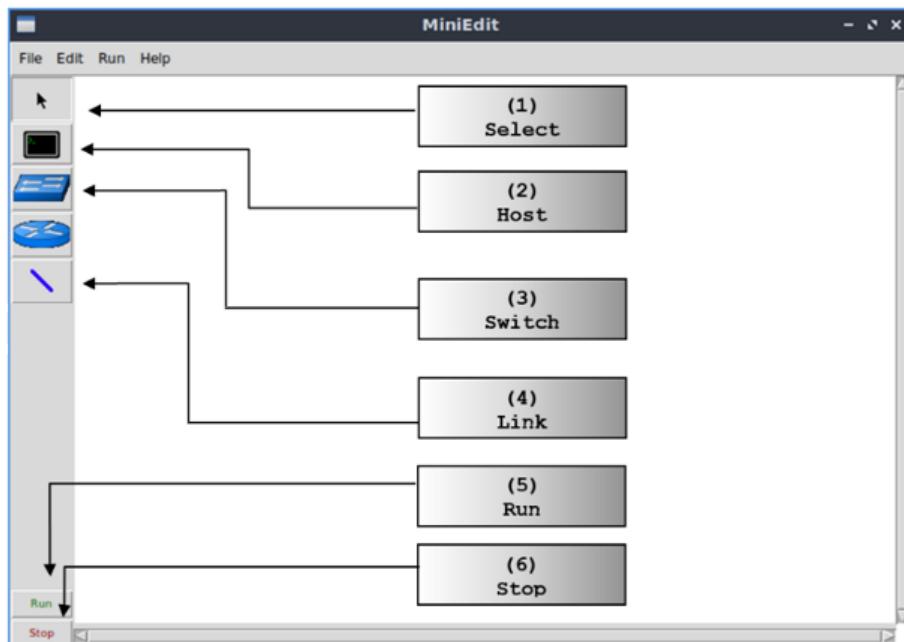


Figure 14. MiniEdit Graphical User Interface (GUI).

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.

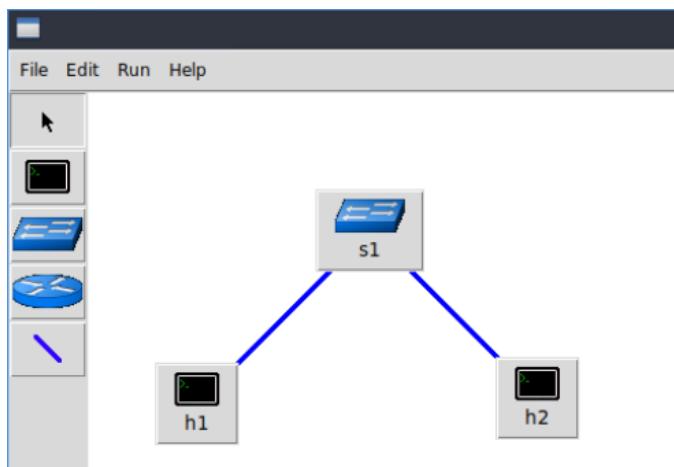


Figure 15. MiniEdit's topology.

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.

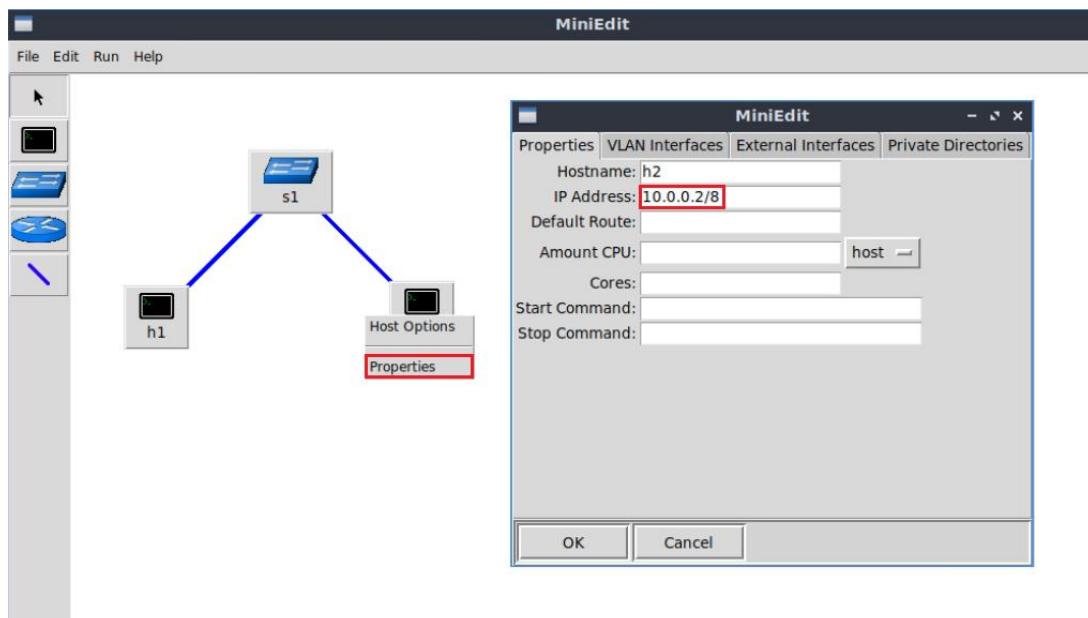


Figure 16. Configuration of a host's properties.

3.2 Test connectivity

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

Step 1. Click 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 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.

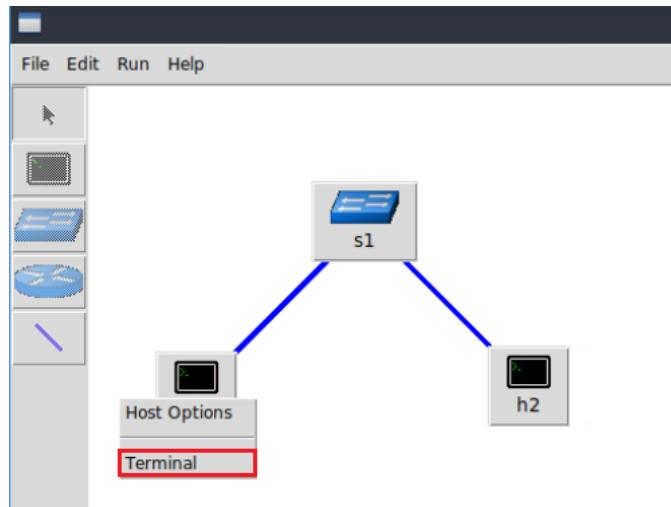


Figure 18. Opening a terminal on host h1.

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

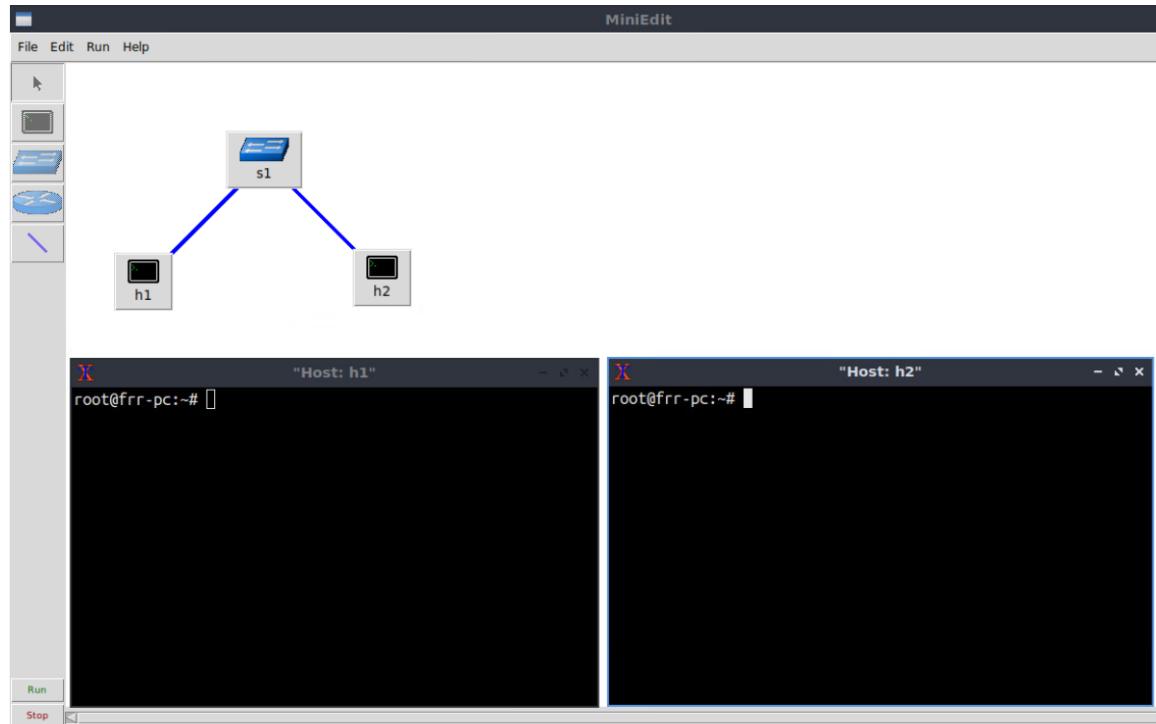
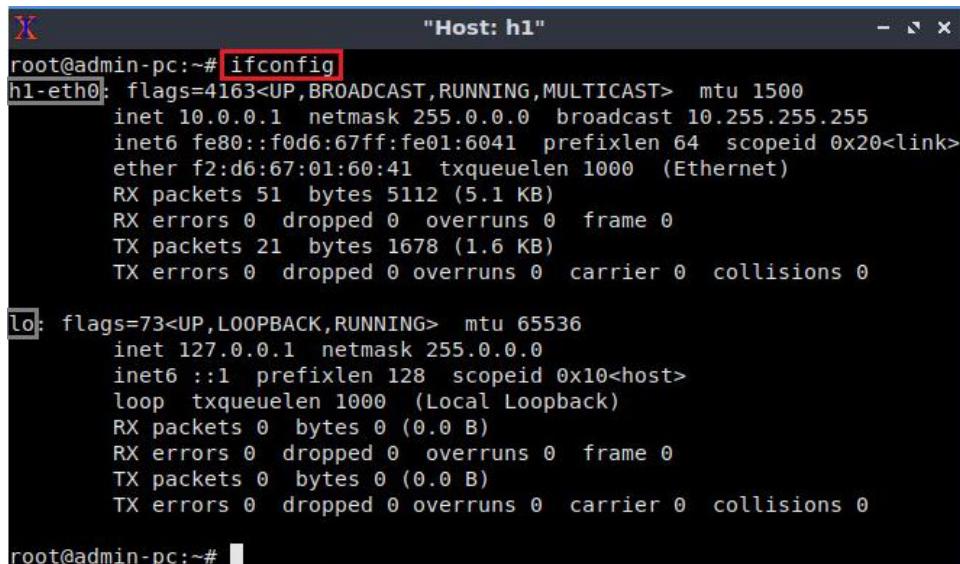


Figure 19. Terminals at host h1 and host h2.

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
```



```
"Host: h1"
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

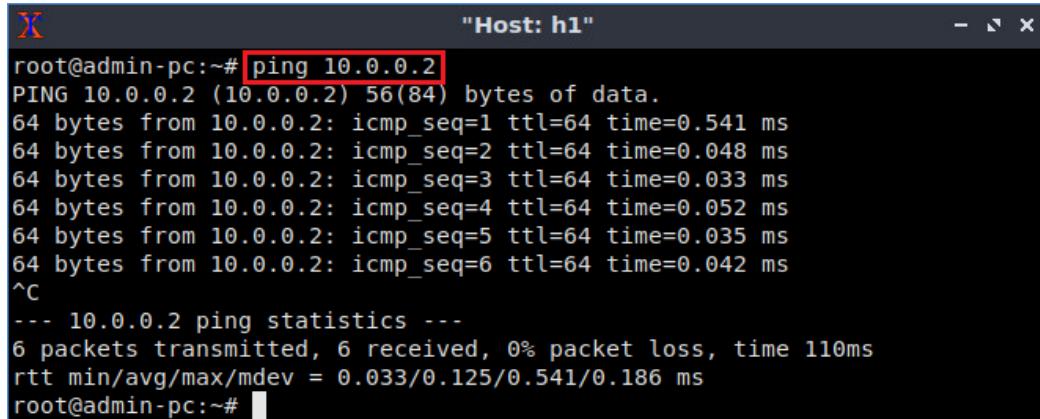
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
              inet6 ::1 prefixlen 128 scopeid 0x10<host>
                loop txqueuelen 1000 (Local Loopback)
                  RX packets 0 bytes 0 (0.0 B)
                  RX errors 0 dropped 0 overruns 0 frame 0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

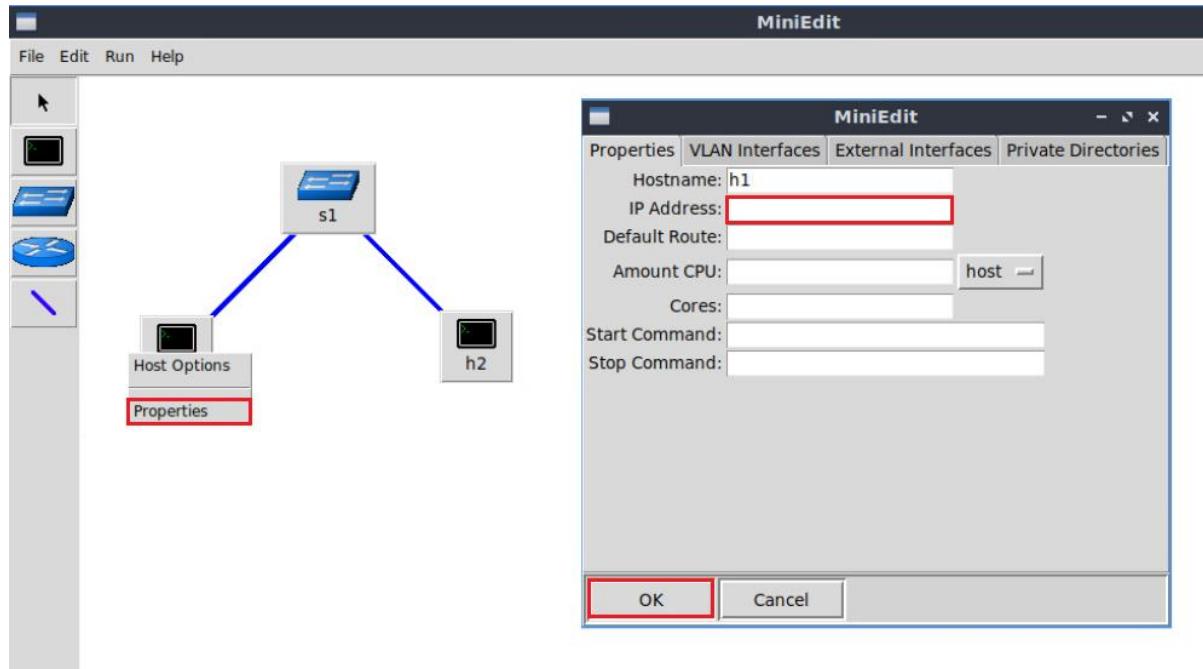


Figure 23. Host h1 properties.

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.

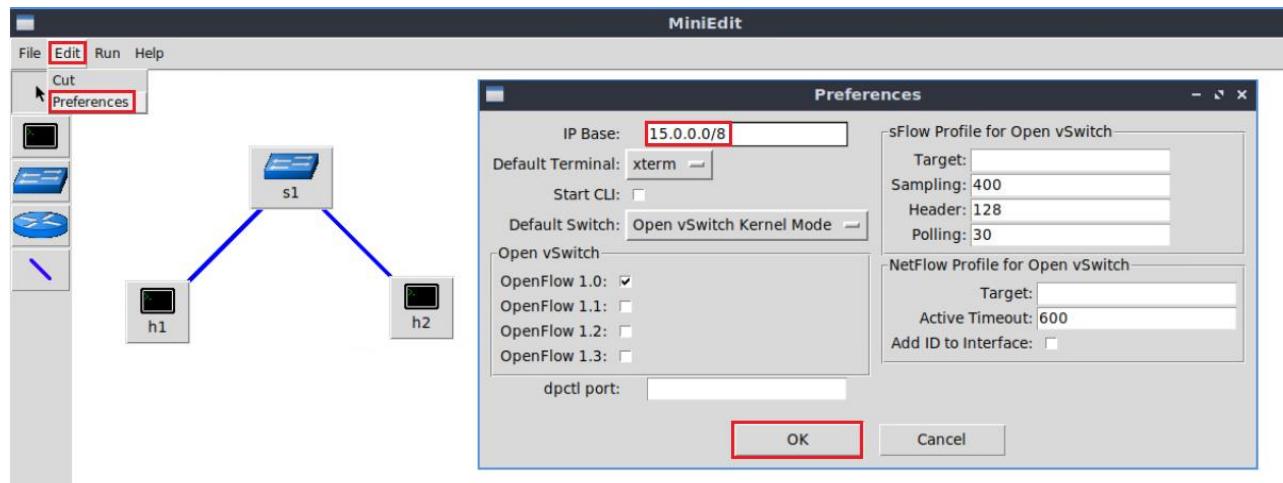


Figure 24. Modification of the IP Base (network address and prefix length).

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.

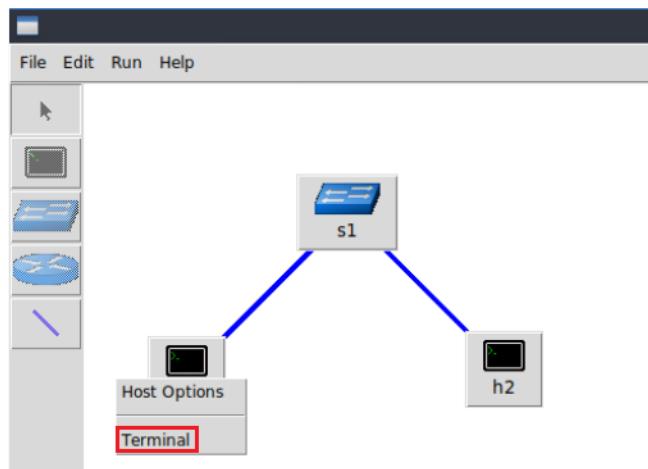
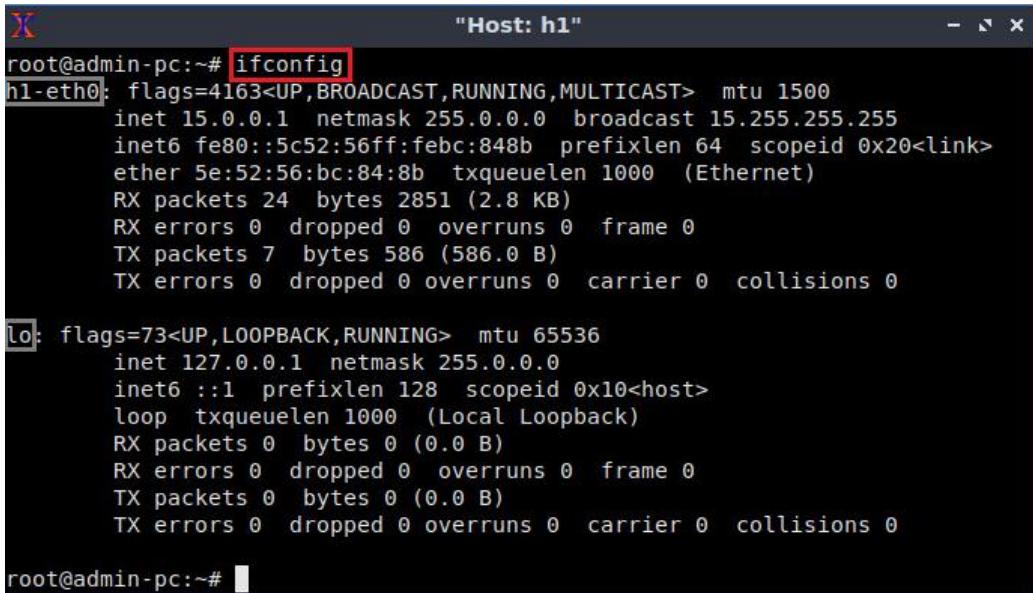


Figure 25. Opening a terminal on host h1.

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.

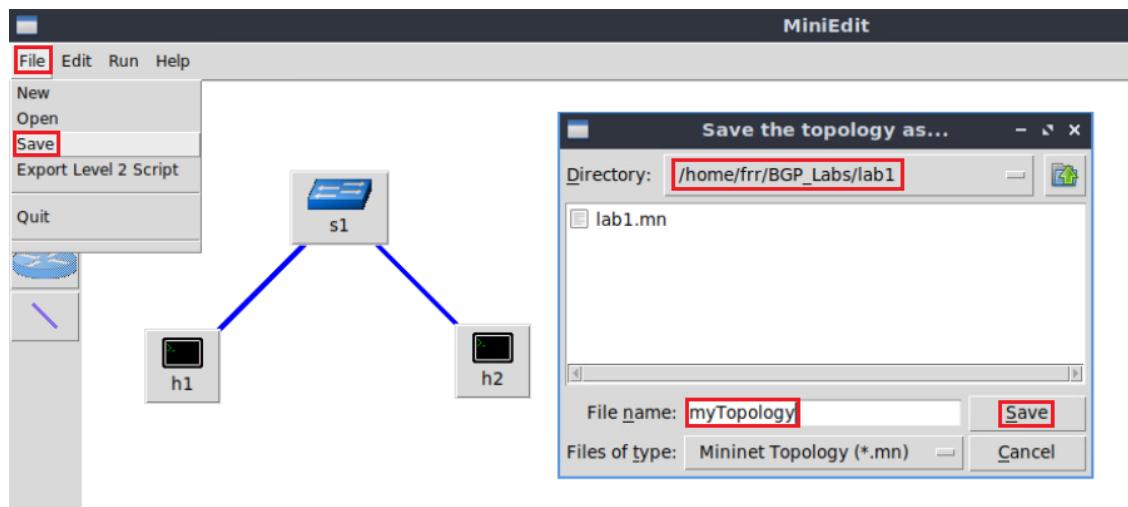


Figure 28. Saving the topology.

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.

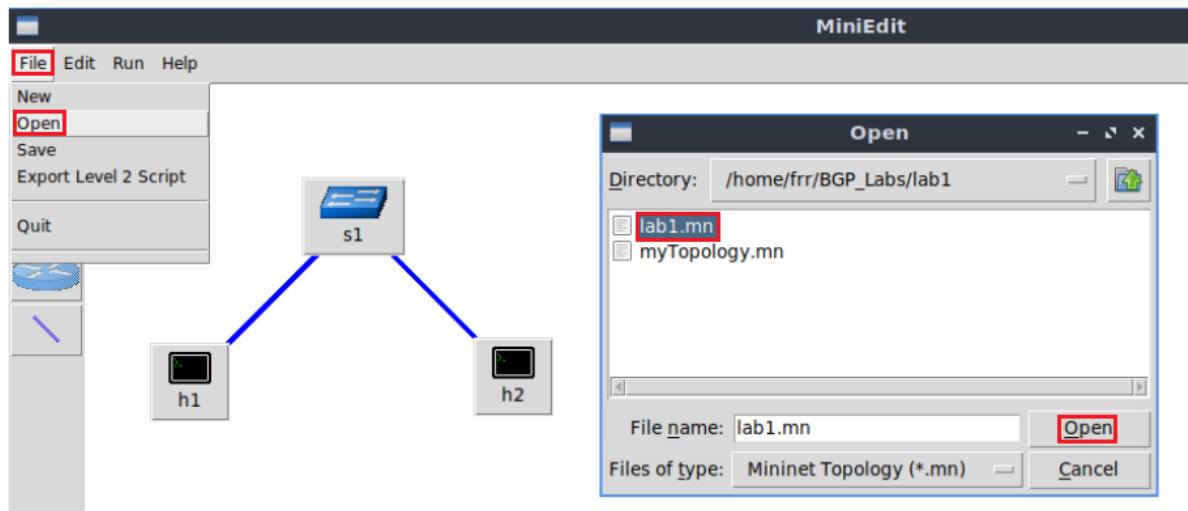


Figure 29. Opening a topology.

4 Configure router r1

In the previous section, you loaded a topology that consists in two networks directly connected to router r1. Consider 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.

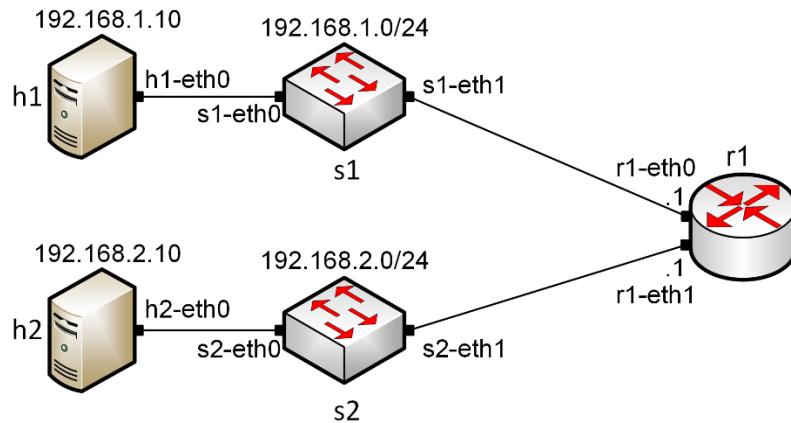


Figure 30. Topology.

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.

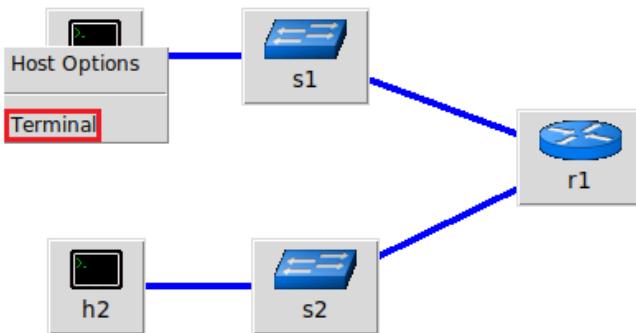


Figure 32. Opening a terminal on host h1.

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
```

```
"Host: h1"
root@frrr-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@frrr-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
```

```

X "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 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*.

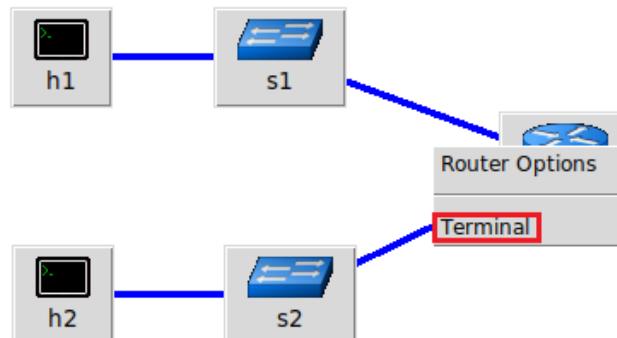
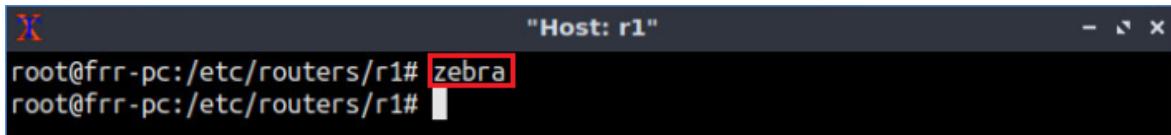


Figure 35. Opening a terminal on router r1.

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

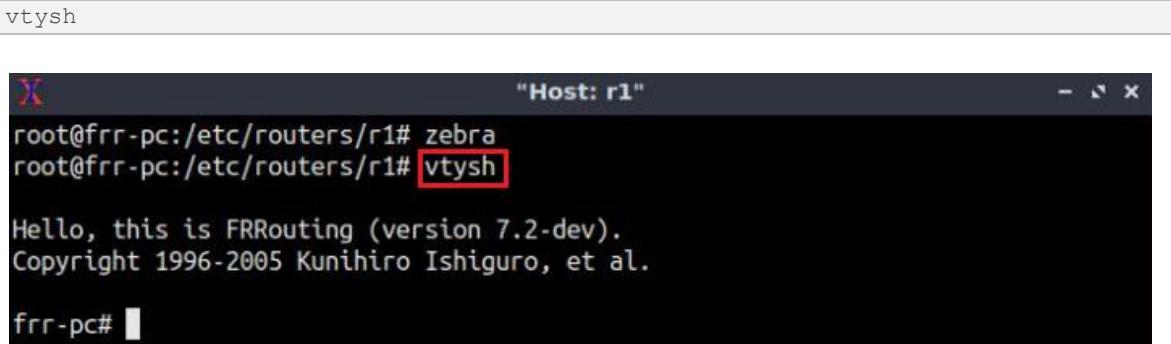
```
zebra
```



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

Figure 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:

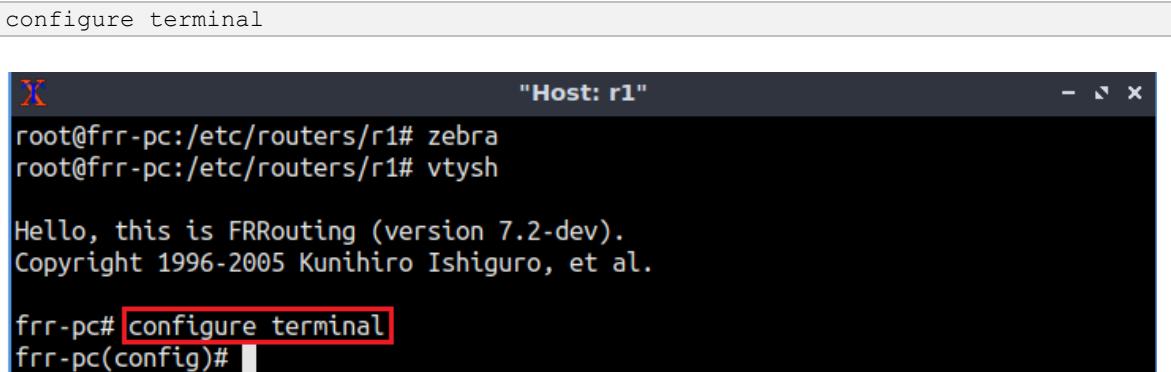


```
vtysh
"Host: r1"
root@frr-pc:/etc/routers/r1# zebra
root@frr-pc:/etc/routers/r1# vtysh
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# "
```

Figure 37. Starting vtysh on router r1.

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

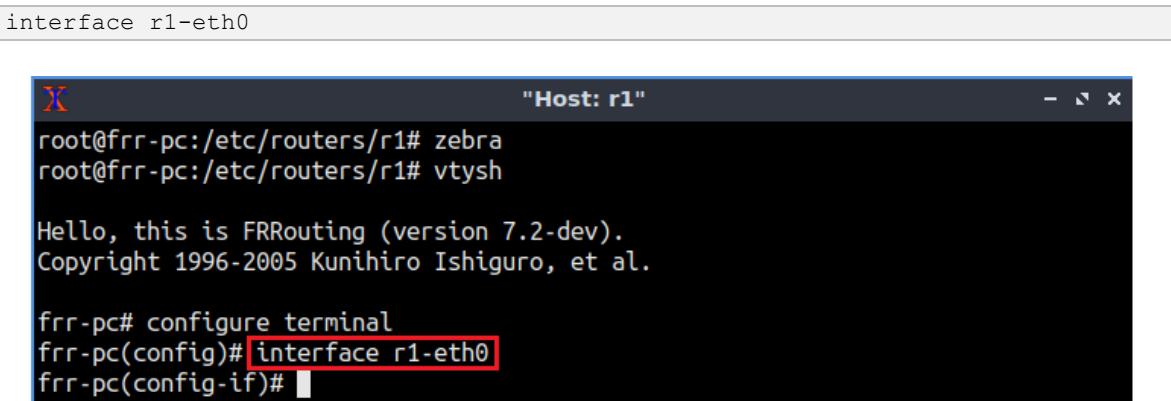


```
configure terminal
"Host: r1"
root@frr-pc:/etc/routers/r1# zebra
root@frr-pc:/etc/routers/r1# vtysh
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

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
```

The terminal window shows the following session:

```
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# 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
```

The terminal window shows the following session:

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

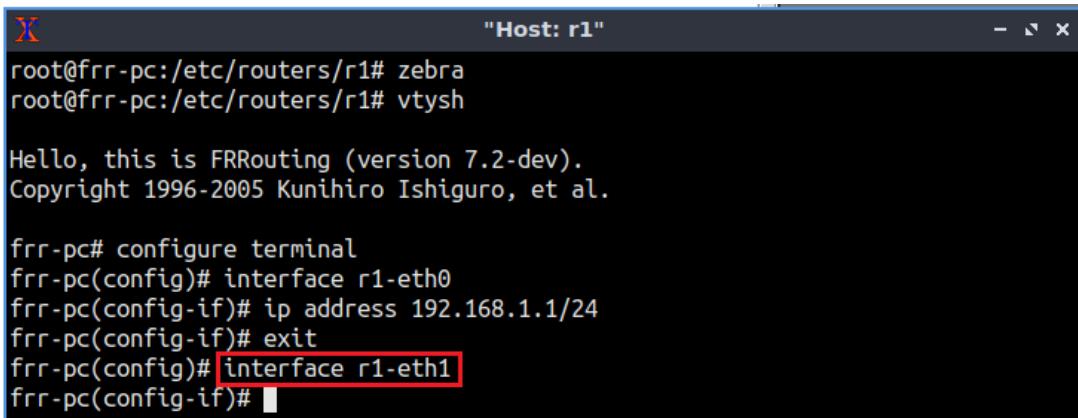
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# 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
```



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

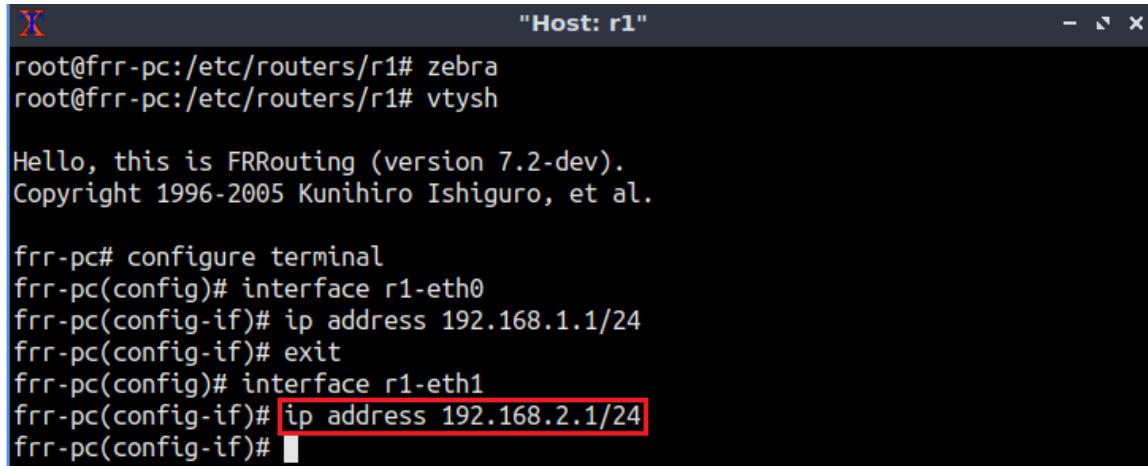
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

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



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

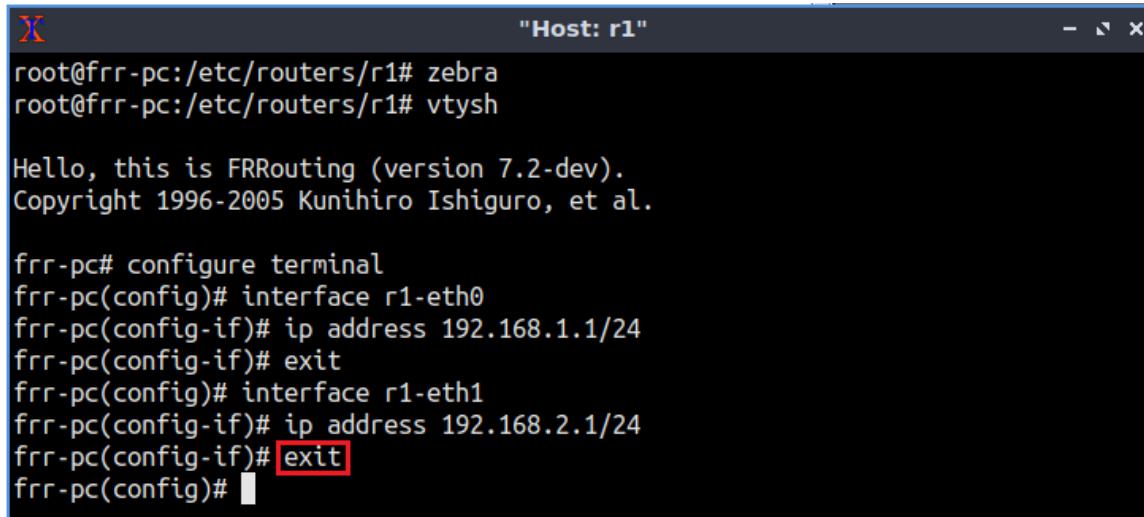
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frrr-pc# configure terminal
frrr-pc(config)# interface r1-eth0
frrr-pc(config-if)# ip address 192.168.1.1/24
frrr-pc(config-if)# exit
frrr-pc(config)# interface r1-eth1
frrr-pc(config-if)# ip address 192.168.2.1/24
frrr-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@frrr-pc:/etc/routers/r1# zebra
root@frrr-pc:/etc/routers/r1# vtysh

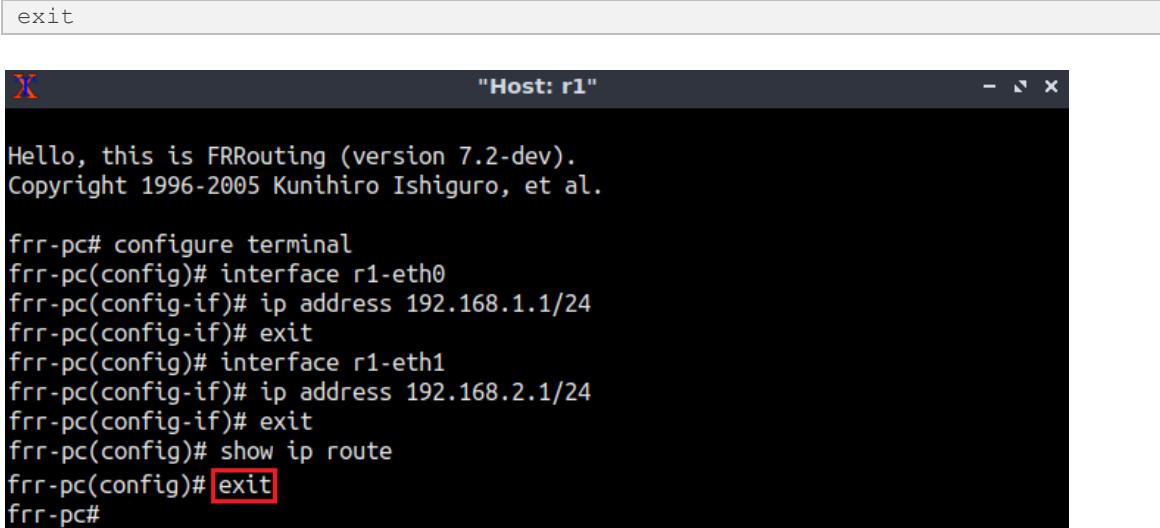
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

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

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

The terminal window shows the FRRouting configuration for router r1. It includes the copyright notice, configuration commands for interfaces r1-eth0 and r1-eth1, and the output of the 'show ip route' command. The output lists two direct connections: 192.168.1.0/24 via r1-eth0 and 192.168.2.0/24 via r1-eth1.

```
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
```

The terminal window shows the root user on host h1 performing a ping test to host h2 (IP 192.168.2.10). The test sends four ICMP echo requests and receives four replies from host h2. The statistics show 4 packets transmitted, 4 received, 0% packet loss, and an average round-trip time of 53ms.

```
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 Linux terminal.

References

1. Mininet walkthrough. [Online]. Available: <http://Mininet.org>.

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



BORDER GATEWAY PROTOCOL

Lab 2: Introduction to Free Range Routing (FRR)

Document Version: **02-18-2020**



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 encompasses 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, students 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 machine containing Mininet.

Table 1. Credentials to access Client1 machine.

Device	Account	Password
Client1	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 limitation such as the cost of the

hardware, 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 that can be deployed.

For example, operators could be interested in collecting some information about the behavior of routing devices, process them, and make them available. Therefore, in order 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 on developing routing protocols by extending an existing one without writing a complete implementation from scratch.

FRR suite¹ is a package of Unix/Linux software that implements common network routing protocols, such as Routing Information Protocol⁴ (RIP), Open Shortest Path First⁵ (OSPF), Border Gateway Protocol⁶ (BGP) and Intermediate System to Intermediate System IS-IS⁷. The package also includes a routing information management process, to act as intermediary between the various 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. In addition, libraries and kernel daemon provide a framework to facilitate the development of new routing protocol daemons. A wide range of functionalities can be attained by combining other software packages to allow the integration into a single device as well as enabling innovative solutions to networking problems.

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

1.1 FRR architecture

FRR takes a different approach compared to traditional routing software which, consists of a single process program that provides all the routing protocol functionalities. FRR is composed by a suite of daemons that work together to build a routing table. Each routing protocol is implemented in its own daemon. These daemons exchange information through another daemon called zebra, which is responsible for encompassing routing decisions and managing the dataplane.

Since all the protocols are running independently, this architecture provides high resiliency, that 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 the modularity makes it easy to implement new protocols and append them to the suite¹. 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 allow the management of 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 e.g., the BGP daemon is implemented by the *bgpd* process, the RIP daemon is implemented by the *ripd* process and so on. 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 *vtysh* allows these processes to be monitored and configured. The *vtysh* 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)². Usually, the kernel manages packet forwarding therefore, 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)² 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 FIB³. 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.

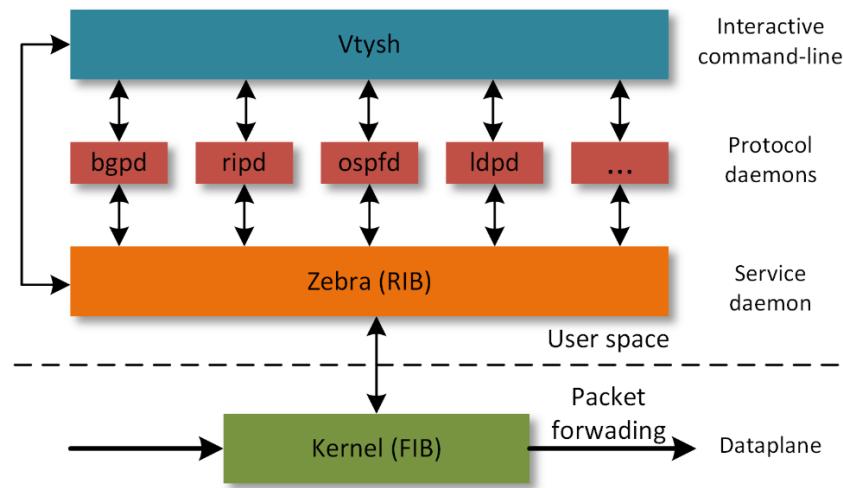


Figure 1. FRR architecture.

1.2 FRR and Mininet integration

Mininet is a network emulator which runs collection of end-hosts, switches, routers and links on a single Linux kernel⁵. Mininet provides network emulation, 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. Hence, the set of commands provided by FRR are inherited and can be run 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 an emulated 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 specify the network, a switch that defines a Local Area Network (LAN) and lastly, a host aimed to test end-to-end connectivity

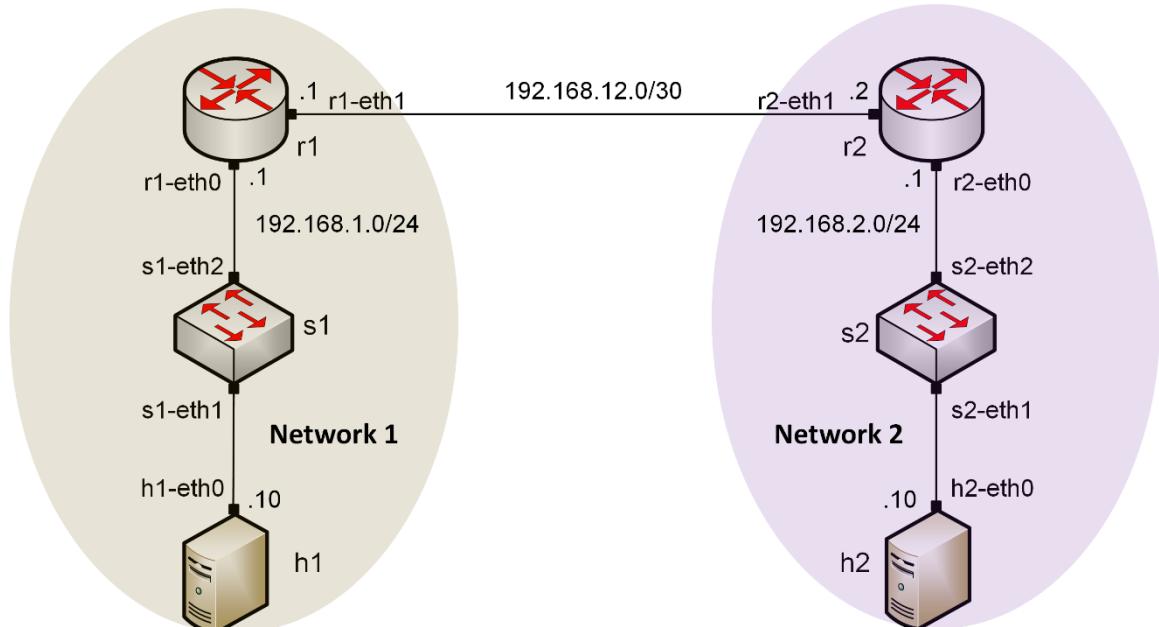


Figure 2. Lab topology.

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 MiniEdit⁹ 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 such as: configuring network elements (i.e IP addresses, default gateway), save the topology and export a layer 2 model.

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 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/BGP_Labs/lab2* and click on *Open*.

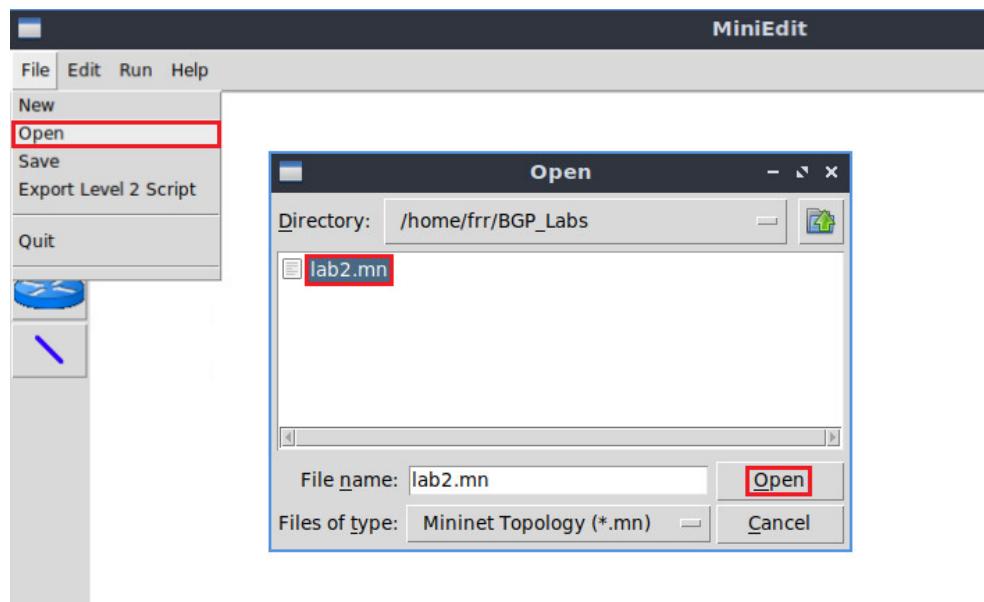


Figure 4. MiniEdit's open dialog.

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

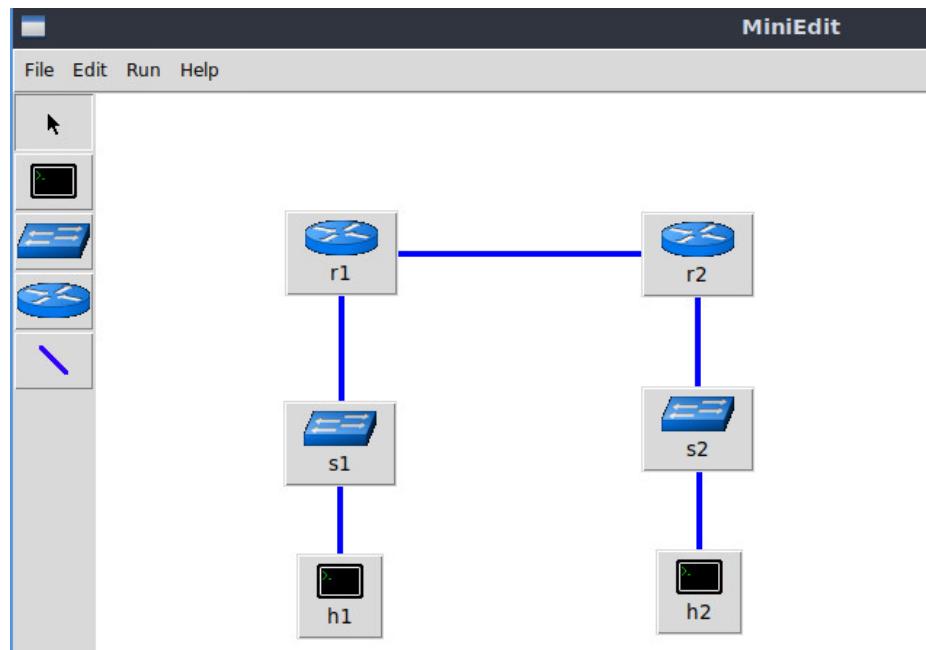


Figure 5. Mininet's topology.

2.3 Load the configuration file

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

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

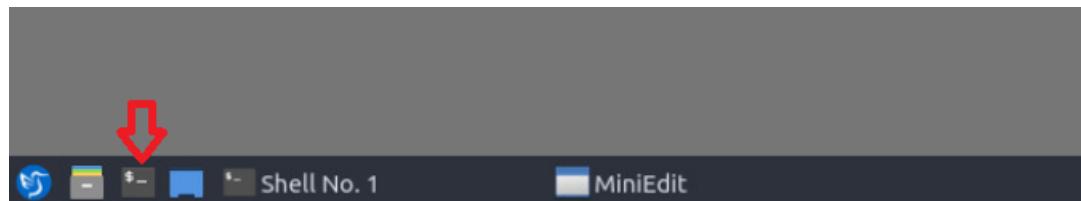
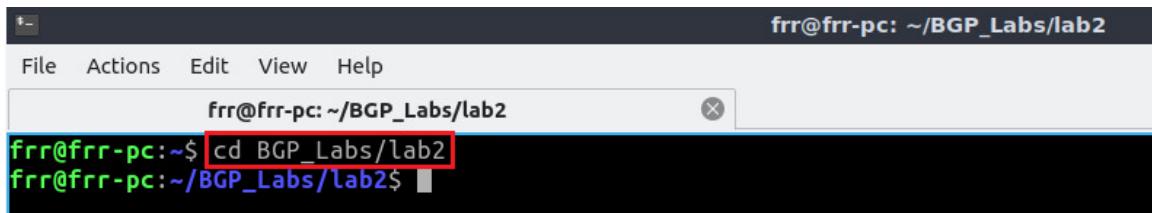


Figure 6. Opening Linux terminal.

Step 2. Click on the Linux's terminal and navigate into *BGP_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 BGP_Labs/lab2
```

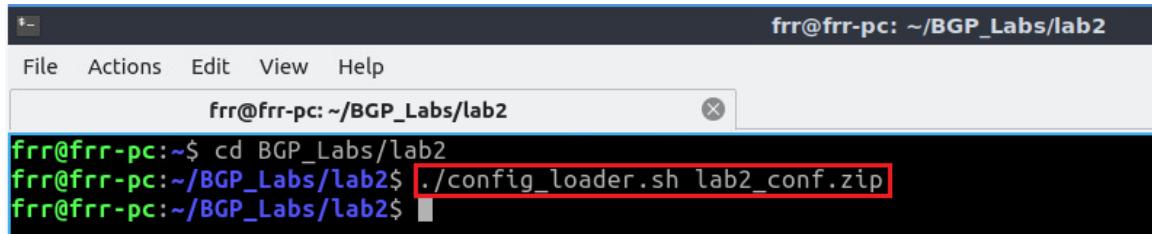


```
frr@frr-pc: ~$ cd BGP_Labs/lab2
```

Figure 7. Entering the *BGP_Labs/lab2* directory.

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

```
./config_loader.sh lab2_config.zip
```

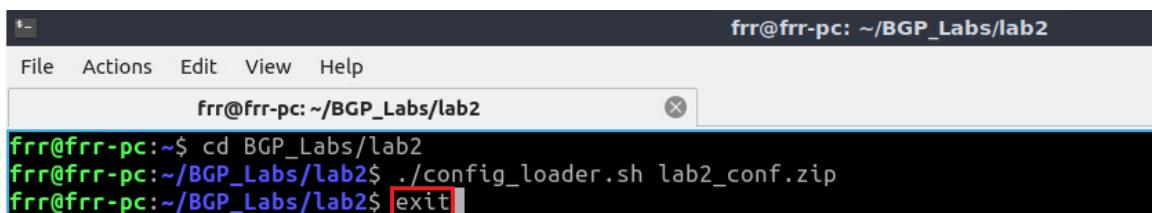


```
frr@frr-pc: ~$ ./config_loader.sh lab2_config.zip
```

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: ~$ 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 lower left-hand side.



Figure 10. Starting the emulation.

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

```
links
```

Shell No. 1

File Actions Edit View Help

Shell No. 1

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

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 the IP addresses assigned to each host 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.

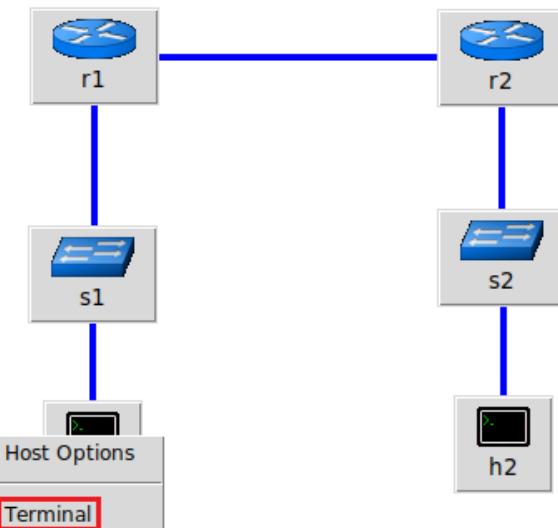


Figure 12. Opening a terminal on host h1.

Step 2. In 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.

```
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. On 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
        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. 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.

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

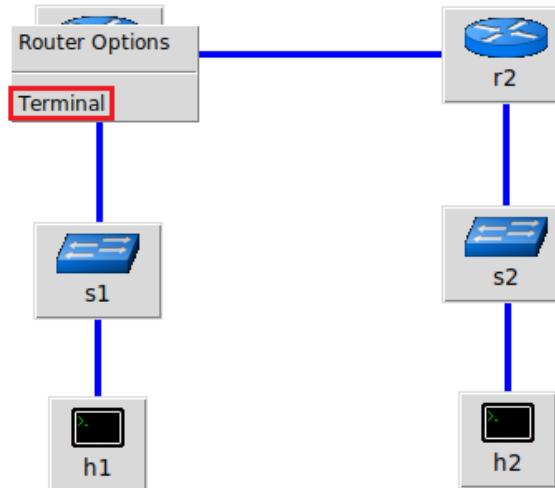
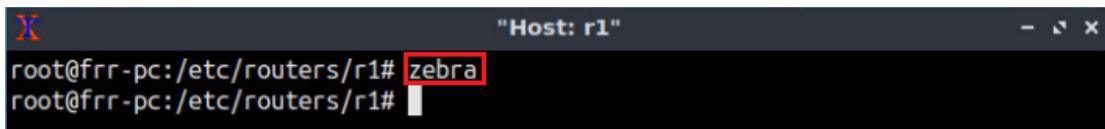


Figure 15. Opening a terminal on router r1.

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

```
zebra
```

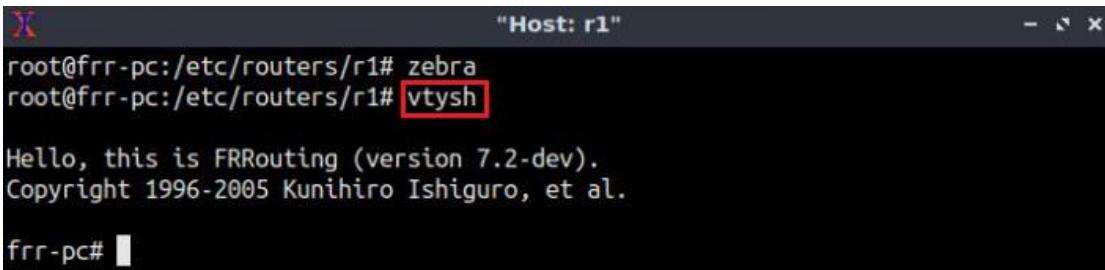


```
"Host: r1"
root@frr-pc:/etc/routers/r1# zebra
root@frr-pc:/etc/routers/r1#
```

Figure 16. Starting zebra daemon.

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

```
vtysh
```



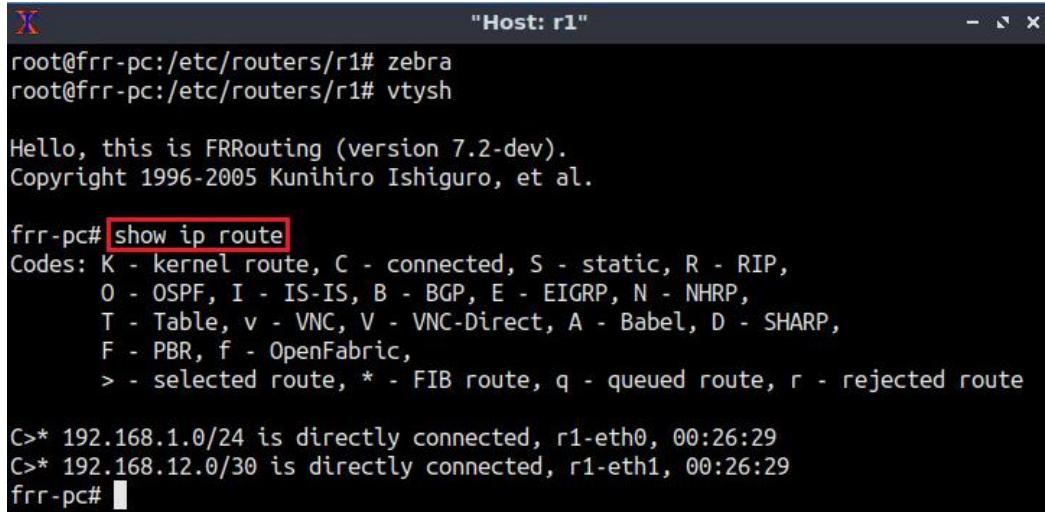
```
"Host: r1"
root@frr-pc:/etc/routers/r1# zebra
root@frr-pc:/etc/routers/r1# vtysh
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

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

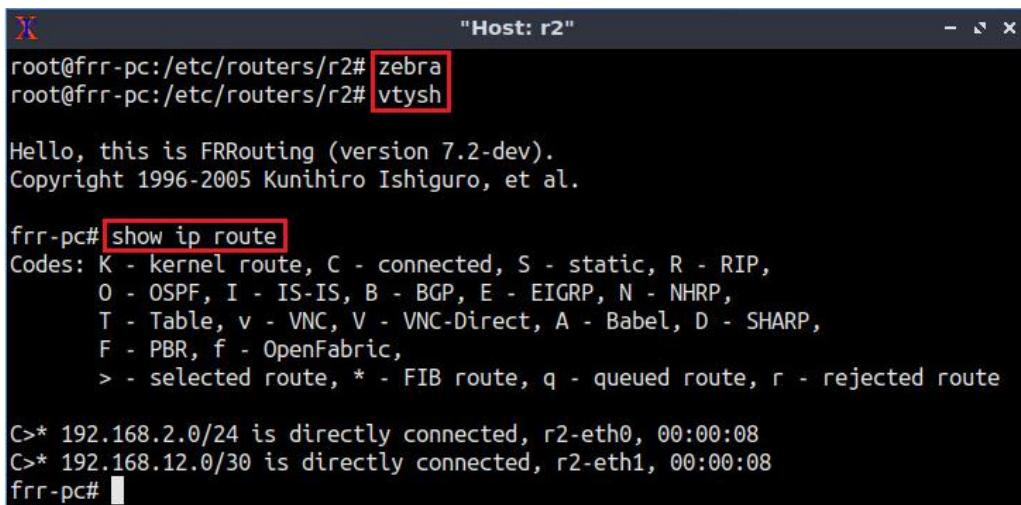
C>* 192.168.1.0/24 is directly connected, r1-eth0, 00:26:29
C>* 192.168.12.0/30 is directly connected, r1-eth1, 00:26:29
frr-pc#
```

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.



The terminal window shows the configuration of router r2. It starts with the command `zebra`, followed by `vtysh`. The output of `vtysh` includes the FRRouting version (7.2-dev) and copyright information. Then, the command `show ip route` is run, displaying the routing table. The table shows two directly connected routes: `C>* 192.168.2.0/24` via interface `r2-eth0` and `C>* 192.168.12.0/30` via interface `r2-eth1`.

```
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,
      0 - 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#
```

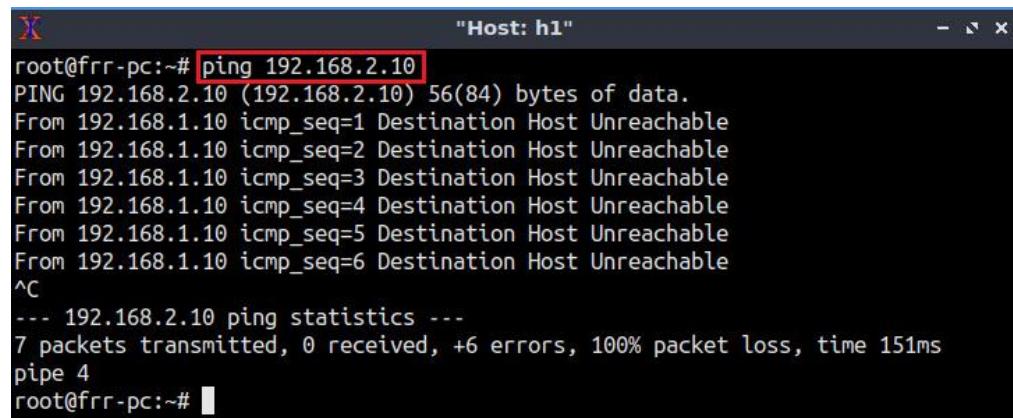
Figure 19. Displaying routing table of router r2.

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 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
```



The terminal window shows a ping test from host h1 to host h2. The command `ping 192.168.2.10` is issued. The output shows multiple ICMP echo requests being sent to the destination, but all are marked as "Destination Host Unreachable". After several attempts, the user presses `ctrl+c` to stop the test. The final statistics show 7 packets transmitted, 0 received, and 100% packet loss.

```
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 in order 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 out

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 IP address of the next hop. Finally, you will verify the configuration.

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

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

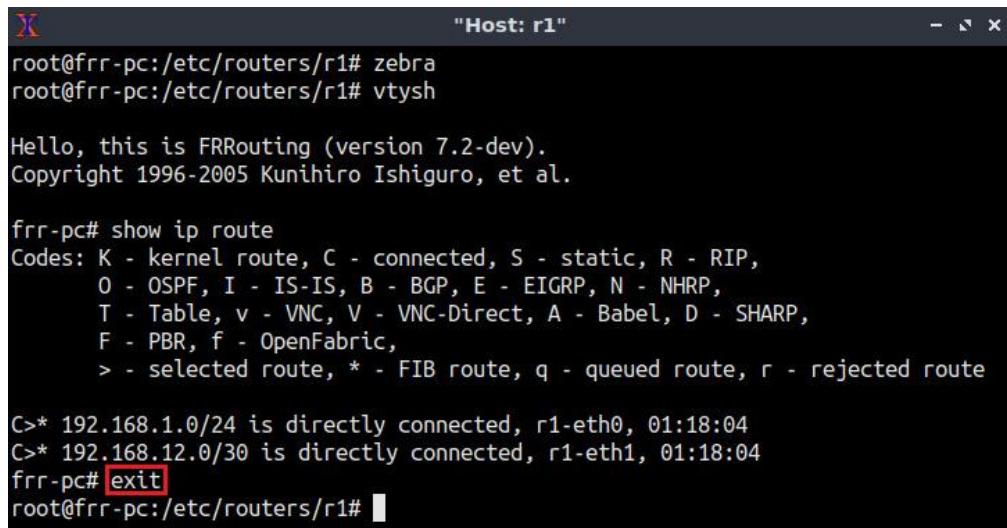
- `ip route`: is used to create or modify routing tables.
- `NETWORK`: specifies the destination network.
- `GATEWAY`: determines the next hop IP address.

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
```



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

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

frrr-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
frrr-pc# exit
root@frrr-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,
      0 - 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 a connectivity between the networks 192.168.1.0/24 and 192.168.2.0/24.

Step 1. In order to enter to router r3 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,
      0 - 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,
      0 - 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 is reachable via the IP address 192.168.12.2. 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 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.

```
traceroute 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:~# 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 `traceroute` 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. P. Jakma, D. Lamparter. “Introduction to the quagga routing suite,” 2014, *IEEE Network* 28.
3. K. Ishiguro, “Gnu zebra.”. [Online]. Available: <http://www.zebra.org> (2002).
4. G. Malkin, “RIP Version 2,” RFC 2453 updated by RFC 4822, 1998. [Online]. Available: <http://www.ietf.org/rfc/rfc2453.txt>.
5. Mininet walkthrough. [Online]. Available: <http://Mininet.org>.
6. G. Malkin, R. Minnear, “RIPng for IPv6,” RFC 2080, 1997. [Online]. Available: <http://www.ietf.org/rfc/rfc2080.txt>.
7. Y. Rekhter, T. Li, S. Hares, “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>.
8. D. Oran, “OSI IS-IS intra-domain routing protocol,” RFC 1142, 1990. [Online]. Available: <http://www.ietf.org/rfc/rfc1142.txt>.
9. B. Lantz, G. Gee, “MiniEdit: a simple network editor for Mininet,” 2013. [Online]. Available: <https://github.com/Mininet/Mininet/blob/master/examples>.



BORDER GATEWAY PROTOCOL

Lab 3: Introduction to BGP

Document Version: **02-18-2020**



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 BGP	3
1.1 Classification of dynamic routing protocols.....	3
1.2 BGP overview	4
2 Lab topology.....	5
2.1 Lab settings.....	6
2.2 Open the topology and load the configuration	6
2.3 Load zebra daemon and verify configuration	9
3 Configure BGP on the routers.....	12
3.1 Add BGP neighbors on the routers	13
3.2 Advertise local networks on the routers.....	17
4 Verify connections	20
References	22

Overview

This lab presents Border Gateway Protocol (BGP) and describes the concept of Internal BGP (IBGP) and External BGP (EBGP). Furthermore, EBGP will be configured and verified between two Autonomous Systems (ASes) that are required to exchange routes.

Objectives

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

1. Explain the concept of BGP.
2. Explain the difference between IBGP and EBGP.
3. Configure and verify EBGP between two ASes.

Lab settings

The information in Table 1 provides the credentials to access Client1 machine.

Table 1. Credentials to access Client1 machine.

Device	Account	Password
Client1	admin	password

Lab roadmap

This lab is organized as follows:

1. Section 1: Introduction to BGP.
2. Section 2: Lab topology.
3. Section 3: Configure BGP on all routers.
4. Section 4: Verify connections.

1 **Introduction to BGP**

1.1 **Classification of dynamic routing protocols**

The Internet can be viewed as a collection of networks or ASes that are interconnected. An AS refers to a group of connected networks under the control of a single administrative entity or domain. Figure 1 illustrates dynamic routing protocols which can be divided in two categories, Interior Gateway Protocol (IGP) and Exterior Gateway Protocol (EGP)¹.

IGP, also called intradomain routing protocol, is used within an AS. The existing IGPs differ by algorithm; however, they all aim to move packets as efficiently as possible from the source to the destination according to different metrics. Some of the extensively used IGPs are listed below¹:

1. Routing Information Protocol (RIP): distance-vector protocol that uses hop count as a cost metric; that is, each link has a cost of 1.
2. Intermediate-System to Intermediate-System (IS-IS) and Open Shortest Path First (OSPF): link-state protocols that use flooding of link-state information and the Dijkstra least-cost path algorithm.

OSPF is an enhancement of the IS-IS protocol, thus, the two protocols are more alike than different. The most important difference is that IS-IS can carry information about multiple network layer protocols at the same time, whereas OSPF does not have this feature. This is an advantage for OSPF in large multiprotocol environments¹.

EGP, also called interdomain routing protocol, is used for routing between different ASes, where it is a scalable protocol used on the Internet to connect different ASes. The protocol that is used in the Internet is called BGP and it differs from IGPs by allowing many kinds of routing policies to be enforced between the ASes, rather than just finding the best route. Routing policies involve political, security, or economic considerations, such as preventing commercial traffic on an educational network¹.

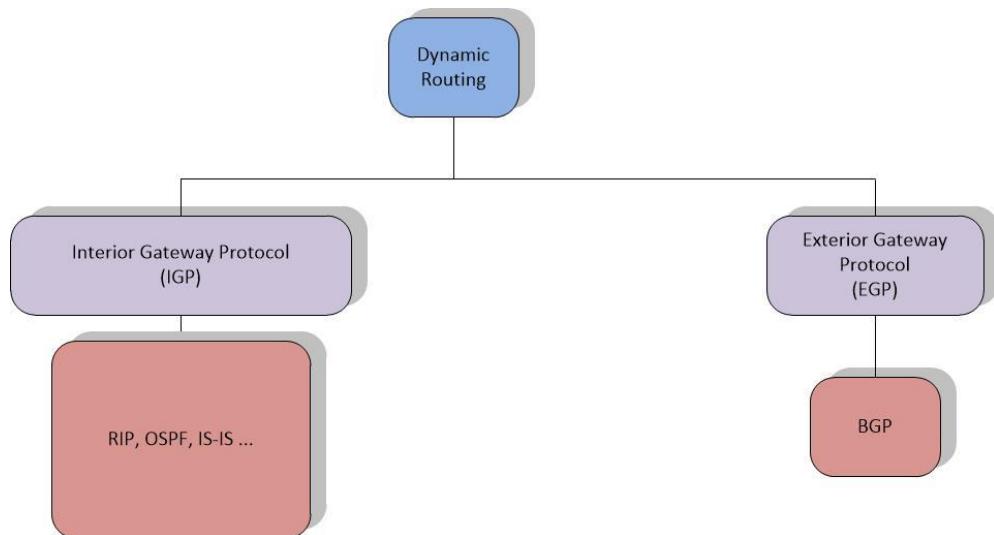


Figure 1. Classification of dynamic routing protocols.

1.2 BGP overview

BGP is an exterior gateway protocol designed to exchange routing and reachability information among ASes on the Internet. BGP is relevant to network administrators of large organizations which connect to one or more Internet Service Providers (ISPs), as well as to ISPs who connect to other network providers. In terms of BGP, an AS is referred to

as a routing domain, where all networked systems operate common routing protocols and are under the control of a single administration¹.

BGP is a form of distance vector protocol. It requires each router to maintain a table, which stores the distance and the output interface (i.e., vector) to remote networks. BGP makes routing decisions based on paths, network policies, or rule set configured by a network administrator and is involved in making core routing decisions¹.

Two routers that establish a BGP connection are referred to as BGP peers or neighbors. BGP sessions run over Transmission Control Protocol (TCP). If a BGP session is established between two neighbors in different ASes, the session is referred to as an EBGP session. If the session is established between two neighbors in the same AS, the session is referred to as IBGP¹. Figure 2 shows a network running BGP protocol. Routers that exchange information within the same AS use IBGP, while routers that exchange information between different ASes use EBGP.

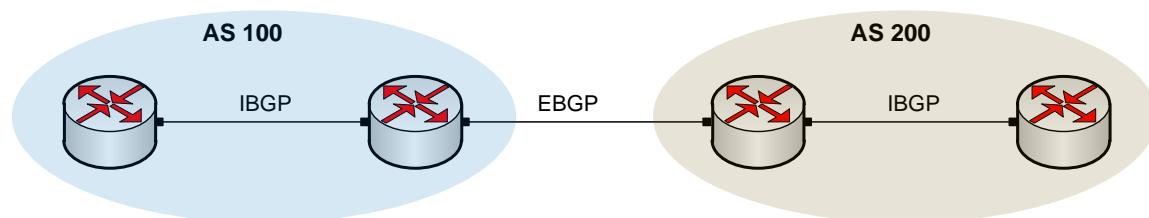


Figure 2. Routers that exchange information within the same AS use IBGP, while routers that exchange information between different ASes use EBGP.

2 Lab topology

Consider Figure 3. The topology consists of two networks, Network 1 and Network 2. The ASNs assigned to routers r1 and r2 are 100 and 200 respectively. Routers r1 and r2 exchange routing information via EBGP.

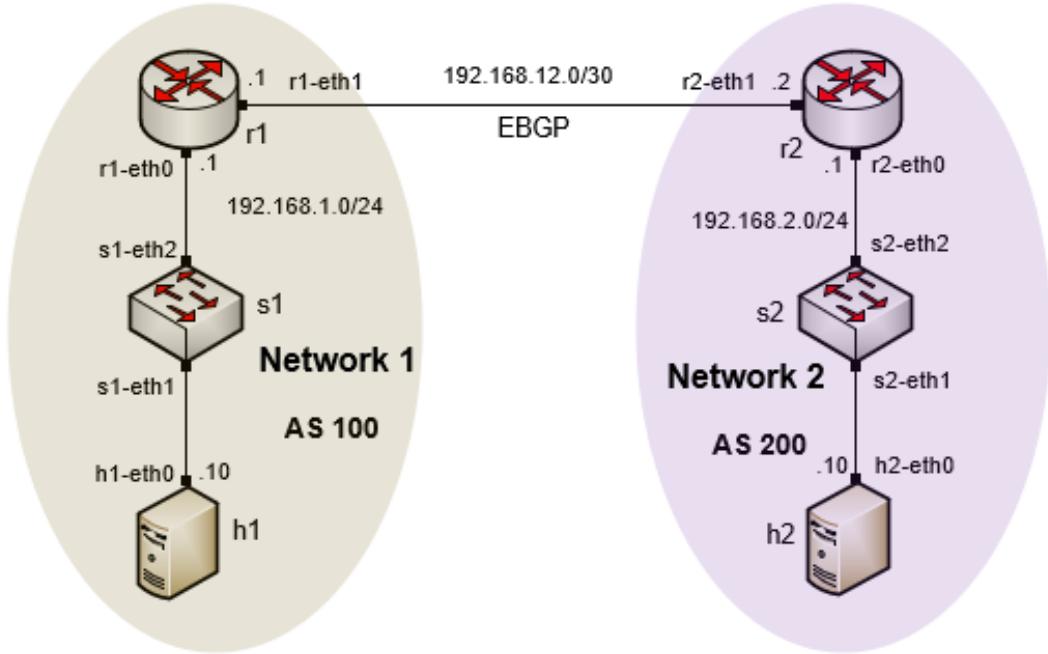


Figure 3. Lab topology.

2.1 Lab settings

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

Table 2. Topology information.

Device	Interface	IPV4 Address	Subnet	Default gateway
Router r1	r1-eth0	192.168.1.1	/24	N/A
	r1-eth1	192.168.12.1	/30	N/A
Router r2	r2-eth0	192.168.2.1	/24	N/A
	r2-eth1	192.168.12.2	/30	N/A
Host h1	h1-eth0	192.168.1.10	/24	192.168.1.1
Host h2	h2-eth0	192.168.2.10	/24	192.168.2.1

2.2 Open the topology and load the configuration

Step 1. Start by launching Miniedit by clicking on Desktop's shortcut. When prompted for a password, type `password`.



Figure 4. MiniEdit shortcut.

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

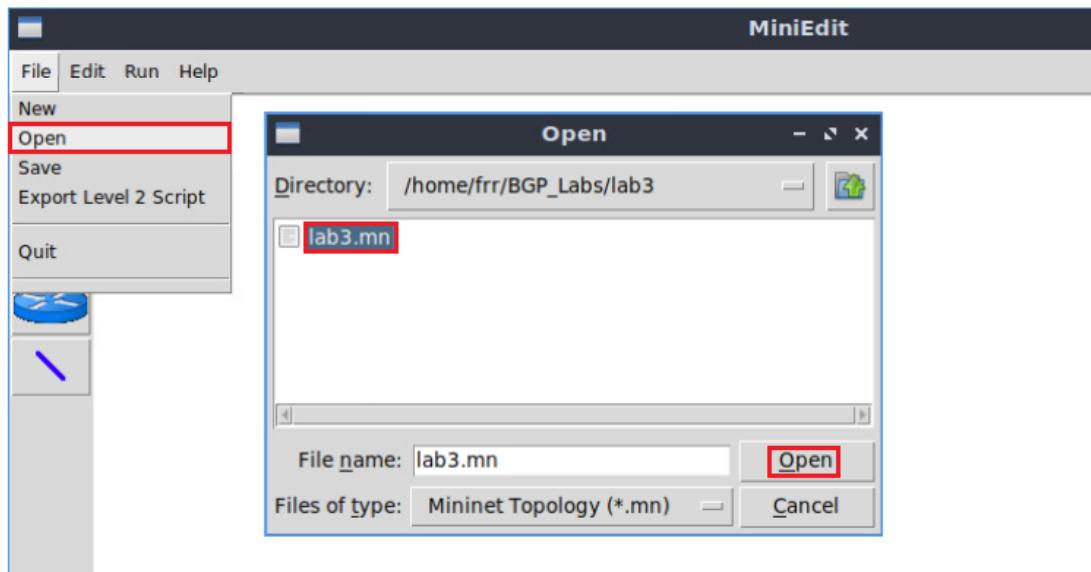


Figure 5. MiniEdit's open dialog.

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

Step 3. Open the Linux terminal.

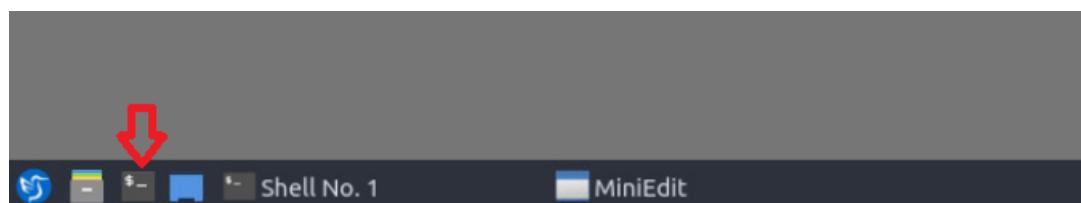
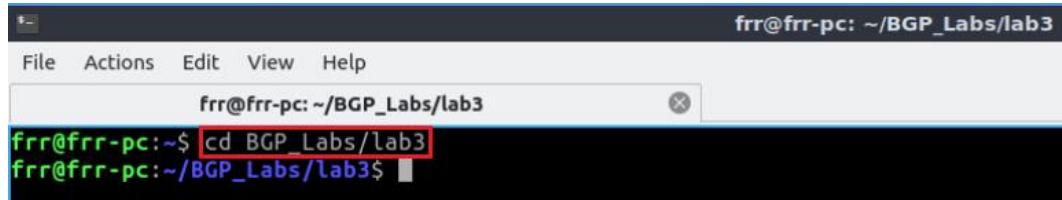


Figure 6. Opening Linux terminal.

Step 4. Click on the Linux's terminal and navigate into *BGP_Labs/lab3* directory. This folder contains a configuration file and the script responsible for loading the

configuration. The configuration file will assign the IP addresses to router r1 and router r2 interfaces. To proceed, type the command shown below. The `cd` command is short for change directory followed by an argument that specifies the destination directory.

```
cd BGP_Labs/lab3
```

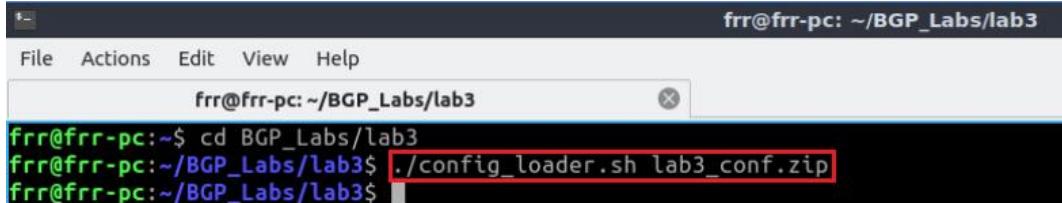


A screenshot of a Linux terminal window titled "frr@frr-pc: ~/BGP_Labs/lab3". The window has a menu bar with File, Actions, Edit, View, and Help. Below the menu is a toolbar with icons for File, Actions, Edit, View, and Help. The terminal prompt is "frr@frr-pc:~\$". The user types "cd BGP_Labs/lab3" and presses Enter. The terminal shows the command in red and the resulting path "frr@frr-pc:~/BGP_Labs/lab3\$".

Figure 7. Entering the *BGP_Labs/lab3* directory.

Step 5. To execute the shell script, type the following command. The argument of the program corresponds to the configuration zip file that will be loaded in all the routers in the topology.

```
./config_loader.sh lab3_conf.zip
```

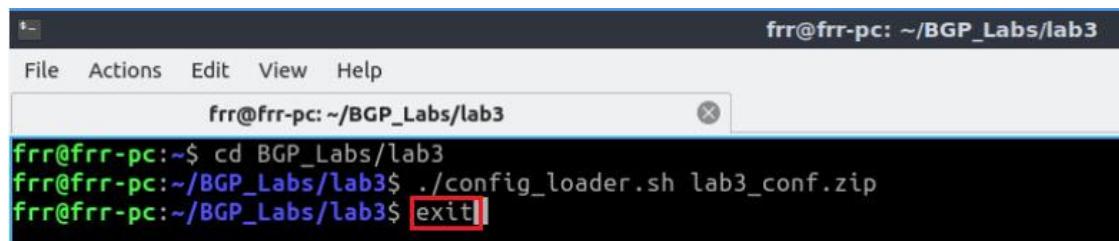


A screenshot of a Linux terminal window titled "frr@frr-pc: ~/BGP_Labs/lab3". The window has a menu bar with File, Actions, Edit, View, and Help. Below the menu is a toolbar with icons for File, Actions, Edit, View, and Help. The terminal prompt is "frr@frr-pc:~\$". The user types "./config_loader.sh lab3_conf.zip" and presses Enter. The terminal shows the command in red and the resulting path "frr@frr-pc:~/BGP_Labs/lab3\$".

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

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

```
exit
```



A screenshot of a Linux terminal window titled "frr@frr-pc: ~/BGP_Labs/lab3". The window has a menu bar with File, Actions, Edit, View, and Help. Below the menu is a toolbar with icons for File, Actions, Edit, View, and Help. The terminal prompt is "frr@frr-pc:~\$". The user types "exit" and presses Enter. The terminal shows the command in red and the resulting path "frr@frr-pc:~/BGP_Labs/lab3\$".

Figure 9. Exiting from the terminal.

Step 8. At this point hosts h1 and h2 interfaces are configured. To proceed with the emulation, click on the *Run* button located in lower left-hand side.



Figure 10. Starting the emulation.

Step 9. Click on Mininet's terminal, i.e., the one launched when MiniEdit was started.

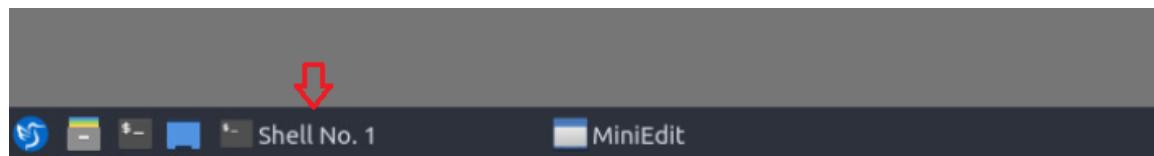


Figure 11. Opening Mininet's terminal.

Step 10. Issue the following command to display the interface names and connections.

A screenshot of the 'Shell No. 1' terminal window. The command 'links' is entered at the mininet> prompt. The output shows five network links: h1-eth0 connected to s1-eth1, s1-eth2 connected to r1-eth0, h2-eth0 connected to s2-eth1, s2-eth2 connected to r2-eth0, and r1-eth1 connected to r2-eth1, all marked as OK. The entire output is enclosed in a gray box.

Figure 12. Displaying network interfaces.

In Figure 12, the link displayed within the gray box indicates that interface eth0 of host h1 connects to interface eth1 of switch s1 (i.e., $h1\text{-}eth0 <-> s1\text{-}eth1$).

2.3 Load zebra daemon and verify configuration

You will verify that the IP addresses listed in Table 2 and inspect the routing table of routers r1 and r2.

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.

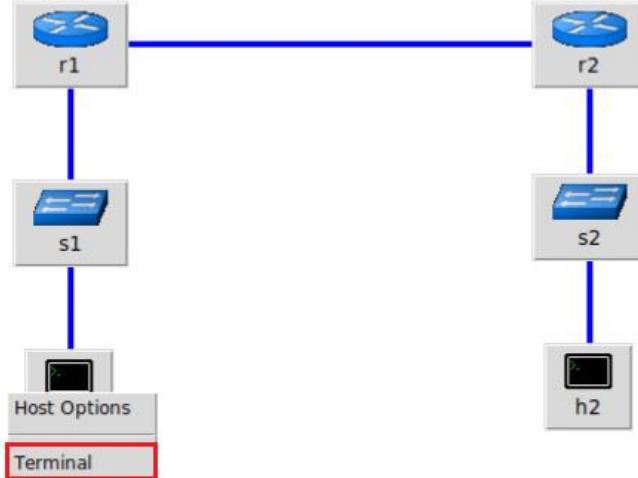


Figure 13. Opening a terminal on host h1.

Step 2. On 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.

```
ifconfig
```

```
"Host: h1"
root@frrr-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
                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@frrr-pc:~#
```

Figure 14. Output of `ifconfig` command.

Step 3. On 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 15. Output of `route` command.

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

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

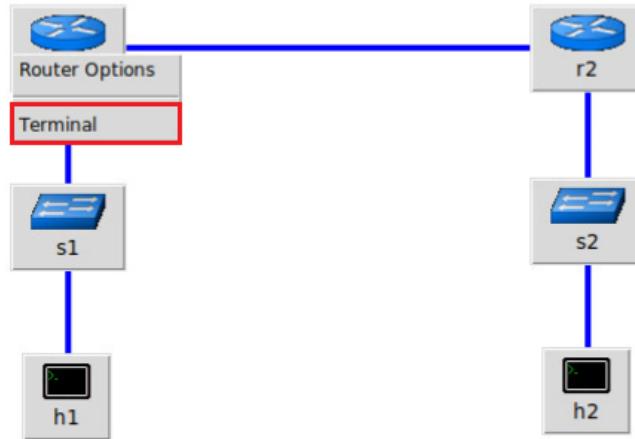


Figure 16. Opening a terminal on router r1.

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

```
zebra
```

The terminal window shows the command "zebra" being run. The output shows the host name "Host: r1" and the command "root@frrr-pc:/etc/routers/r1# zebra". The terminal window has a red border around the command line.

Figure 17. Starting zebra daemon.

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

```
vtysh
```

The terminal window shows the command "vtysh" being run. The output shows the host name "Host: r1" and the command "root@frrr-pc:/etc/routers/r1# vtysh". Below this, it displays the FRRouting version information: "Hello, this is FRRouting (version 7.2-dev). Copyright 1996-2005 Kunihiro Ishiguro, et al." The terminal window has a red border around the command line.

Figure 18. 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
```

The terminal window shows the command "show ip route" being run. The output shows the host name "Host: r1" and the command "frrr-pc# show ip route". Below this, it displays the routing table codes and entries. The entries are: "C>* 192.168.1.0/24 is directly connected, r1-eth0, 00:00:26" and "C>* 192.168.12.0/30 is directly connected, r1-eth1, 00:00:26". The terminal window has a red border around the command line.

Figure 19. Displaying routing table of router r1.

Step 9. In order to verify the routing table of router r2, proceed similarly by repeating from step 5 to step 8 on router terminal. You will verify the directly connected routes in router r2.

3 Configure BGP on the routers

In this section, you will configure BGP in order to establish a connection between AS 100 and AS 200. To configure BGP routing protocol, you need to assign BGP neighbors to allow BGP peering to the remote neighbor, in addition to advertising your local networks.

3.1 Add BGP neighbors on the routers

In this section, you will add the neighbor IP address to allow BGP peering to the remote neighbor.

Step 1. To configure BGP routing protocol, you need to enable the BGP daemon first. In router r1, type the following command to exit the vtysh session:

```
exit
```

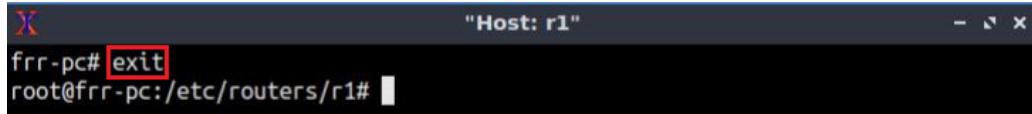


Figure 20. Exiting the vtysh session.

Step 2. Type the following command on router r1 terminal to start BGP routing protocol.

```
bgpd
```



Figure 21. Starting BGP daemon.

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

```
vtysh
```

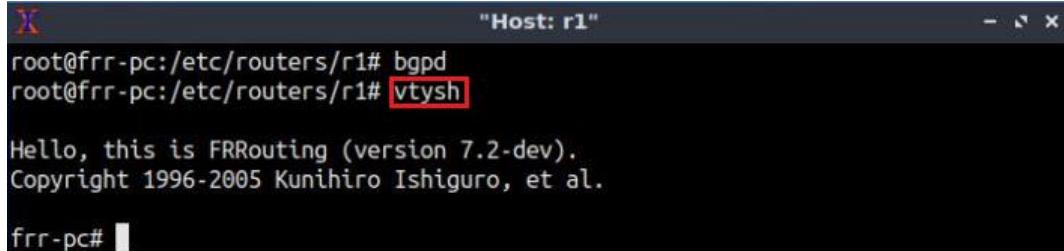
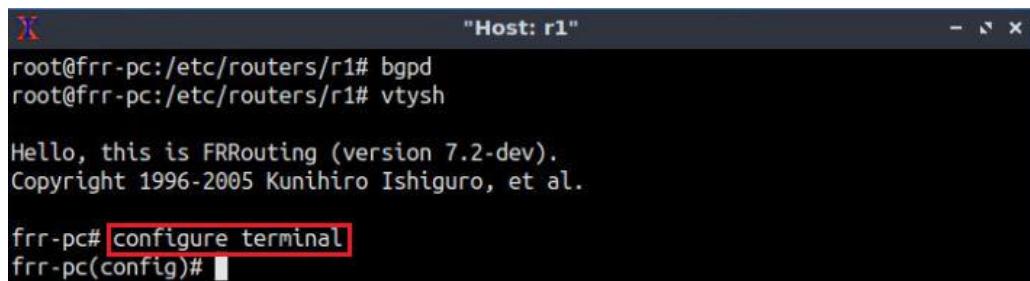


Figure 22. Starting vtysh on router r1.

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

```
configure terminal
```



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

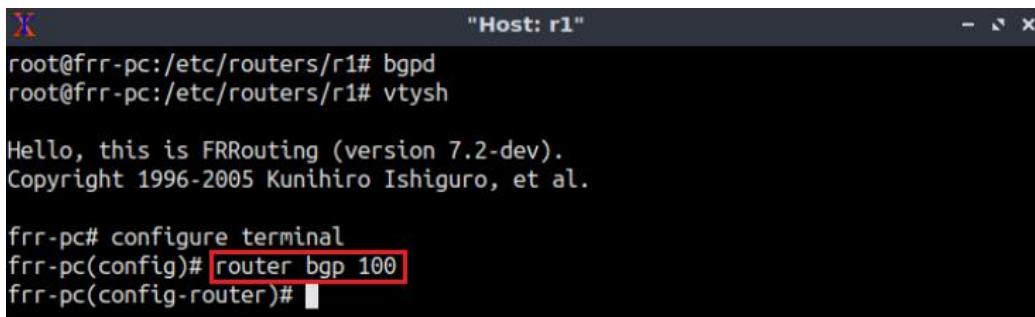
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

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

Figure 23. Enabling configuration mode on router r1.

Step 5. The ASN assigned for router r1 is 100. In order to configure BGP, type the following command:

```
router bgp 100
```



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

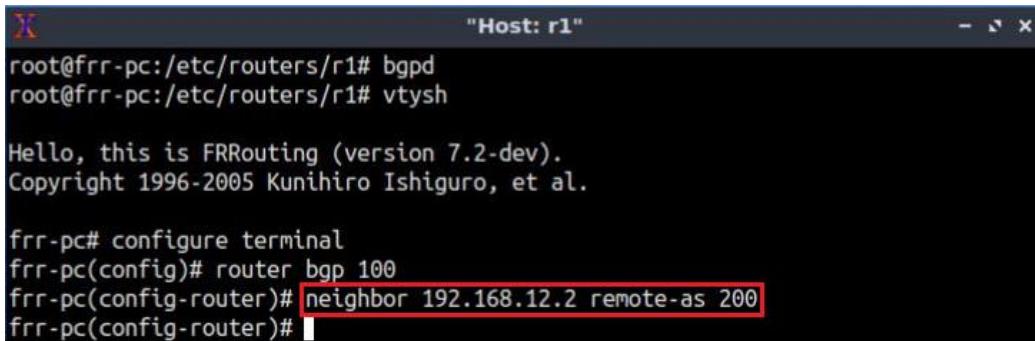
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# router bgp 100
frr-pc(config-router)# 
```

Figure 24. Configuring BGP on router r1.

Step 6. To configure a BGP neighbor to router r1 (AS 100), type the command shown below. This command specifies the neighbor IP address (192.168.12.2) and ASN of the remote BGP peer (AS 200).

```
neighbor 192.168.12.2 remote-as 200
```



```
"Host: r1"
root@frr-pc:/etc/routers/r1# bgpd
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 bgp 100
frr-pc(config-router)# neighbor 192.168.12.2 remote-as 200
frr-pc(config-router)# 
```

Figure 25. Assigning BGP neighbor to router r1.

Step 7. Type the following command to exit from the configuration mode.

```
end
```

```

root@frr-pc:/etc/routers/r1# bgpd
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 bgp 100
frr-pc(config-router)# neighbor 192.168.12.2 remote-as 200
frr-pc(config-router)# end
frr-pc# 
```

Figure 26. Exiting from configuration mode.

Step 8. Type the following command to verify BGP neighbors. You will verify that the neighbor IP address is 192.168.12.2. The corresponding ASN is 200.

```
show ip bgp neighbors
```

```

frr-pc# show ip bgp neighbors
BGP neighbor is [192.168.12.2], remote [AS 200], local AS 100, [external link]
  BGP version 4, remote router ID 0.0.0.0, local router ID 192.168.13.1
  BGP state = Active
  Last read 00:19:08, Last write never
  Hold time is 180, keepalive interval is 60 seconds
  Message statistics:
    Inq depth is 0
    Outq depth is 0
      Sent        Rcvd
      Opens:      0          0
      Notifications: 0          0
      Updates:     0          0
      Keepalives:   0          0
      Route Refresh: 0          0
      Capability:   0          0
      Total:       0          0
  Minimum time between advertisement runs is 0 seconds

  For address family: IPv4 Unicast
    Not part of any update group
    Community attribute sent to this neighbor(all)
    0 accepted prefixes

```

Figure 27. Verifying BGP neighbors on router r1.

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.

The terminal window shows the following session:

```

frr-pc# exit
root@frr-pc:/etc/routers/r2# bgpd
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 bgp 200
frr-pc(config-router)# neighbor 192.168.12.1 remote-as 100
frr-pc(config-router)# end
frr-pc# 

```

Figure 28. Assigning BGP neighbor to router r2.

Step 10. Type the following command to verify BGP neighbors. You will verify that the neighbor IP address is 192.168.12.1. The corresponding ASN is 100.

```
show ip bgp neighbors
```

The terminal window shows the output of the `show ip bgp neighbors` command:

```

frr-pc# show ip bgp neighbors
BGP neighbor is [192.168.12.1], remote [AS 100], local AS 200, [external link]
  BGP version 4, remote router ID 0.0.0.0, local router ID 0.0.0.0
  BGP state = Active
  Last read 00:00:09, Last write never
  Hold time is 180, keepalive interval is 60 seconds
  Message statistics:
    Inq depth is 0
    Outq depth is 0
      Sent          Rcvd
  Opens:          0          0
  Notifications: 0          0
  Updates:        0          0
  Keepalives:     0          0
  Route Refresh: 0          0
  Capability:    0          0
  Total:          0          0
  Minimum time between advertisement runs is 0 seconds

  For address family: IPv4 Unicast
    Not part of any update group
    Community attribute sent to this neighbor(all)
    0 accepted prefixes

```

Figure 29. Verifying BGP neighbors on router r2.

Step 11. In router r2 terminal, perform a connectivity test by running the command shown below. To stop the test, press `Ctrl+C`. The result will show a successful connectivity test between router r1 and router r2.

```
ping 192.168.12.1
```

```
"Host: r2"
frr-pc# ping 192.168.12.1
PING 192.168.12.1 (192.168.12.1) 56(84) bytes of data.
64 bytes from 192.168.12.1: icmp_seq=1 ttl=64 time=0.067 ms
64 bytes from 192.168.12.1: icmp_seq=2 ttl=64 time=0.042 ms
64 bytes from 192.168.12.1: icmp_seq=3 ttl=64 time=0.034 ms
^C
--- 192.168.12.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 30ms
rtt min/avg/max/mdev = 0.034/0.047/0.067/0.016 ms
frr-pc#
```

Figure 30. Connectivity test using `ping` command.

Step 12. In router r2 terminal, perform a connectivity between router r2 and host h1 by issuing the command shown below. To stop the test, press `Ctrl+C`. Router r2 cannot reach to host h1 at this point as the routing table of router r2 does not contain the network address of host h1.

```
ping 192.168.1.10
```

```
"Host: r2"
frr-pc# ping 192.168.1.10
connect: Network is unreachable
frr-pc#
```

Figure 31. Connectivity test using `ping` command.

3.2 Advertise local networks on the routers

In this section, you will advertise a Local Area Network (LAN) so that the neighbor can receive the network address through EBGP.

Step 1. In router r1 terminal, issue the following command.

```
configure terminal
```

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

Figure 32. Enabling configuration mode on router r1.

Step 2. You will advertise the LAN network the router r1 is connected to, via BGP. Type the following command to enable BGP configuration mode.

```
router bgp 100
```

```
"Host: r1"
frr-pc# configure terminal
frr-pc(config)# router bgp 100
frr-pc(config-router)#
```

Figure 33. Entering to BGP configuration mode.

Step 3. Issue the following command so that, router r1 will advertise the network 192.168.1.0/24:

```
network 192.168.1.0/24
```



```
frr-pc# configure terminal
frr-pc(config)# router bgp 100
frr-pc(config-router)# network 192.168.1.0/24
frr-pc(config-router)#

```

Figure 34. Advertising the network connected to router r1.

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

```
end
```



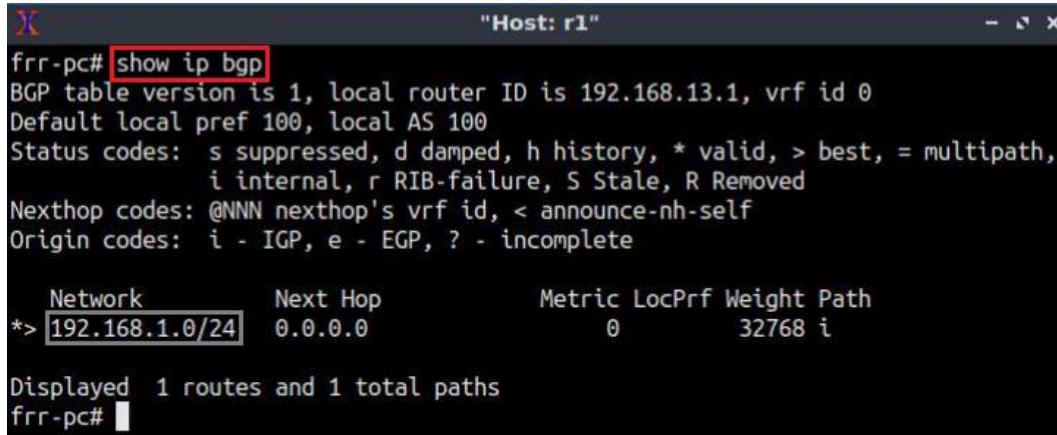
```
frr-pc# configure terminal
frr-pc(config)# router bgp 100
frr-pc(config-router)# network 192.168.1.0/24
frr-pc(config-router)# end
frr-pc#

```

Figure 35. Exiting from configuration mode.

Step 5. Type the following command to verify BGP networks.

```
show ip bgp
```



```
frr-pc# show ip bgp
BGP table version is 1, local router ID is 192.168.13.1, vrf id 0
Default local pref 100, local AS 100
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
              i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*-> 192.168.1.0/24    0.0.0.0                  0        32768 i

Displayed 1 routes and 1 total paths
frr-pc#
```

Figure 36. Verifying BGP networks on router r1.

Step 6. Type the following command to verify the routing table of router r2. You will observe the route to network 192.168.1.0/24, which is advertised by router r1. It also shows that router r2 will use the neighbor IP 192.168.12.1 to reach the network 192.168.1.0/24.

```
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

B>* [192.168.1.0/24] [20/0] via [192.168.12.1], r2-eth1, 00:00:52
C>* 192.168.2.0/24 is directly connected, r2-eth0, 00:18:36
C>* 192.168.12.0/30 is directly connected, r2-eth1, 00:18:02
frr-pc#
```

Figure 37. Verifying routing table of router r2.

Step 7. In order to verify the BGP table of router r2, issue the command shown below. The output indicates that the network connected to router r1 is listed in the BGP table of router r2. Additionally, it displays the next hop IP address (192.168.12.1) which corresponds to router r2's neighbor IP address (router r1).

```
show ip bgp
```

```
frr-pc# show ip bgp
BGP table version is 1, local router ID is 192.168.12.2, vrf id 0
Default local pref 100, local AS 200
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexhop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop           Metric LocPrf Weight Path
*> 192.168.1.0/24    192.168.12.1          0            0 100 i

Displayed  1 routes and 1 total paths
frr-pc#
```

Figure 38. Verifying BGP table of router r2.

Step 8. Follow from step 1 to step 4 but with different metrics in order to advertise the LAN connected to router r2. All these steps are summarized in the following figure.

```
frr-pc# configure terminal
frr-pc(config)# router bgp 200
frr-pc(config-router)# network 192.168.2.0/24
frr-pc(config-router)# end
frr-pc#
```

Figure 39. Advertising the network connected to router r2.

Step 9. In router r2 terminal, issue the following command to verify the BGP table of router r2. The output will list all the available BGP networks. In particular, the routing table contains its own network (192.168.2.0/24) and the remote network (192.168.1.0/24) which was advertised via EBGP.

```
show ip bgp
```

```
frr-pc# show ip bgp
BGP table version is 2, local router ID is 192.168.12.2, vrf id 0
Default local pref 100, local AS 200
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*-> 192.168.1.0/24  192.168.12.1        0          0 100 i
*-> 192.168.2.0/24  0.0.0.0             0          32768 i

Displayed 2 routes and 2 total paths
frr-pc#
```

Figure 40. Verifying BGP table of router r2.

Step 10. In router r1 terminal, verify the routing table by typing the following command. The output lists that router r1 contains a route to the network 192.168.2.0/24. Notice that, this route was advertised by router r2.

```
show ip route
```

```
frr-pc# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       0 - 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, 00:48:10
B>* [192.168.2.0/24] [20/0] via [192.168.12.2], r1-eth0, 00:09:14
C>* 192.168.12.0/30 is directly connected, r1-eth0, 00:48:52
frr-pc#
```

Figure 41. Verifying routing table of router r1.

4 Verify connections

In this section, you will verify that the applied configuration is working correctly by running a connectivity between host h1 and host h2.

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

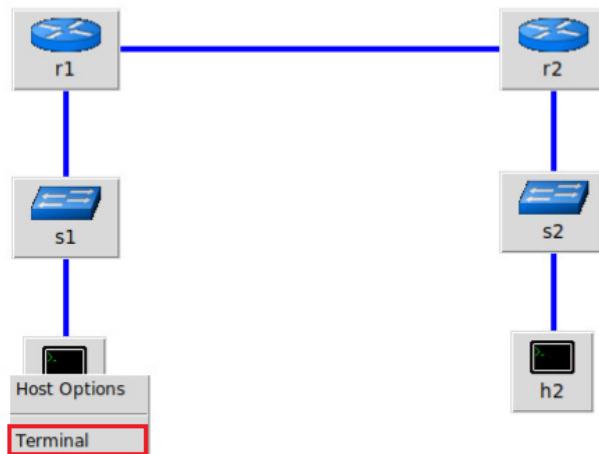


Figure 42. Opening host h1 terminal.

Step 2. On host h1 terminal, perform a connectivity between host h1 and host h2 by issuing the command shown below. To stop the test, press **Ctrl+c**. The result will show a successful connectivity test.

```
ping 192.168.2.10
```

```
X "Host: h1" - x
root@frrr-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=60 time=1.03 ms
64 bytes from 192.168.2.10: icmp_seq=2 ttl=60 time=0.118 ms
64 bytes from 192.168.2.10: icmp_seq=3 ttl=60 time=0.103 ms
64 bytes from 192.168.2.10: icmp_seq=4 ttl=60 time=0.102 ms
^C
--- 192.168.2.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 31ms
rtt min/avg/max/mdev = 0.102/0.338/1.031/0.400 ms
root@frrr-pc:~#
```

Figure 43. Connectivity test using **ping** command.

Step 3. Similarly, hold right-click on host h2 and select *Terminal*. This opens the terminal of host h2.

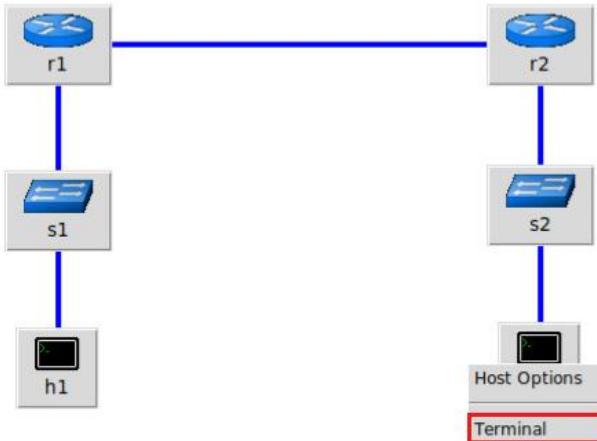


Figure 44. Opening host h2 terminal.

Step 4. Similarly, on host h2 terminal, perform a connectivity between host h2 and host h1 by issuing the command shown below. To stop the test, press **Ctrl+c**. The result will show a successful connectivity test.

```
ping 192.168.1.10
```

```
X "Host: h2" - x
root@frr-pc:~# ping 192.168.1.10
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_seq=1 ttl=60 time=0.136 ms
64 bytes from 192.168.1.10: icmp_seq=2 ttl=60 time=0.110 ms
64 bytes from 192.168.1.10: icmp_seq=3 ttl=60 time=0.115 ms
64 bytes from 192.168.1.10: icmp_seq=4 ttl=60 time=0.105 ms
64 bytes from 192.168.1.10: icmp_seq=5 ttl=60 time=0.102 ms
64 bytes from 192.168.1.10: icmp_seq=6 ttl=60 time=0.089 ms
^C
--- 192.168.1.10 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 113ms
rtt min/avg/max/mdev = 0.089/0.109/0.136/0.017 ms
root@frr-pc:~#
```

Figure 45. Connectivity test using **ping** command.

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

References

1. A. Tanenbaum, D. Wetherall, “Computer networks”, 5th Edition, Pearson, 2012.
2. Linux foundation collaborative projects, “FRR routing documentation”, 2017 [Online] Available: <http://docs.frrouting.org/en/latest/>
3. Y. Rekhter, T. Li, S. Hares, “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>.
4. D. Oran, “OSI IS-IS intra-domain routing protocol,” RFC 1142, 1990. [Online]. Available: <http://www.ietf.org/rfc/rfc1142.txt>.

5. B. Lantz, G. Gee, “MinEdit: a simple network editor for Mininet,” 2013. [Online]. Available: <https://github.com/Mininet/Mininet/blob/master/examples>.



BORDER GATEWAY PROTOCOL

Lab 4: Configure and Verify EBGP

Document Version: **02-17-2020**



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. Intradomain and Interdomain routing protocols.....	4
1.2. Multiprotocol routing and redistribution of protocols	4
2. Lab topology.....	5
2.1. Lab settings.....	6
2.2. Open the topology	7
2.3. Load zebra daemon and verify configuration	10
3. Configure OSPF on router r3 and router r4	14
4. Configure BGP on all routers	19
5. Redistribute routes on router r3 and router r4.....	24
5.1. Inject BGP routes into OSPF	25
5.2. Inject OSPF and directly connected routes into BGP	29
6. Verify connections	32
References	34

Overview

This lab presents Border Gateway Protocol (BGP) and describes the steps to configure and verify the operation of External BGP (EBGP) between two Autonomous Systems (ASes), and Open Shortest Path First (OSPF) protocol within an AS. The focus in this lab is to integrate BGP and OSPF routing protocols by using route redistribution. In this lab, the terms BGP and EBGP will be used interchangeably since they will only be running between ASes.

Objectives

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

1. Explain the concept of BGP.
2. Configure and verify EBGP between two ASes.
3. Enable OSPF redistribution to advertise BGP routes.
4. Enable BGP redistribution to advertise OSPF routes.

Lab settings

The information in Table 1 provides the credentials to access Client1 machine.

Table 1. Credentials to access Client1 machine.

Device	Account	Password
Client1	admin	password

Lab roadmap

This lab is organized as follows:

1. Section 1: Introduction.
2. Section 2: Lab topology.
3. Section 3: Configure OSPF on router r3 and router r4.
4. Section 4: Configure BGP on all routers.
5. Section 5: Redistribute routes on router r3 and router r4.
6. Section 6: Verify connections.

1. Introduction

1.1. Intradomain and Interdomain routing protocols

The Internet consists of many independent administrative domains, referred to as ASes. ASes are operated by different organizations, which can run their own internal routing protocols. A routing protocol that runs within an AS is referred to as intradomain routing protocol. One of the most widely used intradomain protocols is OSPF. Since an AS may be large and nontrivial to manage, OSPF allows an AS to be divided into numbered areas¹. An area is a logical collection of networks, routers, and links. All routers in the same area have detailed information of the topology within their area².

A routing protocol that runs between ASes is referred to as interdomain routing protocol. ASes may use different intradomain routing protocols; however, they must use the same interdomain routing protocol, i.e., BGP. Routers in different ASes exchange information using EBGP. BGP allows the enforcement of different routing policies on the traffic from one AS to another. For example, a security policy can prevent the dissemination of routing information from one AS to another¹.

Consider Figure 1. Routers within AS 1 exchange routing information using OSPF. On the other hand, routers in different ASes exchange routing information using EBGP.

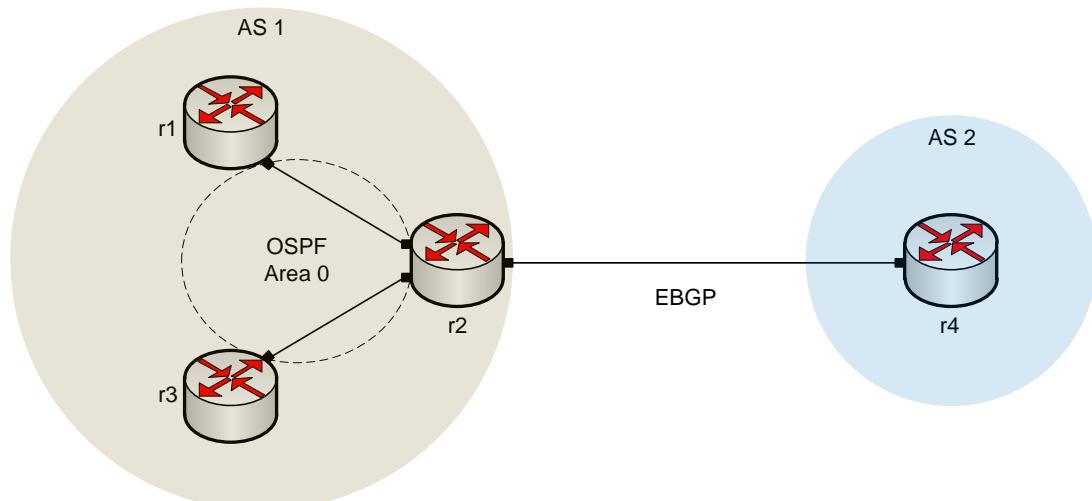


Figure 1. Routers r1, r2, and r3 within AS 1 run OSPF, while routers r2 and r4 in AS 1 and AS 2, respectively, run EBGP.

1.2. Multiprotocol routing and redistribution of protocols

Multi-protocol routing is when two or more routing protocols run in the same router.

The use of a routing protocol to advertise routes that are learned by another routing protocol is called route redistribution³. In Figure 2, router r2 receives routing information from router r1 via EBGP. By using redistribution, this information can then be advertised to AS 2 via OSPF. Similarly, routing information received via OSPF can then be advertised to AS1 via EBGP.

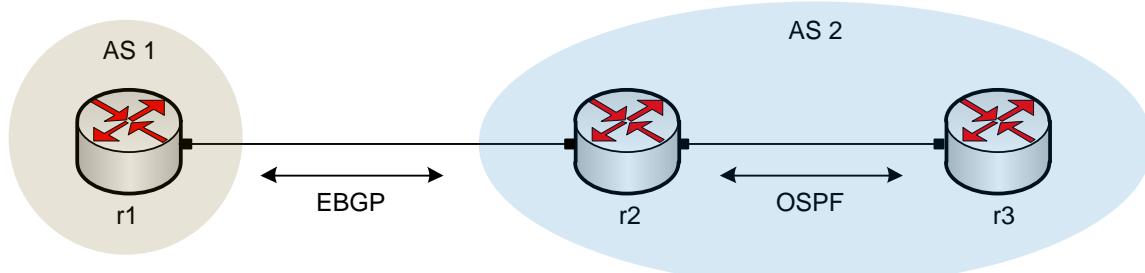


Figure 2. Router r2 redistributes routes between AS1 and AS2.

2. Lab topology

Consider Figure 3. The lab topology consists of three ASes, each identified by an Autonomous System Number (ASN). The ASNs assigned to Campus-1, Campus-2, and the Internet Service Provider (ISP) are 100, 200, and 300, respectively. Campus-1 must exchange routes with Campus-2 using the ISP, which routes the traffic from one AS to another. The communication between ASes is established via EBGP, whereas the communication inside AS 300, i.e., between routers r3 and r4, is established using OSPF.

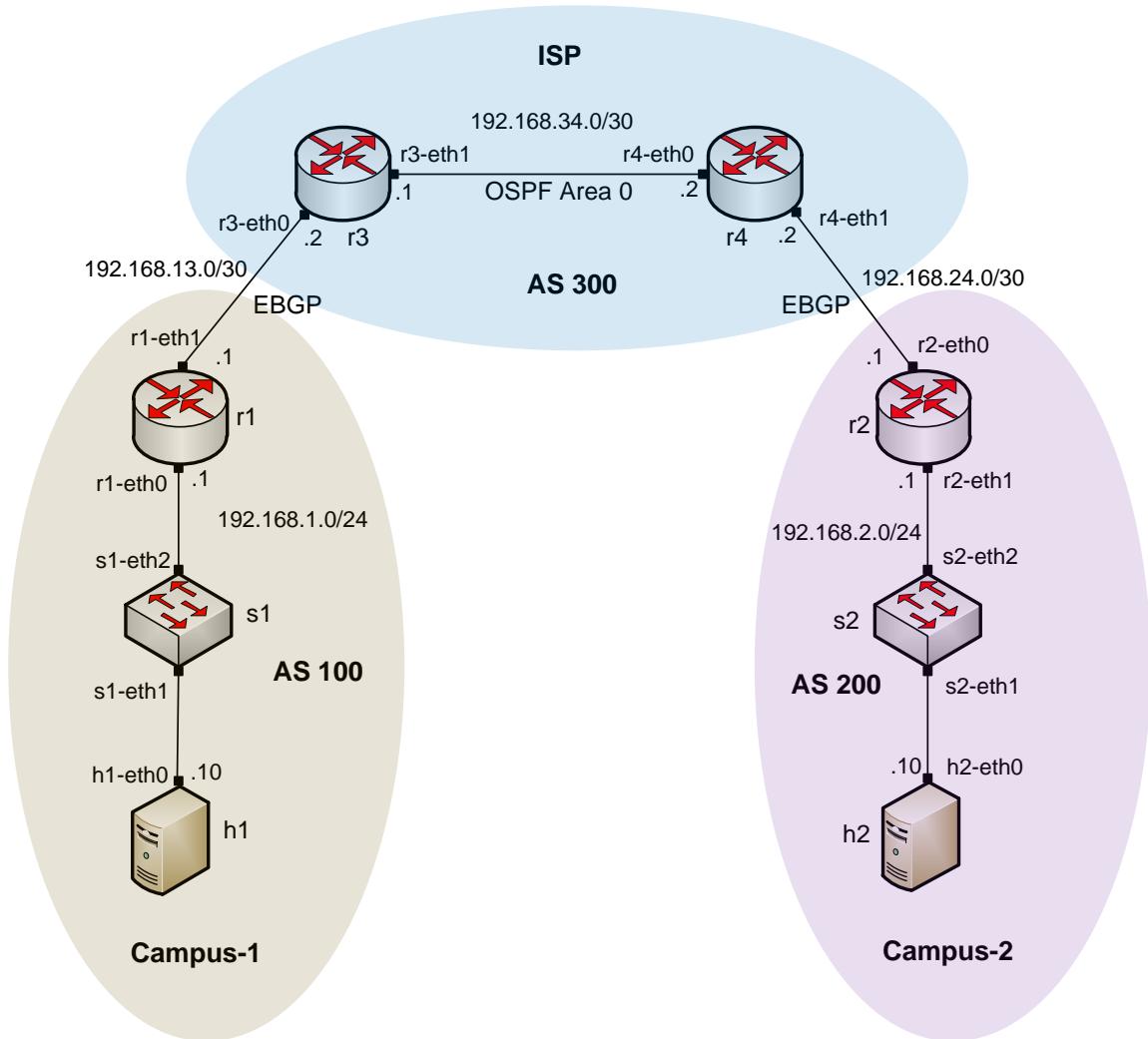


Figure 3. Lab topology.

2.1. Lab settings

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

Table 2. Topology information.

Device	Interface	IPV4 Address	Subnet	Default gateway
r1 (Campus-1)	r1-eth0	192.168.1.1	/24	N/A
	r1-eth1	192.168.13.1	/30	N/A
r2 (Campus-2)	r2-eth0	192.168.2.1	/24	N/A
	r2-eth1	192.168.24.1	/30	N/A
r3 (ISP)	r3-eth0	192.168.13.2	/30	N/A
	r3-eth1	192.168.34.1	/30	N/A

r4 (ISP)	r4-eth0	192.168.34.2	/30	N/A
	r4-eth1	192.168.24.2	/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

Step 1. Start by launching Miniedit by clicking on Desktop's shortcut. When prompted for a password, type `password`.



Figure 4. MiniEdit shortcut.

Step 2. On Miniedit's menu bar, click on *File* then *open* to load the lab's topology. Open the *Lab4.mn* topology file stored in the default directory, */home/frr/BGP_Labs/lab4* and click on *Open*.

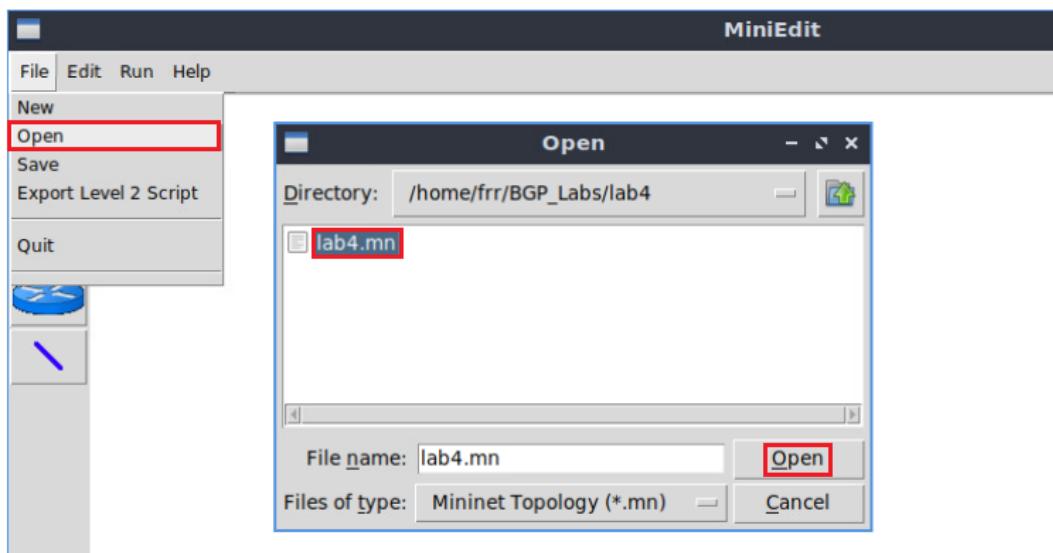


Figure 5. MiniEdit's open dialog.

At this point the topology is loaded with all the required network components. You will execute a script that will load the configuration of the routers.

Step 3. Open the Linux terminal.

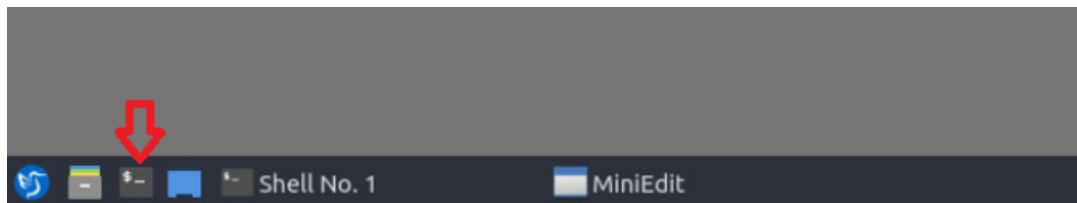


Figure 6. Opening Linux terminal.

Step 4. Click on the Linux's terminal and navigate into *BGP_Labs/lab4* 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 BGP_Labs/lab4
```

A screenshot of a terminal window. The title bar says "frr@frr-pc: ~/BGP_Labs/lab4". The terminal shows the following session:

```
frr@frr-pc: ~$ cd BGP_Labs/lab4
frr@frr-pc:~/BGP_Labs/lab4$
```

The command `cd BGP_Labs/lab4` is highlighted with a red box.

Figure 7. Entering the *BGP_Labs/lab4* directory.

Step 5. To execute the shell script, type the following command. The argument of the program corresponds to the configuration zip file that will be loaded in all the routers in the topology.

```
./config_loader.sh lab4_conf.zip
```

A screenshot of a terminal window. The title bar says "frr@frr-pc: ~/BGP_Labs/lab4". The terminal shows the following session:

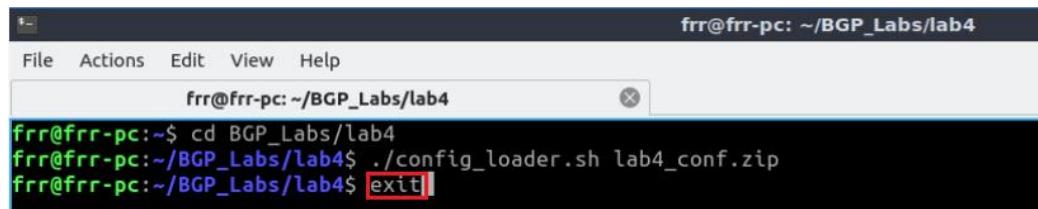
```
frr@frr-pc: ~$ cd BGP_Labs/lab4
frr@frr-pc:~/BGP_Labs/lab4$ ./config_loader.sh lab4_conf.zip
frr@frr-pc:~/BGP_Labs/lab4$
```

The command `./config_loader.sh lab4_conf.zip` is highlighted with a red box.

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

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

```
exit
```



```
frr@frr-pc: ~/BGP_Labs/lab4
File Actions Edit View Help
frr@frr-pc: ~/BGP_Labs/lab4
frr@frr-pc:~/BGP_Labs/lab4$ ./config_loader.sh lab4_conf.zip
frr@frr-pc:~/BGP_Labs/lab4$ exit
```

Figure 9. Exiting from the terminal.

Step 7. At this point hosts h1 and h2 interfaces are configured. To proceed with the emulation, click on the *Run* button located in lower left-hand side.



Figure 10. Starting the emulation.

Step 8. Click on Mininet's terminal, i.e., the one launched when MiniEdit was started.

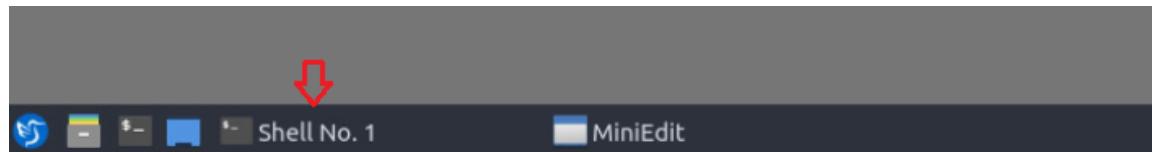
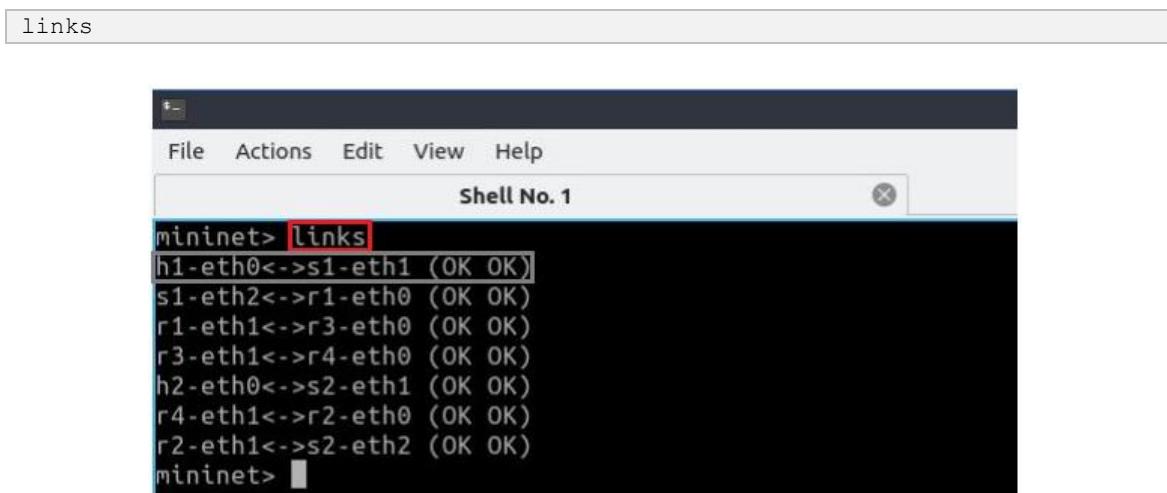


Figure 11. Opening Mininet's terminal.

Step 9. 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)
r1-eth1<->r3-eth0 (OK OK)
r3-eth1<->r4-eth0 (OK OK)
h2-eth0<->s2-eth1 (OK OK)
r4-eth1<->r2-eth0 (OK OK)
r2-eth1<->s2-eth2 (OK OK)
mininet>
```

Figure 12. Displaying network interfaces.

In Figure 12, the link displayed within the gray box indicates that interface *eth0* of host h1 connects to interface *eth1* of switch s1 (i.e., *h1-eth0<->s1-eth1*).

2.3. Load zebra daemon and verify configuration

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

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

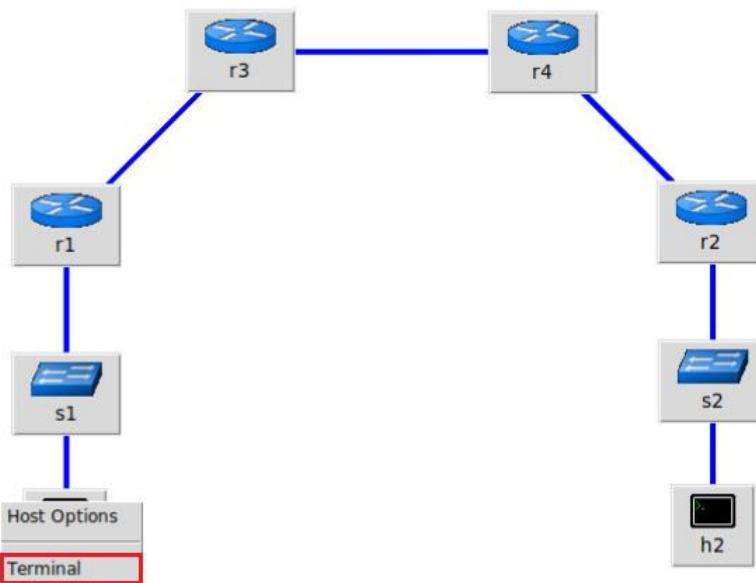
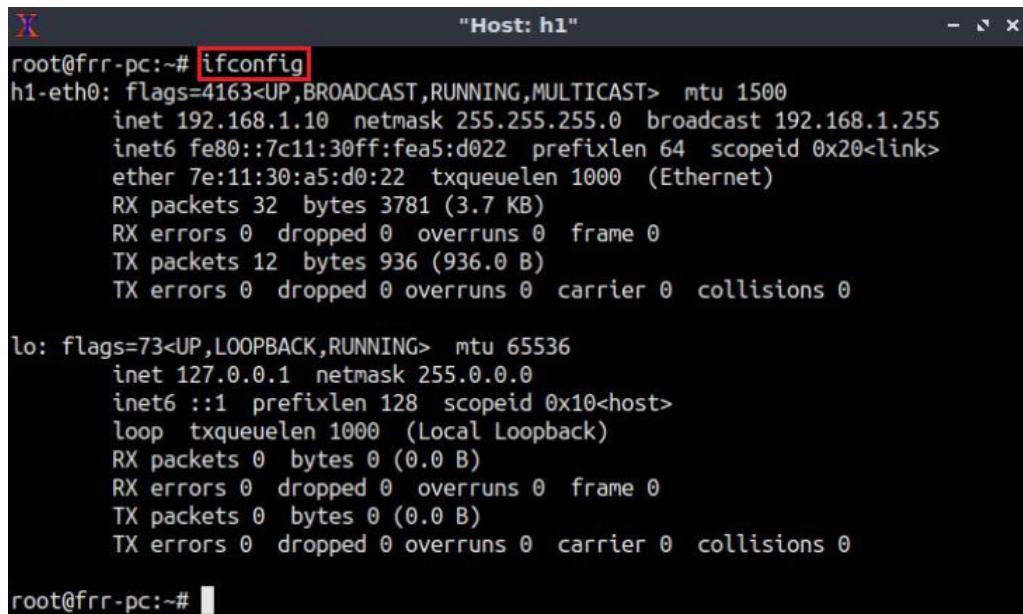


Figure 13. Opening terminal on host h1.

Step 2. On 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.

```
ifconfig
```



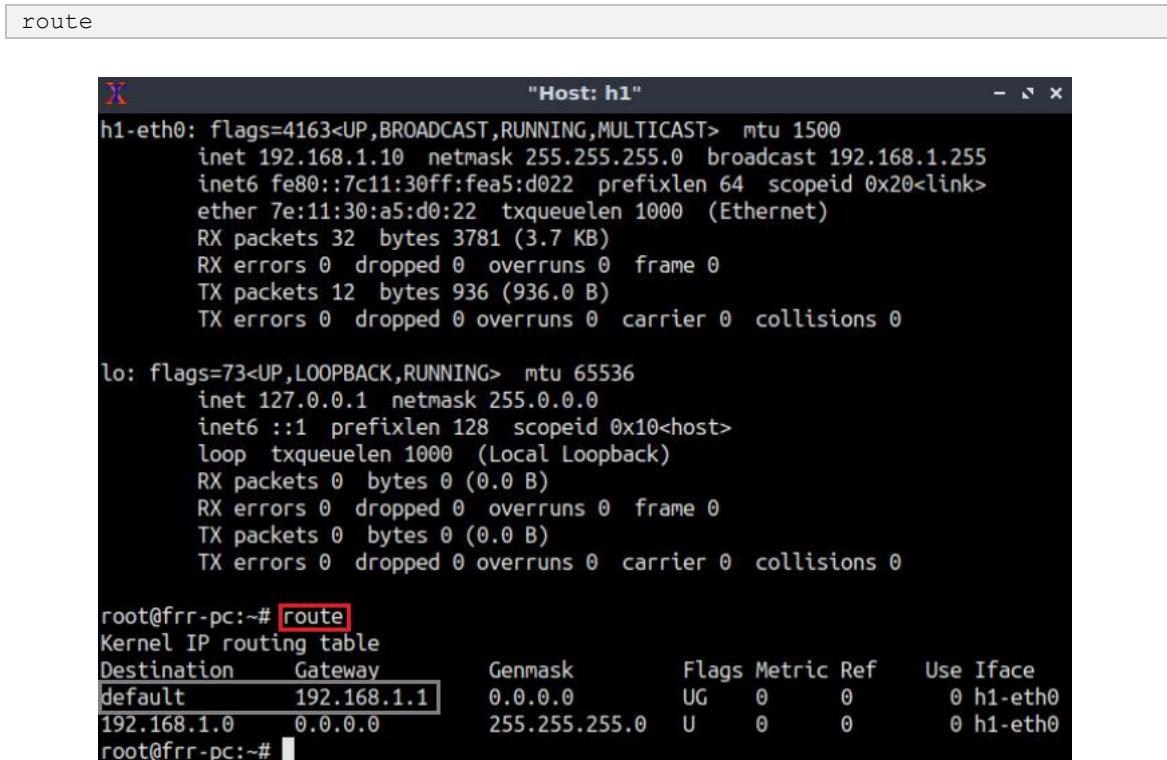
```
"Host: h1"
root@frr-pc:~# ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.1.10 netmask 255.255.255.0 broadcast 192.168.1.255
        inet6 fe80::7c11:30ff:fea5:d022 prefixlen 64 scopeid 0x20<link>
          ether 7e:11:30:a5:d0:22 txqueuelen 1000 (Ethernet)
            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 14. Output of `ifconfig` command.

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



```
route
"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 15. Output of `route` command.

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

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

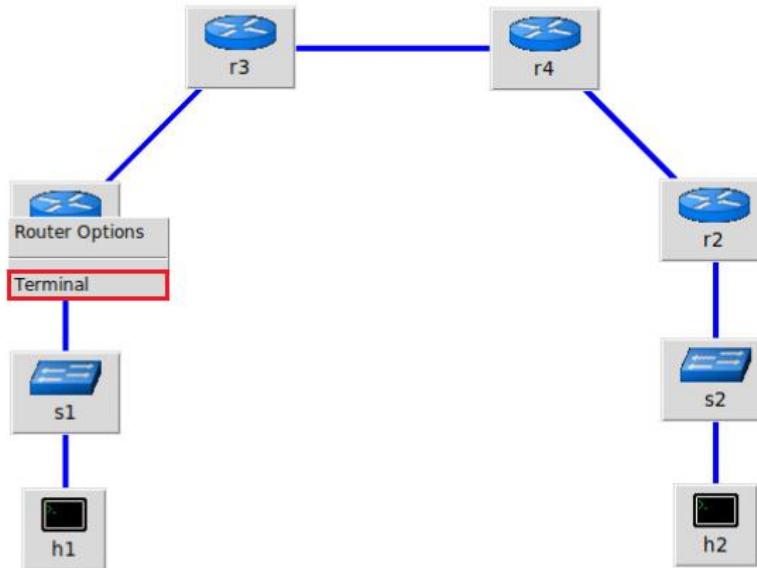


Figure 16. Opening terminal on router r1.

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

```
zebra
```

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

Figure 17. Starting zebra daemon.

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

```
vtysh
```

```
"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 18. Starting vtysh in 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 attached to router r2 (192.168.2.0/24) and router r4 (192.168.24.0/30, 192.168.34.0/30) as there is no routing protocol configured yet.

```
show ip route
```

```

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, 00:00:12
C>* 192.168.13.0/30 is directly connected, r1-eth1, 00:00:12
frr-pc# 
```

Figure 19. Displaying routing table of router r1.

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.

```

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

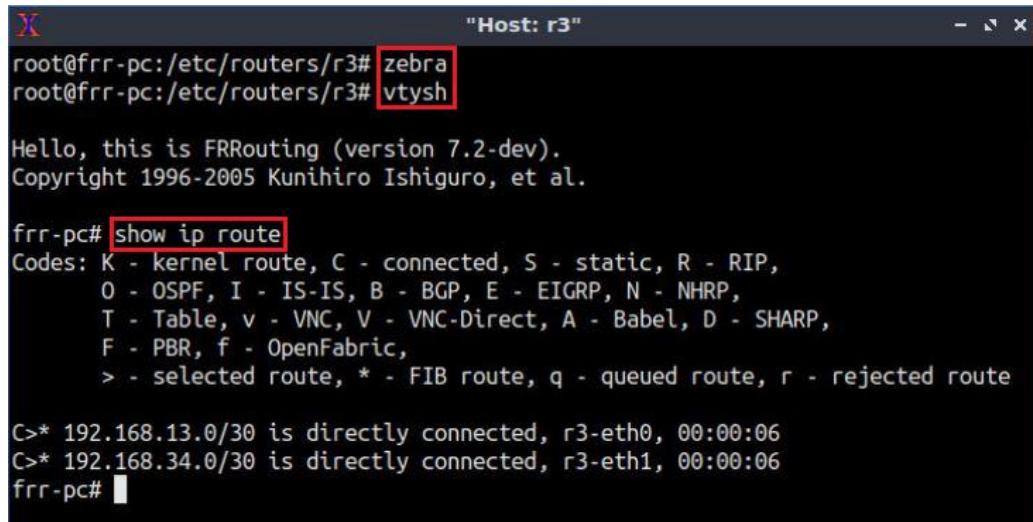
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

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-eth1, 00:00:06
C>* 192.168.24.0/30 is directly connected, r2-eth0, 00:00:06
frr-pc# 
```

Figure 20. Displaying routing table of router r2.

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



The terminal window shows the command "show ip route" being entered. The output displays the routing table with two entries: "192.168.13.0/30" and "192.168.34.0/30", both marked as directly connected via interfaces r3-eth0 and r3-eth1 respectively.

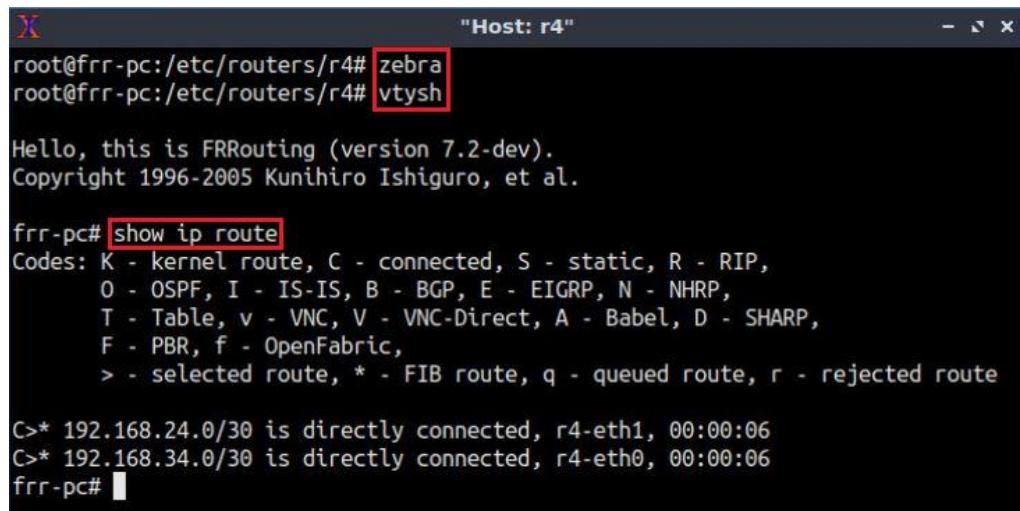
```
"Host: r3"
root@frr-pc:/etc/routers/r3# zebra
root@frr-pc:/etc/routers/r3# vtysh
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

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>* 192.168.13.0/30 is directly connected, r3-eth0, 00:00:06
C>* 192.168.34.0/30 is directly connected, r3-eth1, 00:00:06
frr-pc#
```

Figure 21. Displaying routing table of router r3.

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



The terminal window shows the command "show ip route" being entered. The output displays the routing table with two entries: "192.168.24.0/30" and "192.168.34.0/30", both marked as directly connected via interfaces r4-eth1 and r4-eth0 respectively.

```
"Host: r4"
root@frr-pc:/etc/routers/r4# zebra
root@frr-pc:/etc/routers/r4# 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>* 192.168.24.0/30 is directly connected, r4-eth1, 00:00:06
C>* 192.168.34.0/30 is directly connected, r4-eth0, 00:00:06
frr-pc#
```

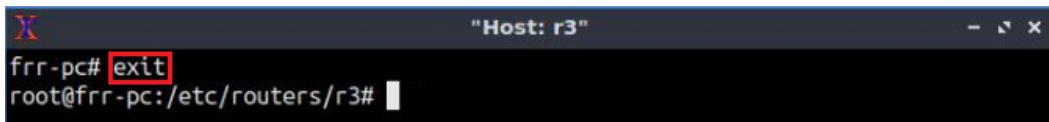
Figure 22. Displaying routing table of router r4.

3. Configure OSPF on router r3 and router r4

In this section, you will configure OSPF routing protocol in router r3 and router r4. First, you will enable the OSPF daemon router r3 and router r4. Second, you will establish single area OSPF, which is classified as area 0 or backbone area. Finally, all the connected networks are advertised.

Step 1. To configure OSPF routing protocol, you need to enable the OSPF daemon first. In router r3, type the following command to exit the vtysh session:

```
exit
```

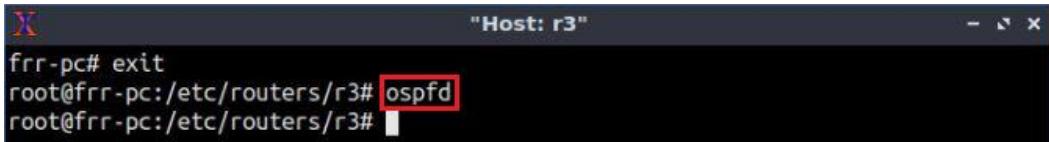


```
frr-pc# exit
root@frr-pc:/etc/routers/r3#
```

Figure 23. Exiting the vtysh session.

Step 2. Type the following command on router r3 terminal to enable OSPF daemon.

```
ospfd
```

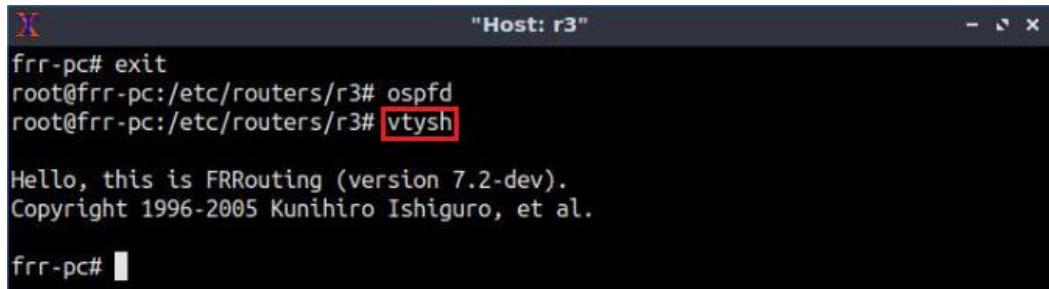


```
frr-pc# exit
root@frr-pc:/etc/routers/r3# ospfd
root@frr-pc:/etc/routers/r3#
```

Figure 24. Starting OSPF daemon.

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

```
vtysh
```



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

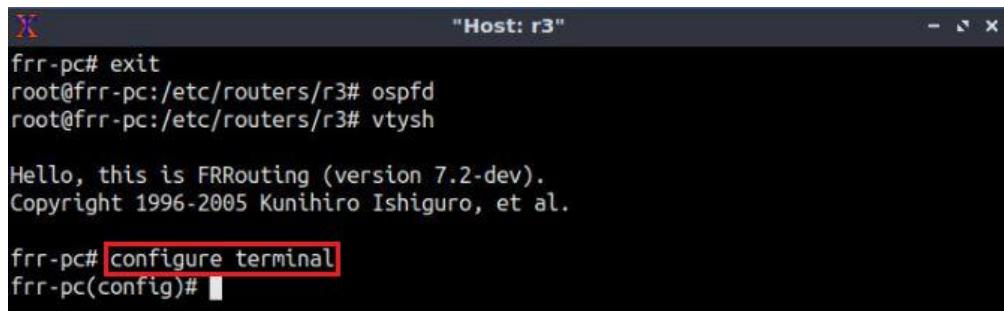
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc#
```

Figure 25. Starting vtysh on router r3.

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

```
configure terminal
```



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

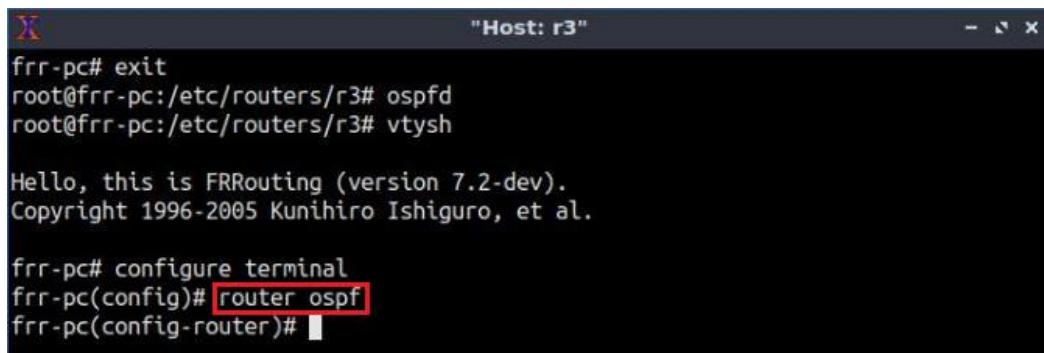
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

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

Figure 26. Enabling configuration mode on router r3.

Step 5. In order to configure OSPF routing protocol, type the command shown below. This command enables OSPF configuration mode where you advertise the networks directly connected to router r3.

```
router ospf
```



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

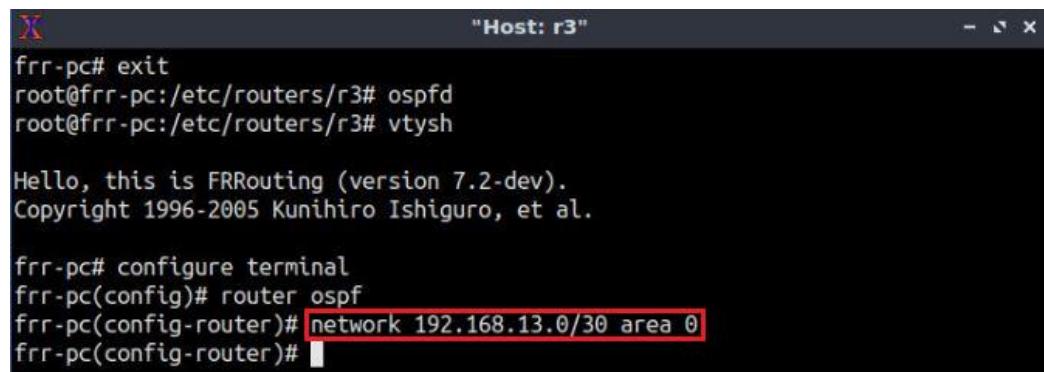
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

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

Figure 27. Configuring OSPF on router r3.

Step 6. In this step, type the following command to enable the interface *r3-eth0*, corresponding to the network 192.168.13.0/30, to participate in the routing process. This network is associated with area 0.

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



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

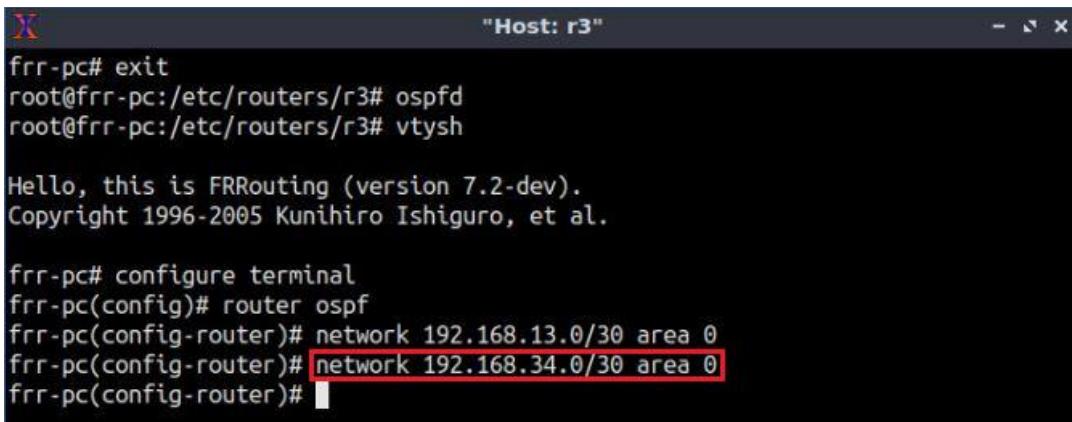
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# network 192.168.13.0/30 area 0
frr-pc(config-router)#
```

Figure 28. Enabling the interface corresponding to the network 192.168.13.0/30 to participate in the OSPF routing process.

Step 7. Similarly, type the following command on router r3 terminal to enable the interface *r3-eth1* to participate in the OSPF routing process.

```
network 192.168.34.0/30 area 0
```



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

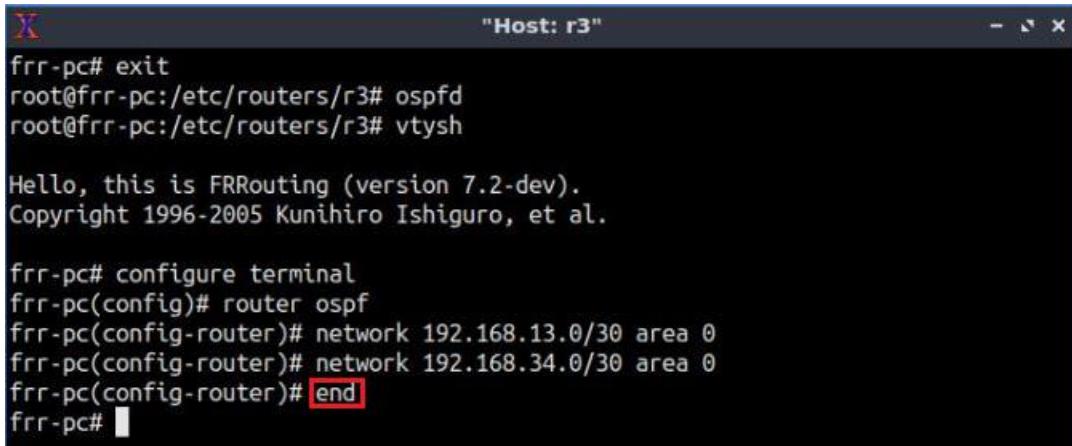
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# network 192.168.13.0/30 area 0
frr-pc(config-router)# network 192.168.34.0/30 area 0
frr-pc(config-router)#
```

Figure 29. Enabling the interface corresponding to 192.168.34.0/30 to participate in the OSPF routing process.

Step 8. Type the following command to exit from the configuration mode.

```
end
```



```
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)# network 192.168.13.0/30 area 0
frr-pc(config-router)# network 192.168.34.0/30 area 0
frr-pc(config-router)# end
frr-pc#
```

Figure 30. Exiting from configuration mode.

Step 9. 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, on router r4 terminal, issue the commands depicted below.

```

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)# network 192.168.34.0/30 area 0
frr-pc(config-router)# network 192.168.24.0/30 area 0
frr-pc(config-router)# end
frr-pc# 

```

Figure 31. Configuring OSPF on router r4.

Step 10. Type the following command to verify the routing table of router r4.

```

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>* 192.168.13.0/30 [110/20] via 192.168.34.1, r4-eth0, 00:00:00
O  192.168.24.0/30 [110/10] is directly connected, r4-eth1, 00:00:12
C>* 192.168.24.0/30 is directly connected, r4-eth1, 00:05:02
O  192.168.34.0/30 [110/10] is directly connected, r4-eth0, 00:00:10
C>* 192.168.34.0/30 is directly connected, r4-eth0, 00:05:17
frr-pc# 

```

Figure 32. Verifying the routing table of router r4.

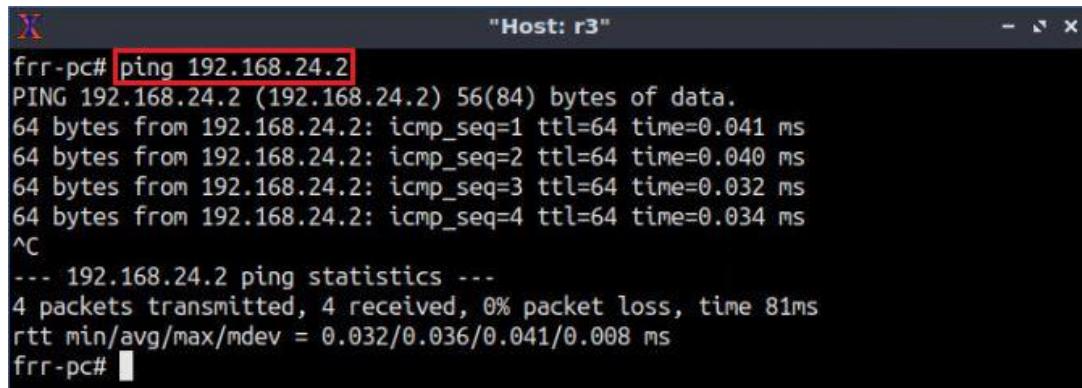
Consider Figure 32. Three additional networks (192.168.13.0/30, 192.168.24.0/30 and 192.168.34.0/30) advertised by OSPF. Router r4 reaches the network 192.168.13.0/30 via the IP address 192.168.34.1. Networks 192.168.24.0/30 and 192.168.34.0/30 have two available paths from router r4. The Administrative Distance (AD) of the paths advertised through OSPF is 110. The AD is a value used by routers to select the best path when there are multiple available routes to the same destination. A smaller AD is always preferable to the routers. The characters **[]** indicates that the following path is used to reach a specific network. Router r4 prefers directly connected networks over OSPF since the former has a lower AD than the latter.

Step 11. In router r3 terminal, test the connectivity between routers r3 and r4 using the **ping** command. Router r3 should ping the IP address 192.168.24.2 after configuring the OSPF routing protocol. To stop the test, press **Ctrl+c**. The figure below shows a successful connectivity test.

```

ping 192.168.24.2

```



```
"Host: r3"
frr-pc# ping 192.168.24.2
PING 192.168.24.2 (192.168.24.2) 56(84) bytes of data.
64 bytes from 192.168.24.2: icmp_seq=1 ttl=64 time=0.041 ms
64 bytes from 192.168.24.2: icmp_seq=2 ttl=64 time=0.040 ms
64 bytes from 192.168.24.2: icmp_seq=3 ttl=64 time=0.032 ms
64 bytes from 192.168.24.2: icmp_seq=4 ttl=64 time=0.034 ms
^C
--- 192.168.24.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 81ms
rtt min/avg/max/mdev = 0.032/0.036/0.041/0.008 ms
frr-pc#
```

Figure 33. Output of `ping` command on router r3.

4. Configure BGP on all routers

In this section, you will configure EBGP in the routers that are hosted in different ASes. You will assign BGP neighbors to allow the routers to exchange BGP routes. Furthermore, routers r1 and r2 will advertise their LANs via BGP. Therefore, router r3 and router r4 will receive route information about LAN 192.168.1.0/24 and 192.168.2.0/24, respectively.

Step 1. To configure BGP routing protocol, you need to enable the BGP daemon first. In router r1, type the following command to exit the vtysh session:



```
exit
```

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

Figure 34. Exiting the vtysh session.

Step 2. Type the following command on router r1 terminal to enable and to start BGP routing protocol.



```
bgpd
```

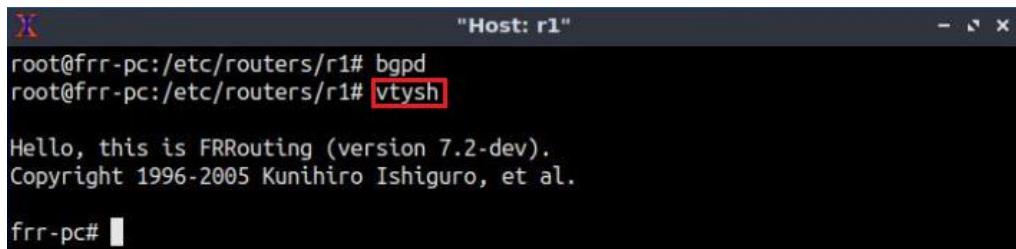
```
"Host: r1"
root@frr-pc:/etc/routers/r1# bgpd
root@frr-pc:/etc/routers/r1#
```

Figure 35. Starting BGP daemon.

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



```
vtysh
```



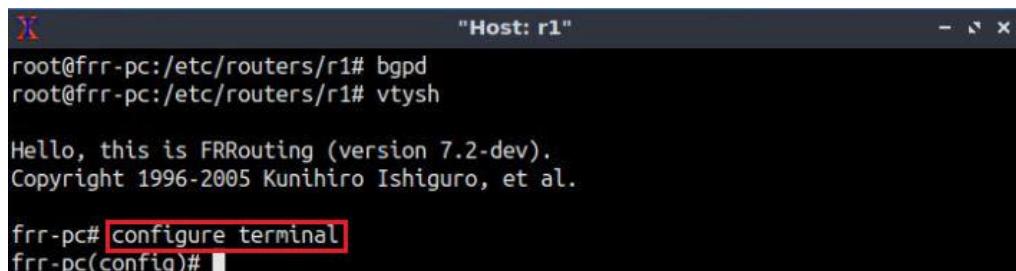
```
"Host: r1"
root@frr-pc:/etc/routers/r1# bgpd
root@frr-pc:/etc/routers/r1# vtysh
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc#
```

Figure 36. Starting vtysh on router r1.

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

```
configure terminal
```



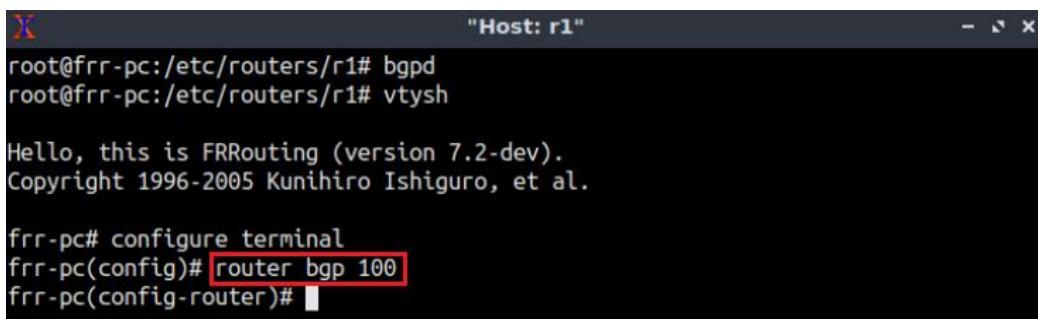
```
"Host: r1"
root@frr-pc:/etc/routers/r1# bgpd
root@frr-pc:/etc/routers/r1# vtysh
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

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

Figure 37. Enabling configuration mode on router r1.

Step 5. The ASN assigned for router r1 is 100. In order to configure BGP, type the following command:

```
router bgp 100
```



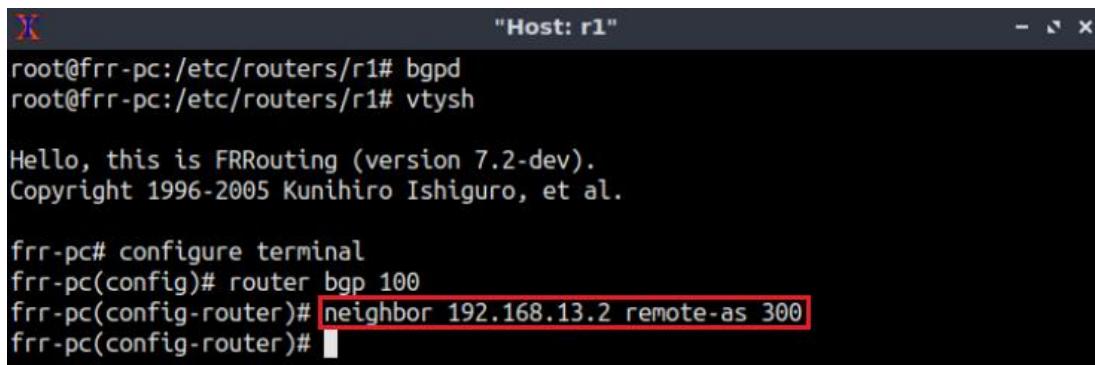
```
"Host: r1"
root@frr-pc:/etc/routers/r1# bgpd
root@frr-pc:/etc/routers/r1# vtysh
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# router bgp 100
frr-pc(config-router)#
```

Figure 38. Configuring BGP on router r1.

Step 6. To configure a BGP neighbor to router r1 (AS 100), type the command shown below. This command specifies the neighbor IP address (192.168.13.2) and ASN of the remote BGP peer (AS 300).

```
neighbor 192.168.13.2 remote-as 300
```



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

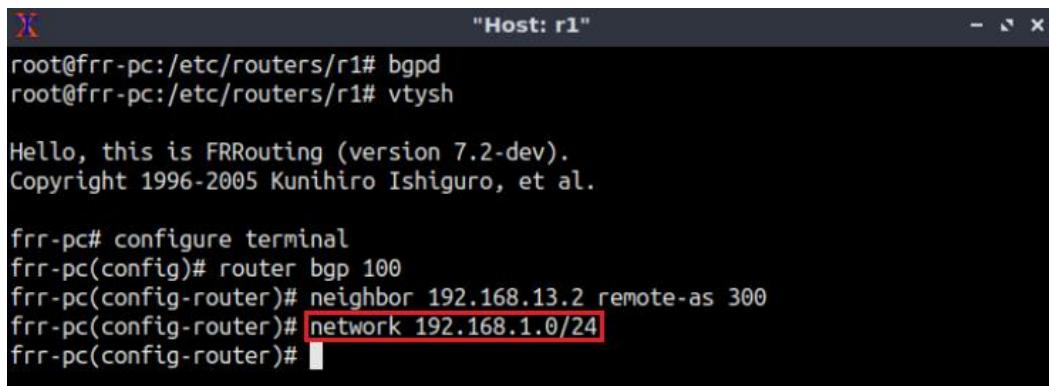
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# router bgp 100
frr-pc(config-router)# neighbor 192.168.13.2 remote-as 300
frr-pc(config-router)# 
```

Figure 39. Assigning BGP neighbor to router r1.

Step 7. In this step, router r1 will advertise the Local Area Network (LAN) 192.168.1.0/24 to router r3 through EBGP. To do so, issue the following command:

```
network 192.168.1.0/24
```



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

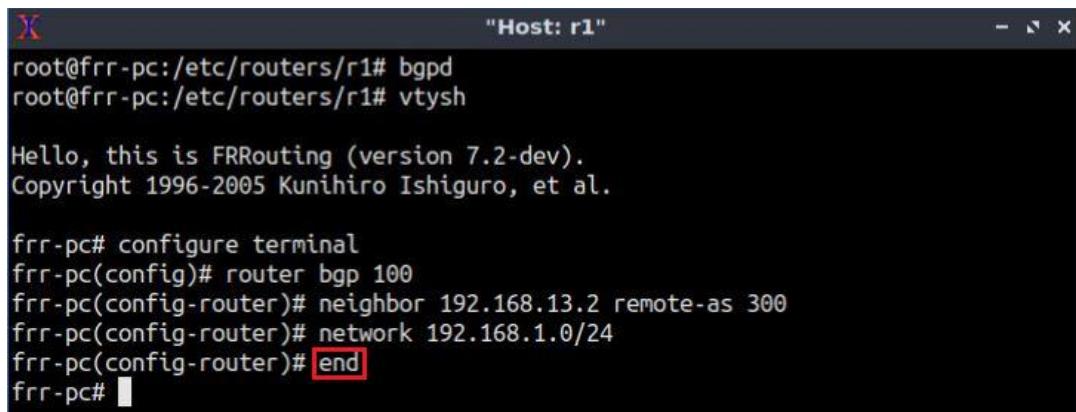
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# router bgp 100
frr-pc(config-router)# neighbor 192.168.13.2 remote-as 300
frr-pc(config-router)# network 192.168.1.0/24
frr-pc(config-router)# 
```

Figure 40. Advertising a network on router r1.

Step 8. Type the following command to exit from the configuration mode.

```
end
```



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

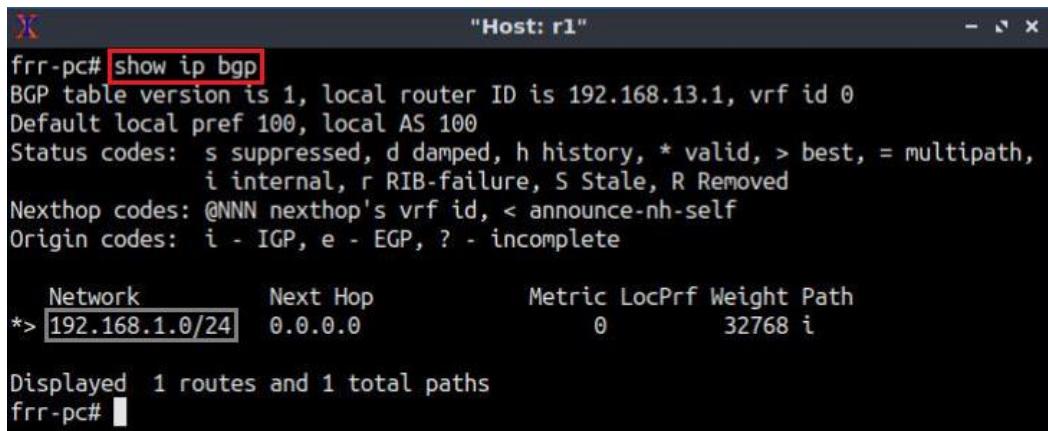
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# router bgp 100
frr-pc(config-router)# neighbor 192.168.13.2 remote-as 300
frr-pc(config-router)# network 192.168.1.0/24
frr-pc(config-router)# end
frr-pc# 
```

Figure 41. Exiting from configuration mode.

Step 9. Type the following command to verify BGP networks. You will observe the LAN network of router r1.

```
show ip bgp
```



The terminal window shows the output of the 'show ip bgp' command. It displays the BGP table version, local router ID, vrf id, and various status and nexthop codes. A single route is listed with a network of 192.168.1.0/24, next hop 0.0.0.0, metric 0, local preference 32768, weight 1, and path i.

```
frr-pc# show ip bgp
BGP table version is 1, local router ID is 192.168.13.1, vrf id 0
Default local pref 100, local AS 100
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

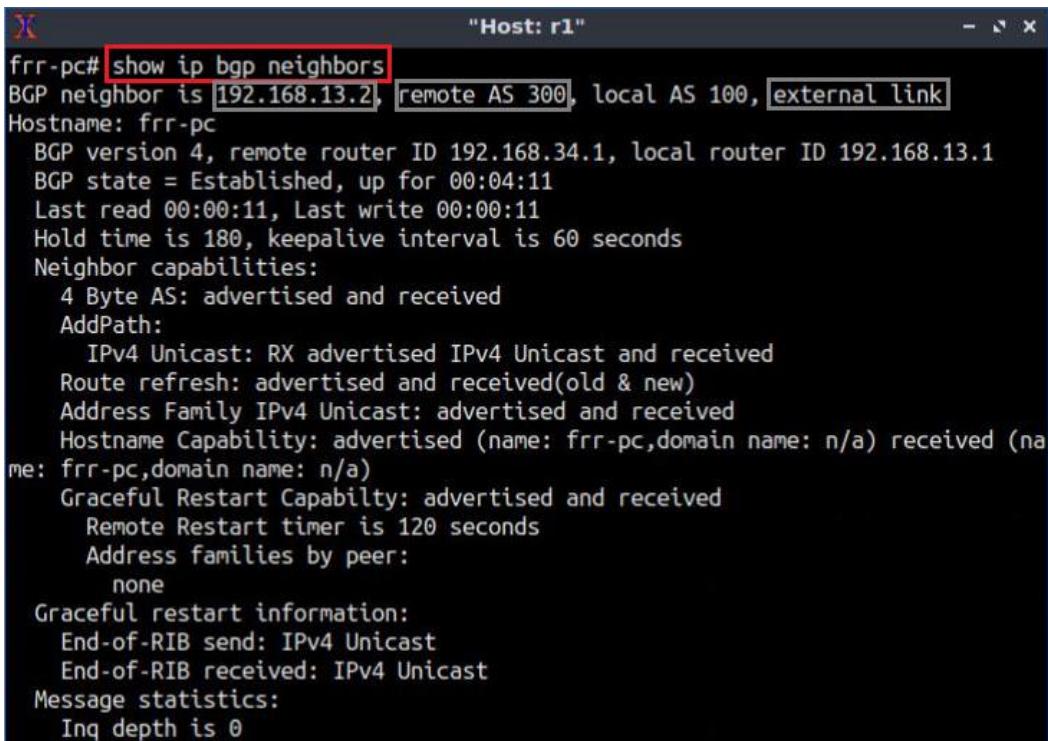
Network          Next Hop           Metric LocPrf Weight Path
*> [192.168.1.0/24]  0.0.0.0                  0        32768 i

Displayed 1 routes and 1 total paths
frr-pc#
```

Figure 42. Verifying BGP networks on router r1.

Step 10. Type the following command to verify BGP neighbors. You will verify that the neighbor IP address is 192.168.13.2. The corresponding ASN is 300.

```
show ip bgp neighbors
```



The terminal window shows the output of the 'show ip bgp neighbors' command. It details a BGP neighbor at 192.168.13.2 with remote AS 300, local AS 100, and an external link. The neighbor has BGP version 4, state Established, and a hold time of 180 seconds. It lists neighbor capabilities like 4 Byte AS, AddPath, and Graceful Restart. It also shows message statistics and address families.

```
frr-pc# show ip bgp neighbors
BGP neighbor is [192.168.13.2], [remote AS 300], local AS 100, [external link]
Hostname: frr-pc
BGP version 4, remote router ID 192.168.34.1, local router ID 192.168.13.1
BGP state = Established, up for 00:04:11
Last read 00:00:11, Last write 00:00:11
Hold time is 180, keepalive interval is 60 seconds
Neighbor capabilities:
  4 Byte AS: advertised and received
  AddPath:
    IPv4 Unicast: RX advertised IPv4 Unicast and received
    Route refresh: advertised and received(old & new)
    Address Family IPv4 Unicast: advertised and received
    Hostname Capability: advertised (name: frr-pc, domain name: n/a) received (name: frr-pc, domain name: n/a)
    Graceful Restart Capabilty: advertised and received
      Remote Restart timer is 120 seconds
      Address families by peer:
        none
    Graceful restart information:
      End-of-RIB send: IPv4 Unicast
      End-of-RIB received: IPv4 Unicast
    Message statistics:
      Inq depth is 0
```

Figure 43. Verifying BGP neighbors on router r1.

Step 11. Follow from step 1 to step 8 but with different metrics in order to configure BGP in router r2. All these steps are summarized in the following figure.

```
frr-pc# exit
root@frr-pc:/etc/routers/r2# bgpd
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 bgp 200
frr-pc(config-router)# neighbor 192.168.24.2 remote-as 300
frr-pc(config-router)# network 192.168.2.0/24
frr-pc(config-router)# end
frr-pc#
```

Figure 44. Configuring BGP on router r2.

Step 12. To configure BGP on router r3, you will add the neighbor router r1 so that router r3 receives router r1 advertised routes. To do so, type all the commands summarized in the following figure.

```
frr-pc# exit
root@frr-pc:/etc/routers/r3# bgpd
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 bgp 300
frr-pc(config-router)# neighbor 192.168.13.1 remote-as 100
frr-pc(config-router)# exit
frr-pc(config)#
```

Figure 45. Configuring BGP on router r3.

Step 13. To configure BGP on router r4, you will add the neighbor router r2 so that router r4 receives the network address of router r2. To do so, type all the commands summarized in the following figure.

```
frr-pc# exit
root@frr-pc:/etc/routers/r4# bgpd
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 bgp 300
frr-pc(config-router)# neighbor 192.168.24.1 remote-as 200
frr-pc(config-router)# end
frr-pc#
```

Figure 46. Configuring BGP on router r4.

Step 14. Type the following command to verify the routing table of router r4. The LAN of router r2 network (192.168.2.0/24) is advertised to router r4 through EBGP.

```
show ip route
```

```
frr-pc# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       0 - 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

B>* [192.168.2.0/24] [20/0] via 192.168.24.1, r4-eth1, 00:00:24
0>* 192.168.13.0/30 [110/20] via 192.168.34.1, r4-eth0, 03:02:48
0 192.168.24.0/30 [110/10] is directly connected, r4-eth1, 03:03:00
C>* 192.168.24.0/30 is directly connected, r4-eth1, 03:07:50
0 192.168.34.0/30 [110/10] is directly connected, r4-eth0, 03:02:58
C>* 192.168.34.0/30 is directly connected, r4-eth0, 03:08:05
frr-pc#
```

Figure 47. Verifying the routing table of router r4.

Step 15. Type the following command to verify the BGP table of router r4. The network address 192.168.2.0/24 of the LAN connected to router r2 is present in router r4 BGP table. The table also shows the next hop to reach the network, which is the IP address of the neighbor router r4.

```
show ip bgp
```

```
frr-pc# show ip bgp
BGP table version is 1, local router ID is 192.168.34.2, vrf id 0
Default local pref 100, local AS 300
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop           Metric LocPrf Weight Path
*> [192.168.2.0/24]  [192.168.24.1]        0          0 200 i

Displayed 1 routes and 1 total paths
frr-pc#
```

Figure 48. Verifying the BGP table of router r4.

5. Redistribute routes on router r3 and router r4

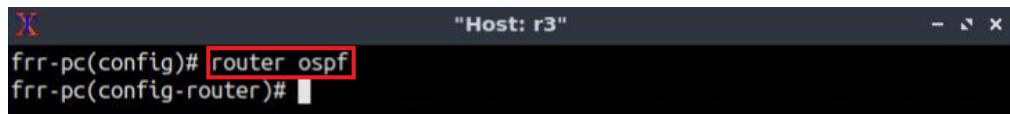
In this section, you will configure redistribution in routers running multiple routing protocols (OSPF, BGP), i.e., routers r3 and r4. At this point, routing protocols do not share their learned routes with each other. Thus, you will redistribute the routes of each routing protocol so that routers r3 and r4 share all the routes with each other. In section 5.1, you will redistribute BGP routes into OSPF routing protocol with a default metric. Then, in

section 5.2, OSPF and directly connected routes will be redistributed into BGP routing protocol.

5.1. Inject BGP routes into OSPF

Step 1. Router r3 received the network 192.168.1.0/24 through EBGP. By doing the redistribution, r3 will share the network with router r4 via OSPF. In this step, you will enable OSPF configuration mode so that you can redistribute the BGP route into OSPF. To enable OSPF configuration mode, type the following command in router r3 terminal:

```
router ospf
```



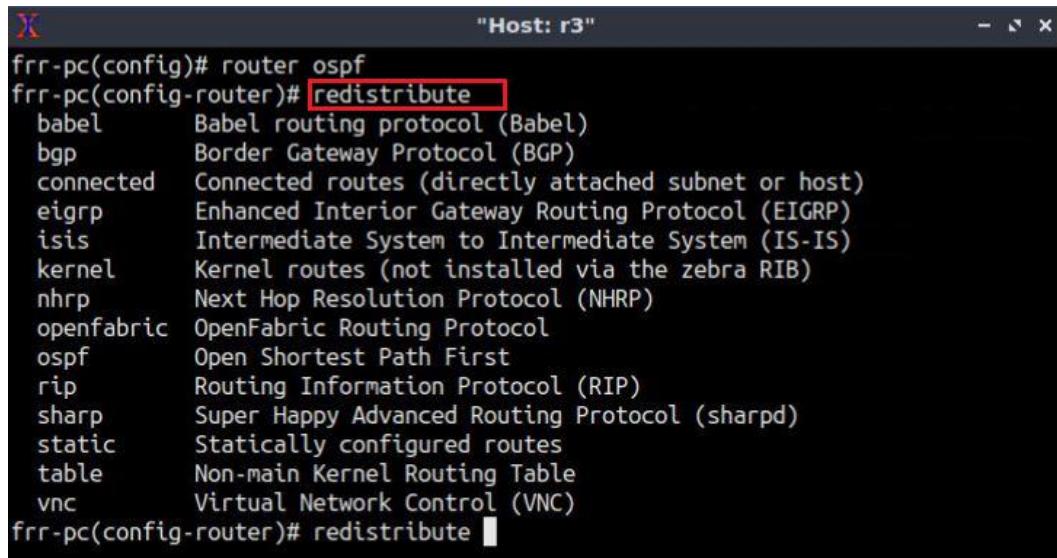
```
"Host: r3"
frr-pc(config)# router ospf
frr-pc(config-router)#

```

Figure 49. Enabling OSPF configuration.

Step 2. Type the command shown below to display all the options available for route redistribution. Then, the option `bgp` will be listed.

```
redistribute ?
```



```
"Host: r3"
frr-pc(config)# router ospf
frr-pc(config-router)# redistribute
babel      Babel routing protocol (Babel)
bgp        Border Gateway Protocol (BGP)
connected   Connected routes (directly attached subnet or host)
eigrp      Enhanced Interior Gateway Routing Protocol (EIGRP)
isis       Intermediate System to Intermediate System (IS-IS)
kernel     Kernel routes (not installed via the zebra RIB)
nhrp      Next Hop Resolution Protocol (NHRP)
openfabric  OpenFabric Routing Protocol
ospf       Open Shortest Path First
rip        Routing Information Protocol (RIP)
sharp      Super Happy Advanced Routing Protocol (sharpd)
static     Statically configured routes
table      Non-main Kernel Routing Table
vnc       Virtual Network Control (VNC)
frr-pc(config-router)# redistribute

```

Figure 50. Listing all the redistribution options.

Notice that the character `[?]` will not be displayed in the command prompt, it will display a list of the commands you can use after the word *redistribution* instead.

Step 3. To list the BGP options, type the command.

```
bgp ?
```

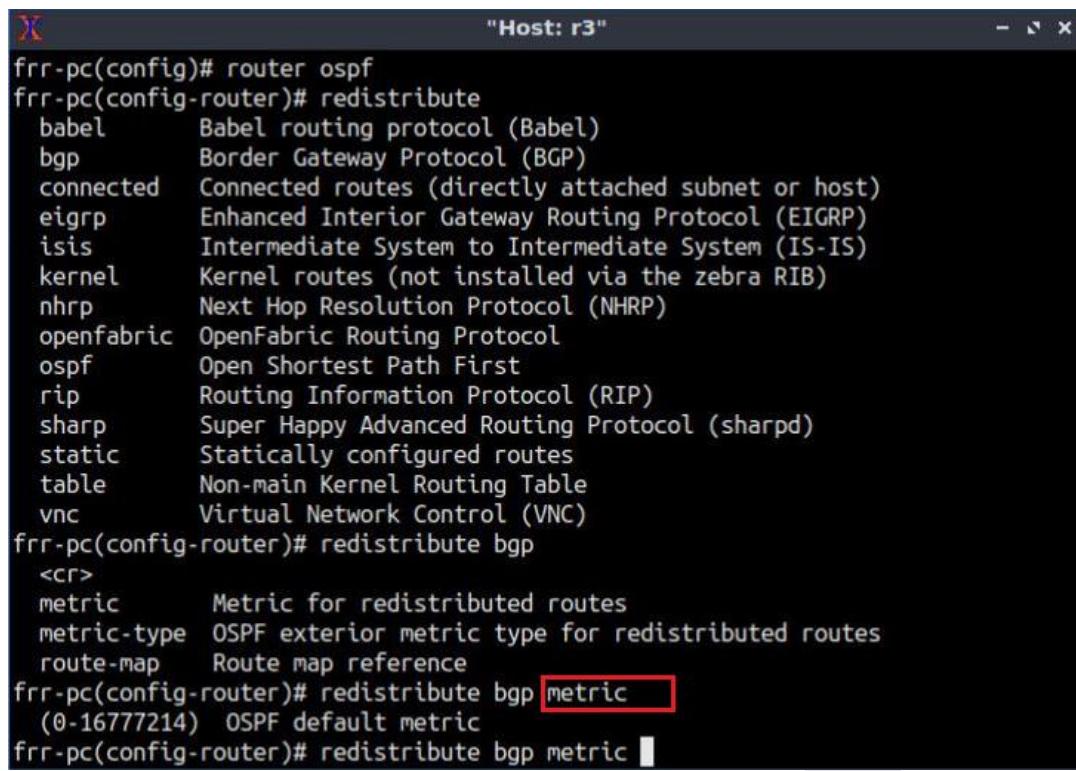
```
frr-pc(config)# router ospf
frr-pc(config-router)# redistribute
  babel      Babel routing protocol (Babel)
  bgp        Border Gateway Protocol (BGP)
  connected  Connected routes (directly attached subnet or host)
  eigrp      Enhanced Interior Gateway Routing Protocol (EIGRP)
  isis       Intermediate System to Intermediate System (IS-IS)
  kernel     Kernel routes (not installed via the zebra RIB)
  nhrp       Next Hop Resolution Protocol (NHRP)
  openfabric OpenFabric Routing Protocol
  ospf       Open Shortest Path First
  rip        Routing Information Protocol (RIP)
  sharp      Super Happy Advanced Routing Protocol (sharpd)
  static     Statically configured routes
  table      Non-main Kernel Routing Table
  vnc        Virtual Network Control (VNC)
frr-pc(config-router)# redistribute bgp
<cr>
  metric     Metric for redistributed routes
  metric-type OSPF exterior metric type for redistributed routes
  route-map  Route map reference
frr-pc(config-router)# redistribute bgp
```

Figure 51. Listing the options available for `bgp`.

Notice that the `bgp ?` will not be displayed in the command prompt instead, you will get a list of options that you can use with `bgp` command.

Step 4. Type the following command to configure the default metric. Notice that the character `?` will not be displayed in the command prompt instead, you will get the range of number that `metric` can adopt. You can choose any number within the range in order to configure the default metric.

```
metric ?
```



```
"Host: r3"
frr-pc(config)# router ospf
frr-pc(config-router)# redistribute
  babel      Babel routing protocol (Babel)
  bgp        Border Gateway Protocol (BGP)
  connected  Connected routes (directly attached subnet or host)
  eigrp      Enhanced Interior Gateway Routing Protocol (EIGRP)
  isis       Intermediate System to Intermediate System (IS-IS)
  kernel     Kernel routes (not installed via the zebra RIB)
  nhrp       Next Hop Resolution Protocol (NHRP)
  openfabric OpenFabric Routing Protocol
  ospf       Open Shortest Path First
  rip        Routing Information Protocol (RIP)
  sharp      Super Happy Advanced Routing Protocol (sharpd)
  static     Statically configured routes
  table      Non-main Kernel Routing Table
  vnc        Virtual Network Control (VNC)
frr-pc(config-router)# redistribute bgp
<cr>
  metric      Metric for redistributed routes
  metric-type OSPF exterior metric type for redistributed routes
  route-map   Route map reference
frr-pc(config-router)# redistribute bgp metric metric
  (0-16777214) OSPF default metric
frr-pc(config-router)# redistribute bgp metric █
```

Figure 52. Showing the range of values available for `metric`.

Step 5. In order to redistribute BGP routes, a specific metric is required. For the purpose of this lab, you will specify the metric 12. Type the following command to assign a BGP metric.

12

```

frr-pc(config-router)# redistribute
  babel      Babel routing protocol (Babel)
  bgp       Border Gateway Protocol (BGP)
  connected Connected routes (directly attached subnet or host)
  eigrp     Enhanced Interior Gateway Routing Protocol (EIGRP)
  isis      Intermediate System to Intermediate System (IS-IS)
  kernel    Kernel routes (not installed via the zebra RIB)
  nhrp     Next Hop Resolution Protocol (NHRP)
  openfabric OpenFabric Routing Protocol
  ospf      Open Shortest Path First
  rip       Routing Information Protocol (RIP)
  sharp     Super Happy Advanced Routing Protocol (sharpd)
  static    Statically configured routes
  table    Non-main Kernel Routing Table
  vnc      Virtual Network Control (VNC)
frr-pc(config-router)# redistribute bgp
<cr>
  metric    Metric for redistributed routes
  metric-type OSPF exterior metric type for redistributed routes
  route-map Route map reference
frr-pc(config-router)# redistribute bgp metric
(0-16777214) OSPF default metric
frr-pc(config-router)# redistribute bgp metric 12
frr-pc(config-router)#

```

Figure 53. Setting the metric number to redistribute BGP routes.

Step 6. At this point, you injected BGP routes into OSPF routing protocol. To proceed, type the following command to exit from configuration mode.

```

end

frr-pc(config-router)# redistribute bgp
<cr>
  metric    Metric for redistributed routes
  metric-type OSPF exterior metric type for redistributed routes
  route-map Route map reference
frr-pc(config-router)# redistribute bgp metric
(0-16777214) OSPF default metric
frr-pc(config-router)# redistribute bgp metric 12
frr-pc(config-router)# end
frr-pc#

```

Figure 54. Exiting from configuration mode.

Step 7. In router r4 terminal, type the following command to enable the configuration mode.

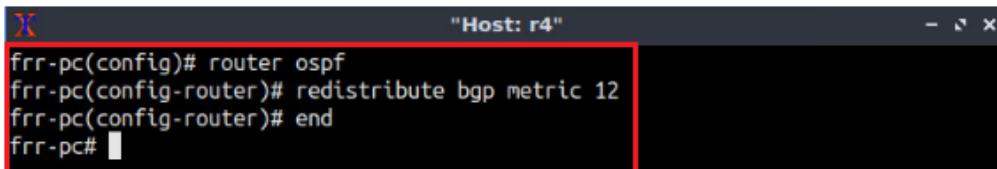
```
configure terminal
```



```
"Host: r4"
frr-pc# configure terminal
frr-pc(config)#
```

Figure 55. Enabling configuration mode on router r4.

Step 8. Router r4 received the network 192.168.2.0/24 through EBGP. By doing the redistribution, r4 will be sharing the network with router r3 via OSPF. In order to redistribute BGP routes in router r4, repeat from step 1 to step 6. All the commands are summarized in the figure below.

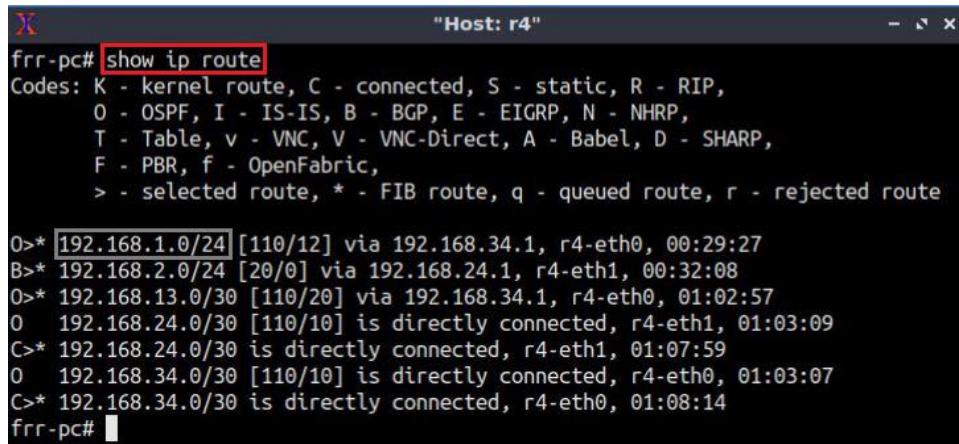


```
"Host: r4"
frr-pc(config)# router ospf
frr-pc(config-router)# redistribute bgp metric 12
frr-pc(config-router)# end
frr-pc#
```

Figure 56. Redistributing BGP routes on router r4.

Step 9. In order to verify the routing table of router r4, type the following command. You will verify that the network 192.168.1.0/24 is added to the routing table of router r4. Additionally, this network is reachable via the IP address 192.168.34.1 using OSPF routing protocol.

```
show ip route
```



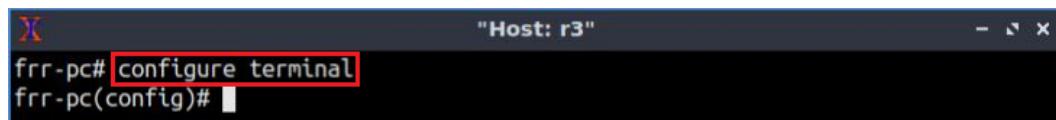
```
"Host: r4"
frr-pc# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued route, r - rejected route
O>* [192.168.1.0/24] [110/12] via 192.168.34.1, r4-eth0, 00:29:27
B>* 192.168.2.0/24 [20/0] via 192.168.24.1, r4-eth1, 00:32:08
O>* 192.168.13.0/30 [110/20] via 192.168.34.1, r4-eth0, 01:02:57
O  192.168.24.0/30 [110/10] is directly connected, r4-eth1, 01:03:09
C>* 192.168.24.0/30 is directly connected, r4-eth1, 01:07:59
O  192.168.34.0/30 [110/10] is directly connected, r4-eth0, 01:03:07
C>* 192.168.34.0/30 is directly connected, r4-eth0, 01:08:14
frr-pc#
```

Figure 57. Verifying routing table of router r4.

5.2. Inject OSPF and directly connected routes into BGP

Step 1. To enable router r3 configuration mode, issue the following command:

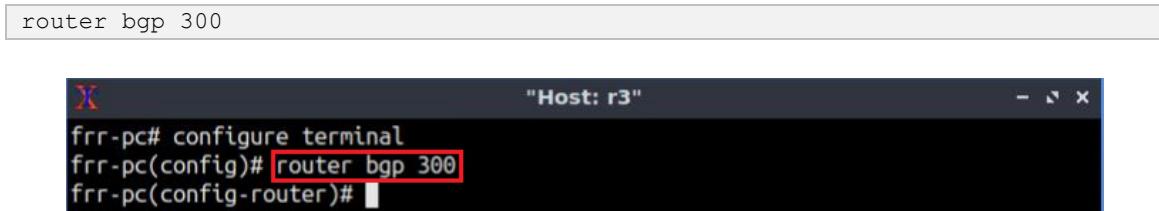
```
configure terminal
```



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

Figure 58. Enabling configuration mode on router r3.

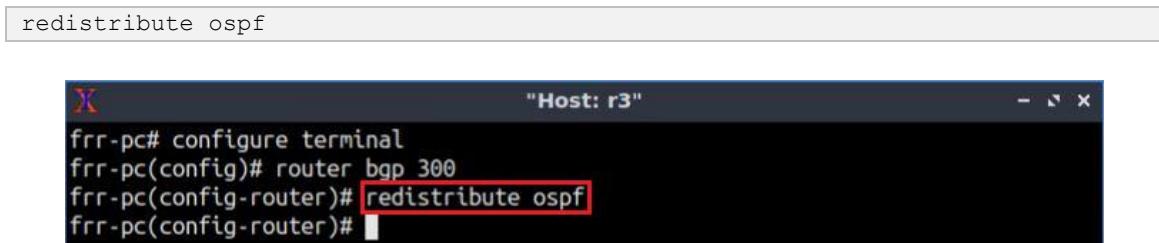
Step 2. In router r3, you will redistribute the OSPF routes (192.168.2.0/24 and 192.168.24.0/30) into BGP so that router r1 can reach the networks. Additionally, you will redistribute the directly connected routes (192.168.13.0/30, 192.168.34.0/30) so that router r1 can learn the paths to reach the networks. Type the following command to enter BGP configuration mode.



```
router bgp 300  
  
frr-pc# configure terminal  
frr-pc(config)# router bgp 300  
frr-pc(config-router)#
```

Figure 59. Entering to BGP Configuration mode.

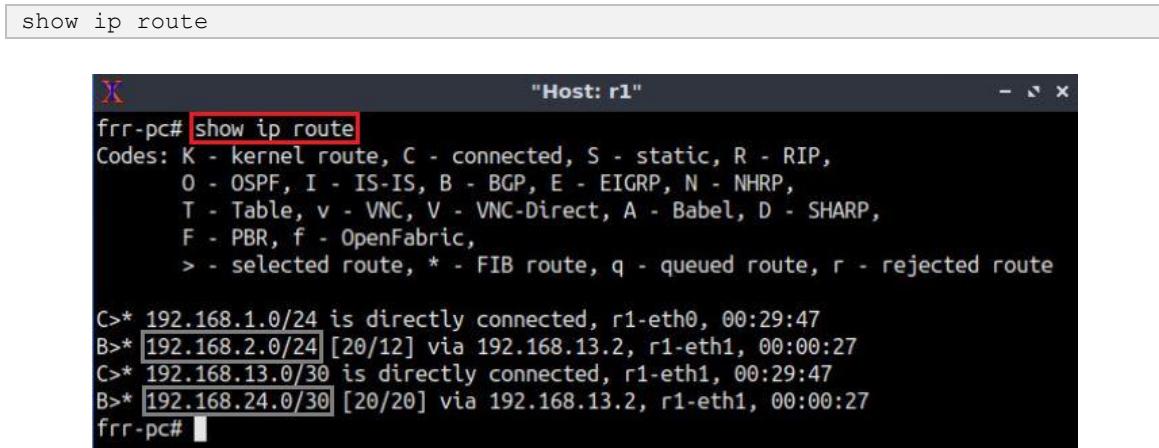
Step 3. Type the following command to redistribute all the OSPF networks.



```
redistribute ospf  
  
frr-pc# configure terminal  
frr-pc(config)# router bgp 300  
frr-pc(config-router)# redistribute ospf  
frr-pc(config-router)#
```

Figure 60. Redistributing OSPF routes on router r3.

Step 4. Type the following command to verify the routing table of router r1. You will notice new networks shared by router r3 (192.168.2.0/24 and 192.168.24.0/30). It also shows that router r1 can reach these networks via the IP address 192.168.13.2 using EBGP.



```
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-eth0, 00:29:47  
B>* 192.168.2.0/24 [20/12] via 192.168.13.2, r1-eth1, 00:00:27  
C>* 192.168.13.0/30 is directly connected, r1-eth1, 00:29:47  
B>* 192.168.24.0/30 [20/20] via 192.168.13.2, r1-eth1, 00:00:27  
frr-pc#
```

Figure 61. Verifying routing table of router r1.

Step 5. In router r3 terminal, type the following command to redistribute the directly connected networks (192.168.13.0/30, 192.168.34.0/30) into BGP.

```
redistribute connected
```

```
frr-pc# configure terminal
frr-pc(config)# router bgp 300
frr-pc(config-router)# redistribute ospf
frr-pc(config-router)# redistribute connected
frr-pc(config-router)#
```

Figure 62. Redistributing connected networks.

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

```
end
```

```
frr-pc# configure terminal
frr-pc(config)# router bgp 300
frr-pc(config-router)# redistribute ospf
frr-pc(config-router)# redistribute connected
frr-pc(config-router)# end
frr-pc#
```

Figure 63. Exiting from configuration mode.

Step 7. Type the following command to verify the routing table of router r1. You will verify additional network in router r1 table. These networks (192.168.13.0/30 and 192.168.34.0/30) are shared by router r3. It is also listed that router r1 can reach these networks via 192.168.13.2 using EBGP.

```
show ip route
```

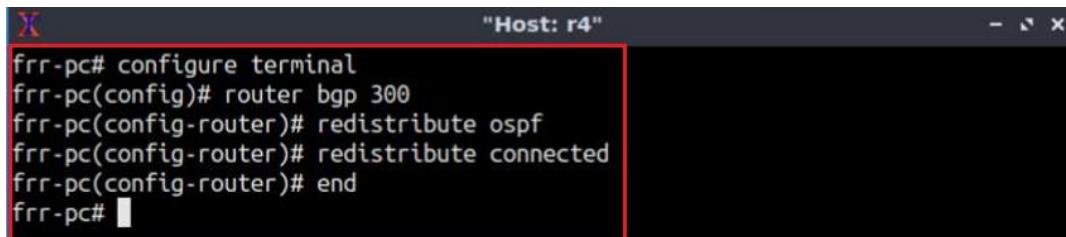
```
frr-pc# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       0 - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued route,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

C>* 192.168.1.0/24 is directly connected, r1-eth0, 00:33:59
B>* 192.168.2.0/24 [20/12] via 192.168.13.2, r1-eth1, 00:04:39
B [192.168.13.0/30] [20/0] via 192.168.13.2 inactive, 00:00:09
C>* 192.168.13.0/30 is directly connected, r1-eth1, 00:33:59
B>* 192.168.24.0/30 [20/20] via 192.168.13.2, r1-eth1, 00:04:39
B>* [192.168.34.0/30] [20/0] via 192.168.13.2, r1-eth1, 00:00:09
frr-pc#
```

Figure 64. Verifying routing table of router r1.

Step 8. In router r4 terminal, type the following command to redistribute the OSPF routes. This directive will allow router r2 to receive routing information from router r4. These routes carry the information about how router r4 reaches the networks 192.168.1.0/24

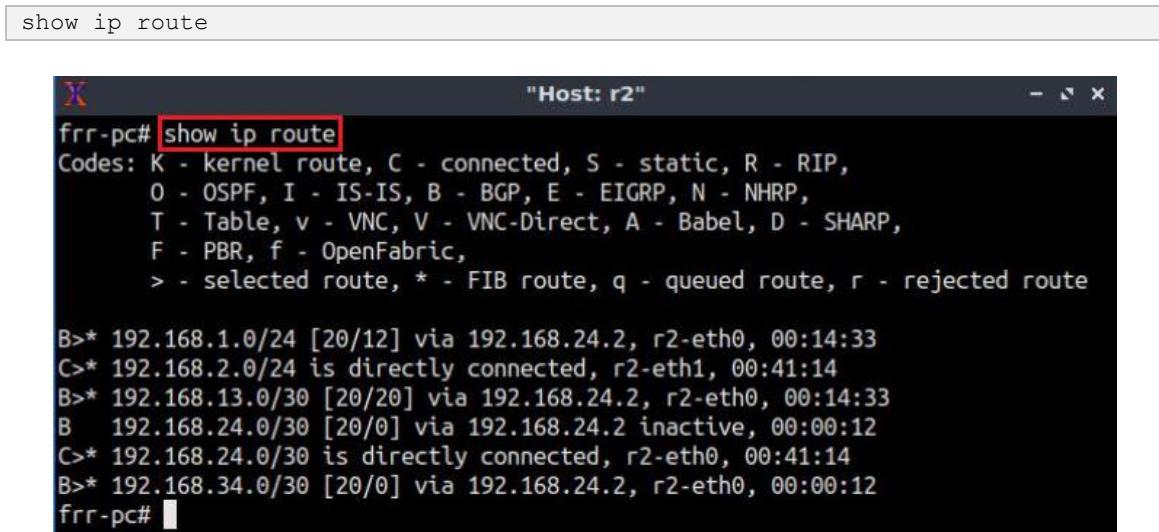
and 192.168.13.0/30. In addition, you will redistribute the information about how to reach the networks 192.168.24.0/30 and 192.168.34.0/30. The latter are the directly connected networks.



```
frr-pc# configure terminal
frr-pc(config)# router bgp 300
frr-pc(config-router)# redistribute ospf
frr-pc(config-router)# redistribute connected
frr-pc(config-router)# end
frr-pc#
```

Figure 65. Redistributing OSPF routes on router r4.

Step 9. Type the following command to verify the routing table of router r2. You will see the networks advertised through OSPF (192.168.1.0/24 and 192.168.13.0/30) and the directly connected networks with router r2 (192.168.24.0/30 and 192.168.34.0/30).



```
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

B>* 192.168.1.0/24 [20/12] via 192.168.24.2, r2-eth0, 00:14:33
C>* 192.168.2.0/24 is directly connected, r2-eth1, 00:41:14
B>* 192.168.13.0/30 [20/20] via 192.168.24.2, r2-eth0, 00:14:33
B   192.168.24.0/30 [20/0] via 192.168.24.2 inactive, 00:00:12
C>* 192.168.24.0/30 is directly connected, r2-eth0, 00:41:14
B>* 192.168.34.0/30 [20/0] via 192.168.24.2, r2-eth0, 00:00:12
frr-pc#
```

Figure 66. Verifying the routing table of router r2.

6. Verify connections

In this section, you will verify if the configuration is working correctly. You will also verify that Campus-1 and Campus-2 have properly formed an EBGP adjacency with the ISP.

Step 1. In the lab topology, hold right-click on host h1 and select *Terminal*. This opens the terminal of host h1.

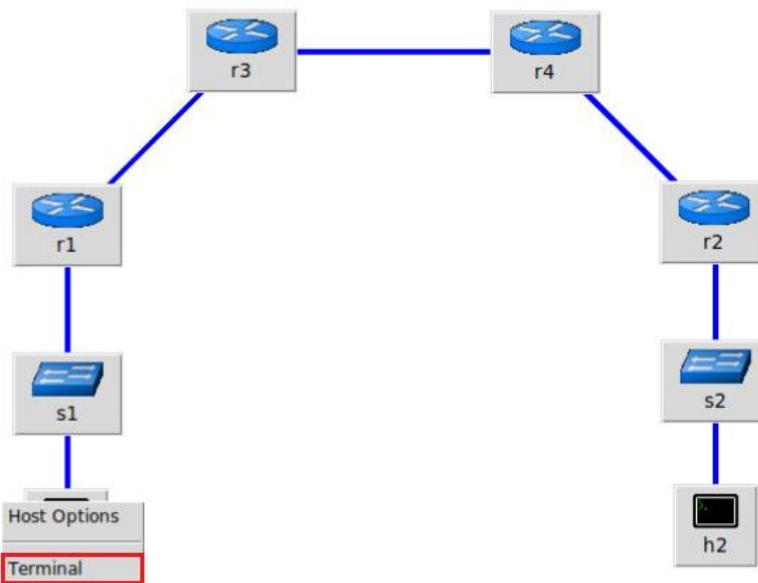


Figure 67. Opening host h1 terminal.

Step 2. Test the connectivity between host h1 and host h2 using the `ping` command. In host h1, type the command specified below. To stop the test, press `Ctrl+d`. The figure below shows a successful connectivity test.

```
ping 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=60 time=1.03 ms
64 bytes from 192.168.2.10: icmp_seq=2 ttl=60 time=0.118 ms
64 bytes from 192.168.2.10: icmp_seq=3 ttl=60 time=0.103 ms
64 bytes from 192.168.2.10: icmp_seq=4 ttl=60 time=0.102 ms
^C
--- 192.168.2.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 31ms
rtt min/avg/max/mdev = 0.102/0.338/1.031/0.400 ms
root@frr-pc:~#
```

Figure 68. Connectivity test using `ping` command.

Step 3. Similarly, hold right-click on host h2 and select *Terminal*. This opens the terminal of host h2.

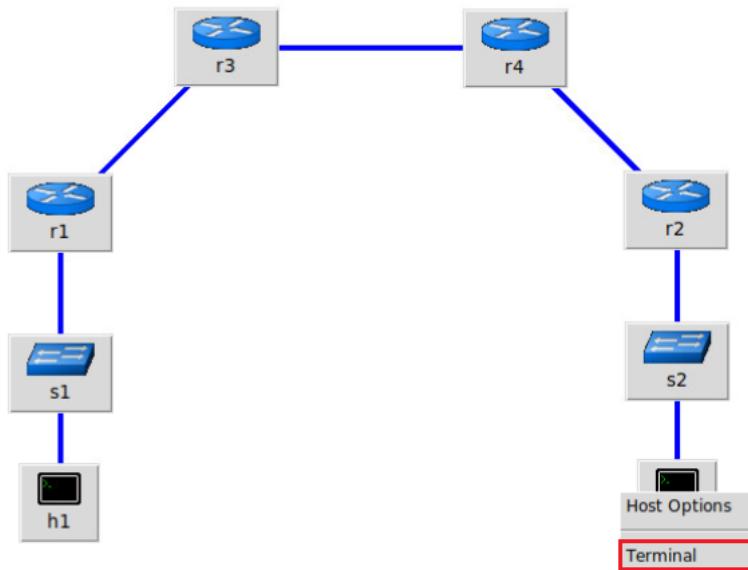


Figure 69. Opening host h2 terminal.

Step 4. Test the connectivity between host h2 and host h1 using the `ping` command. On host h2, type the command specified below. To stop the test, press `Ctrl+d`. The figure below shows a successful connectivity test.

```
ping 192.168.1.10
```

```
X "Host: h2"
root@frr-pc:~# ping 192.168.1.10
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_seq=1 ttl=60 time=0.136 ms
64 bytes from 192.168.1.10: icmp_seq=2 ttl=60 time=0.110 ms
64 bytes from 192.168.1.10: icmp_seq=3 ttl=60 time=0.115 ms
64 bytes from 192.168.1.10: icmp_seq=4 ttl=60 time=0.105 ms
64 bytes from 192.168.1.10: icmp_seq=5 ttl=60 time=0.102 ms
64 bytes from 192.168.1.10: icmp_seq=6 ttl=60 time=0.089 ms
^C
--- 192.168.1.10 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 113ms
rtt min/avg/max/mdev = 0.089/0.109/0.136/0.017 ms
root@frr-pc:~#
```

Figure 70. Connectivity test using `ping` command.

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

References

1. A. Tanenbaum, D. Wetherall, “Computer networks”, 5th Edition, Pearson, 2012.
2. Cisco, “What are OSPF Areas and virtual links?”, 2016. [Online]. Available: <https://www.cisco.com/c/en/us/support/docs/ip/open-shortest-path-first-ospf/13703-8.html>
3. Cisco, “Redistributing routing protocols”, 2012. [Online]. Available: <https://www.cisco.com/c/en/us/support/docs/ip/enhanced-interior-gateway-routing-protocol-eigrp/8606-redist.html>

4. Linux foundation collaborative projects, “FRR routing documentation”, 2017. [Online]. Available: <http://docs.frrouting.org/en/latest/>
5. D. Teare, B. Vachon, R. Graziani, “Implementing cisco IP Routing (Route), foundation learning guide”, CCNP ROUTE 300-101.



BORDER GATEWAY PROTOCOL

Lab 5: BGP Authentication

Document Version: **03-4-2020**



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 BGP overview	3
1.2 MD5 hash algorithm.....	4
1.3 BGP authentication	5
2 Lab topology.....	6
2.1 Lab settings.....	6
2.2 Open the topology and load the configuration	7
2.3 Load zebra daemon and verify configuration	10
3 Configure EBGP on the routers.....	14
4 Configure and verify MD5 authentication on the routers.....	19
4.1 Configure MD5 authentication	19
4.2 Verify MD5 authentication.....	21
References	22

Overview

This lab introduces Border Gateway Protocol (BGP) authentication that is used to safeguard routing sessions between peer routers. In this lab, External BGP (EBGP) will be configured and verified among three Autonomous Systems (ASes). Furthermore, Message Digest 5 (MD5) authentication will be configured on a Transmission Control Protocol (TCP) connection between BGP peers. In this lab, the terms BGP and EBGP will be used interchangeably since they will only be running between ASes.

Objectives

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

1. Understand the concept of EBGP.
2. Configure and verify BGP between two ASes.
3. Use MD5 authentication between BGP peers.
4. Enable authentication mechanism in networks running BGP.

Lab settings

The information in Table 1 provides the credentials to access Client1 machine.

Table 1. Credentials to access Client1 machine.

Device	Account	Password
Client1	admin	password

Lab roadmap

This lab is organized as follows:

1. Section 1: Introduction.
2. Section 2: Lab topology.
3. Section 3: Configure EBGP on the routers.
4. Section 4: Configure and verify MD5 authentication on the routers.

1 Introduction

1.1 BGP overview

BGP is an exterior gateway protocol designed to exchange routing and reachability information among ASes on the Internet. BGP is relevant to network administrators of large organizations which connect to one or more Internet Service Providers (ISPs), as well as to ISPs who connect to other network providers. In terms of BGP, an AS is referred to as a routing domain, where all networked systems operate common routing protocols and are under the control of a single administration¹.

BGP is a form of distance vector protocol. It requires each router to maintain a table, which stores the distance and the output interface (i.e., vector) to remote networks. BGP makes routing decisions based on paths, network policies, or rule set configured by a network administrator and is involved in making core routing decisions¹.

Two routers that establish a BGP connection are referred to as BGP peers or neighbors. BGP sessions run over TCP. If a BGP session is established between two neighbors in different ASes, the session is referred to as an EBGP session¹. Figure 2 shows a network running BGP protocol (EBGP) between three routers in different ASes.

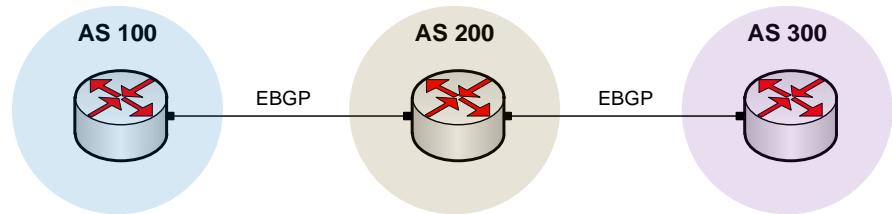


Figure 1. Routers in different ASes run EBGP to advertise routing information.

1.2 TCP MD5 authentication

MD5 hashing algorithm is a cryptographic function that takes as input a message of arbitrary length and produces as output a 128-bit message digest of that input. It is computationally hard to produce two messages that have the same message digest, or to produce a message from a given message digest. The MD5 hash algorithm is a widely used mechanism to secure TCP connection using a shared secret key between each end².

Consider Figure 2. The TCP segment generated by the sender contains the message and its digest encrypted with a shared secret key. When the receiver receives the TCP segment, it will calculate the digest (hash) of the message in the same way the sender did, decrypt the received digest using the shared secret key, and compare these two values. If the digest calculated by the receiver does not match the one sent by the sender, the session drops the segment.

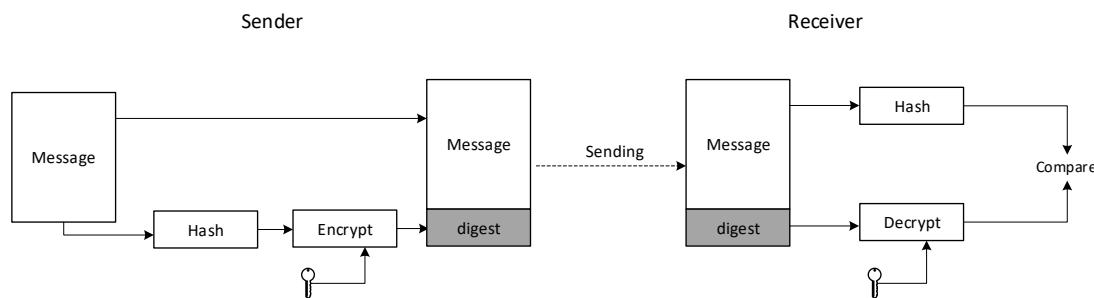


Figure 2. MD5 hash algorithm.

1.3 BGP authentication

BGP authentication enables the routers to share information only if they can verify that they are talking to a trusted source, based on a password (key). TCP MD5 authentication between BGP peers verifies each transmitted message sent via the BGP session. During an authenticated BGP session, BGP peers must be configured with the same password to establish BGP neighbor relationship³.

Consider Figure 3. routers that are configured with the same password establish BGP neighbor relationship (see Figure 3a), whereas, routers that are configured with different password will not be able to maintain BGP neighbor relationship (see Figure 3b).

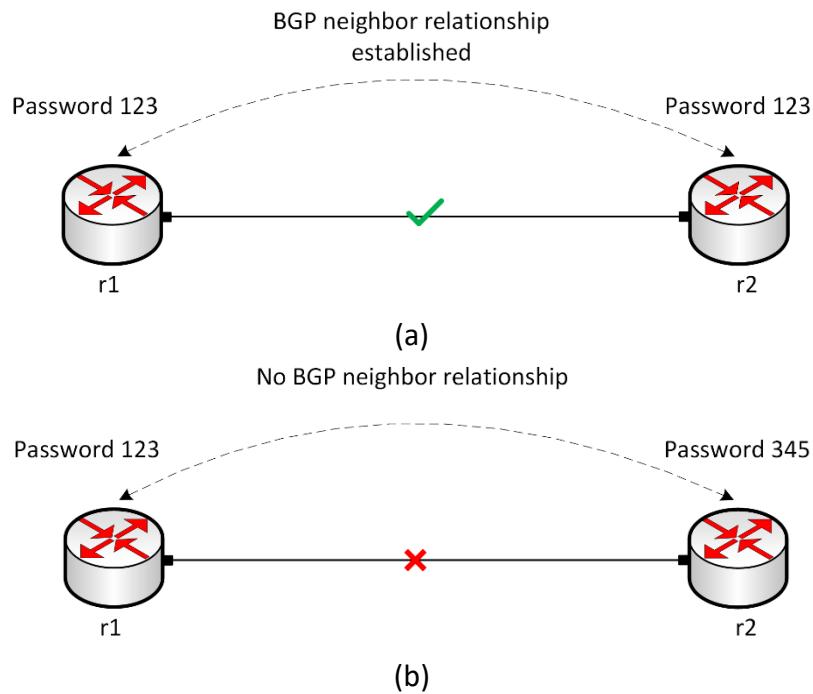


Figure 3.a and 3.b. BGP authentication.

2 Lab topology

Consider Figure 4. The lab topology consists of three networks, Network 1, Network 2, and Network 3 that lie within AS 100, AS 200, and AS 300, respectively. An MD5 authentication system has been used to authenticate BGP peer relationship. This allows the routers to exchange routing information via EBGP with validated peers only.

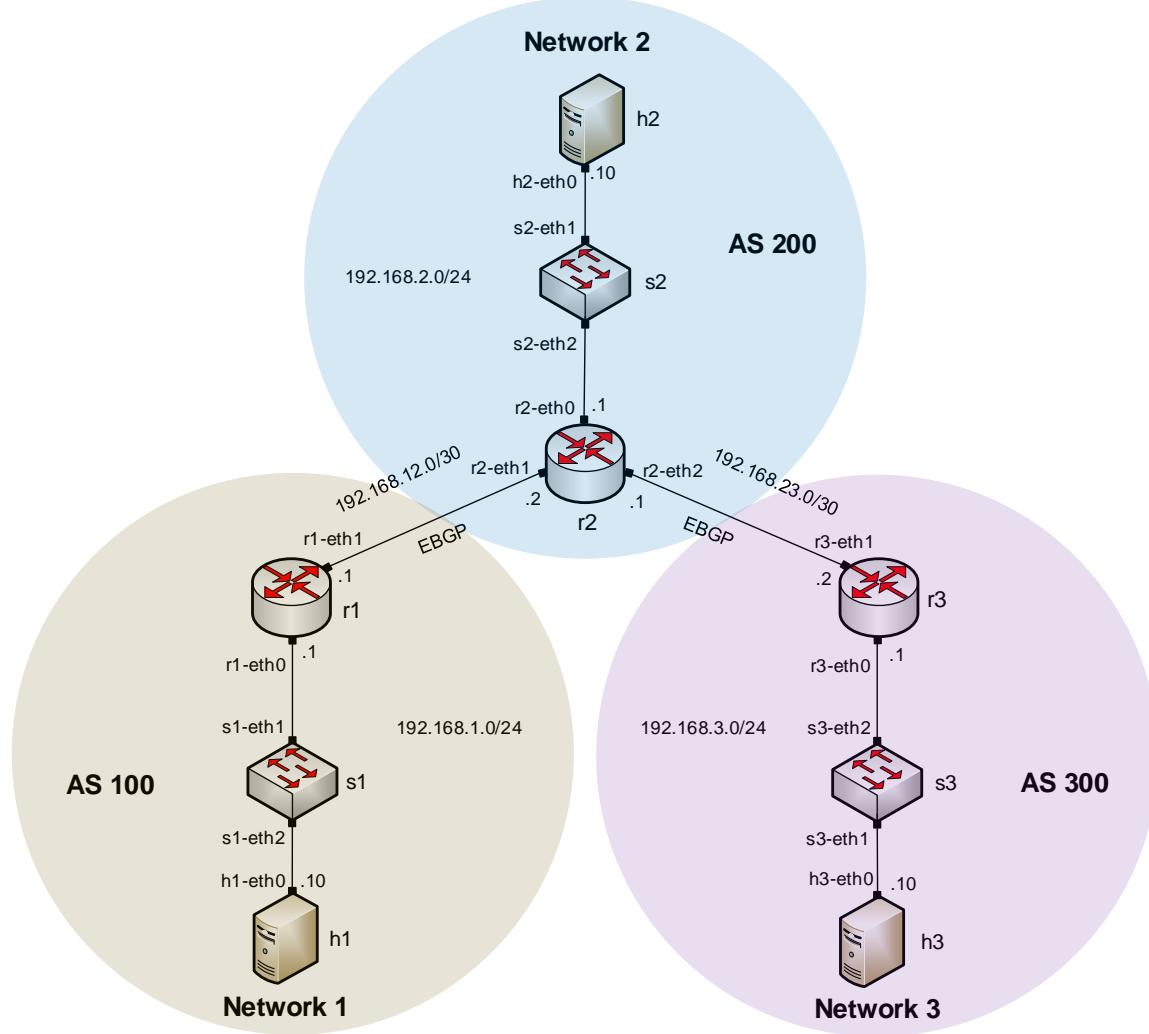


Figure 4. Lab topology.

2.1 Lab settings

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

Table 2. Topology information.

Device	Interface	IPv4 Address	Subnet	Default gateway
r1 (Network 1)	r1-eth0	192.168.1.1	/24	N/A
	r1-eth1	192.168.12.1	/30	N/A

r2 (Network 2)	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
r3 (Network 3)	r3-eth0	192.168.3.1	/24	N/A
	r3-eth1	192.168.23.2	/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
h3	h3-eth0	192.168.3.10	/24	192.168.3.1

2.2 Open the topology and load the configuration

Step 1. Start by launching Miniedit by clicking on Desktop's shortcut. When prompted for a password, type `password`.



Figure 5. MiniEdit shortcut.

Step 2. On Miniedit's menu bar, click on *File* then *open* to load the lab's topology. Locate the *Lab5.mn* topology file in the default directory, */home/frr/BGP_Labs/lab5* and click on *Open*.

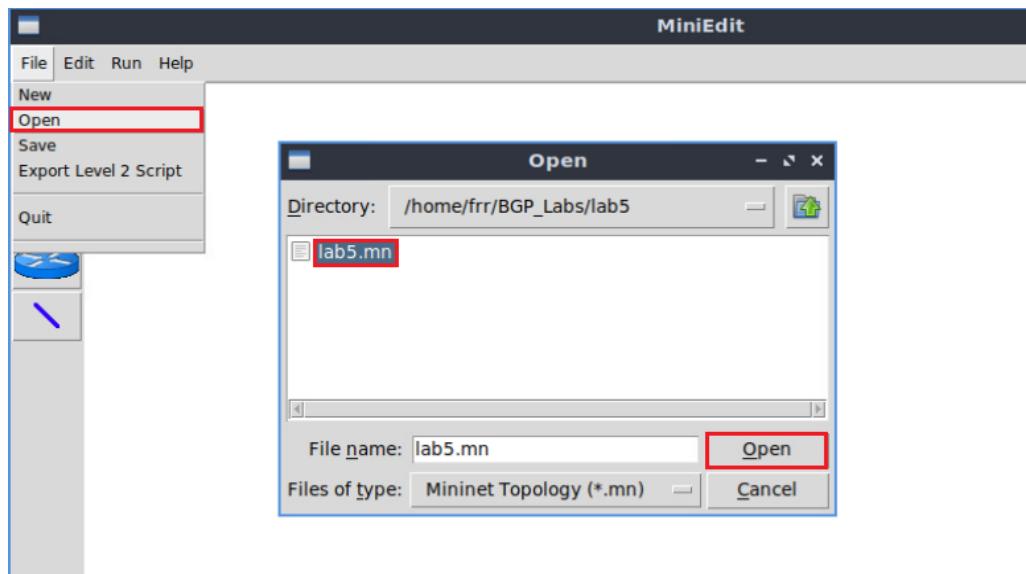


Figure 6. MiniEdit's Open dialog.

At this point the topology is loaded with all the required network components. You will execute a script that will load the configuration of the routers.

Step 3. Open the Linux terminal.

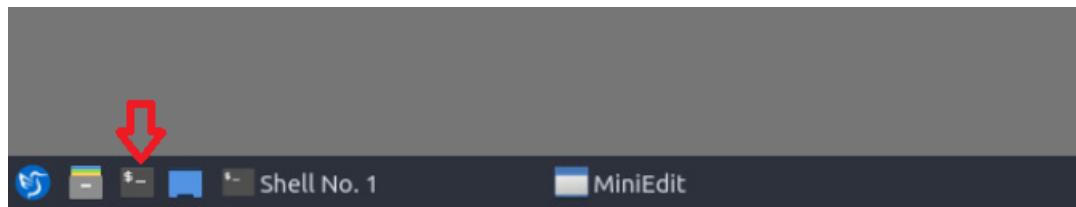


Figure 7. Opening Linux terminal.

Step 4. Click on the Linux's terminal and navigate into *BGP_Labs/lab5* 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.

A screenshot of a Linux terminal window. The window title is 'frr@frr-pc: ~/BGP_Labs/lab5'. The terminal prompt is 'frr@frr-pc:~\$'. The user has typed the command 'cd BGP_Labs/lab5' and is pressing the Enter key. The terminal window has a dark background with light-colored text.

Figure 8. Entering the BGP_Labs/lab5 directory.

Step 5. To execute the shell script, type the following command. The argument of the program corresponds to the configuration zip file that will be loaded in all the routers in the topology.

```
./config_loader.sh lab5_conf.zip
```

A screenshot of a terminal window titled "frr@frr-pc: ~/BGP_Labs/lab5". The window shows the command `./config_loader.sh lab5_conf.zip` being typed at the prompt. The entire command line is highlighted with a red box.

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

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

```
exit
```

A screenshot of a terminal window titled "frr@frr-pc: ~/BGP_Labs/lab5". The window shows the command `exit` being typed at the prompt. The entire command line is highlighted with a red box.

Figure 10. Exiting from the terminal.

Step 7. At this point hosts h1, h2 and h3 interfaces are configured. To proceed with the emulation, click on the *Run* button located in lower left-hand side.



Figure 11. Starting the emulation.

Step 8. Click on Mininet's terminal, i.e., the one launched when MiniEdit was started.

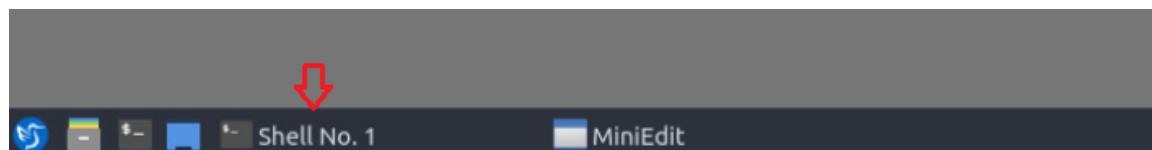


Figure 12. Opening Mininet's terminal.

Step 9. Issue the following command to display the interface names and connections.

```
links
```

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

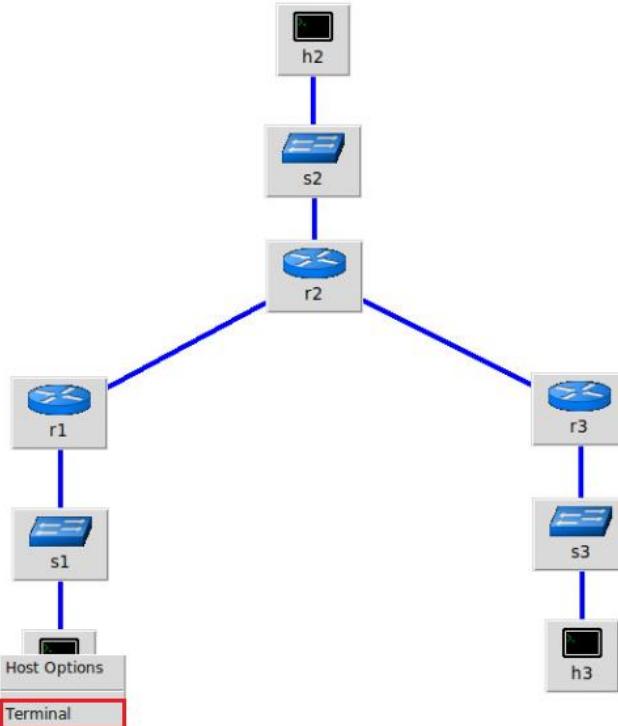
Figure 13. Displaying network interfaces.

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

2.3 Load zebra daemon and verify configuration

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

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

Figure 14. Opening a terminal on host *h1*.

Step 2. On 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.

```
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 15. Output of `ifconfig` command.

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

```
route
```

```
"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 16. Output of `route` command.

Step 4. In order to verify hosts h2 and h3, proceed similarly by repeating from step 1 to step 3 on hosts h2 and h3 terminals. Similar results should be observed.

Step 5. You will validate that the router interfaces are configured correctly according to Table 2. In order to verify router r1, hold right-click on router r1 and select Terminal.

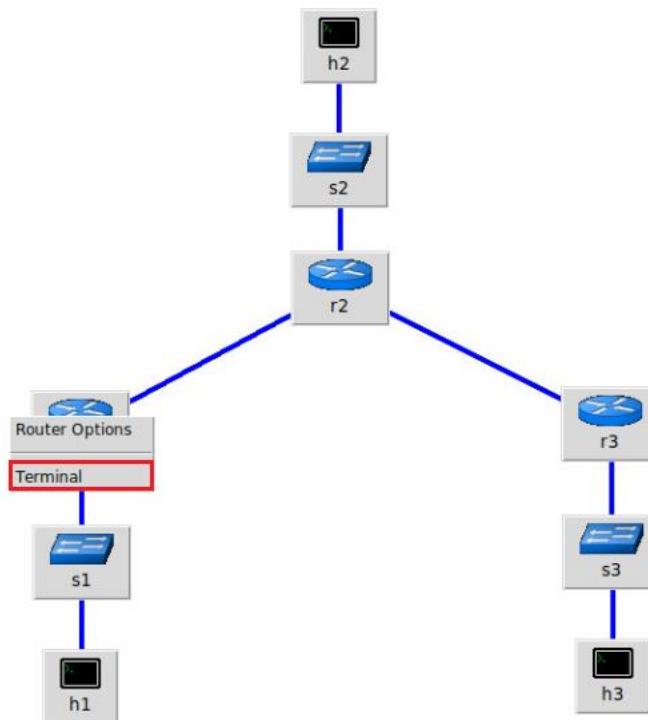


Figure 17. Opening a terminal on router r1.

Step 6. In this step, you will start zebra daemon, which is a multi-server routing software that provides TCP/IP based routing protocols. The configuration will not be working if you do not enable zebra daemon initially. In order 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 18. Starting zebra daemon.

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

```
vtysh
```

```
"Host: r1"
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 19. 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 attached to routers r2 (192.168.2.0/24) and router r3 (192.168.3.0/24) 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, 00:00:17
C>* 192.168.12.0/30 is directly connected, r1-eth1, 00:00:17
frr-pc#
```

Figure 20. Displaying routing table of router r1.

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 networks directly connected networks of router r2.

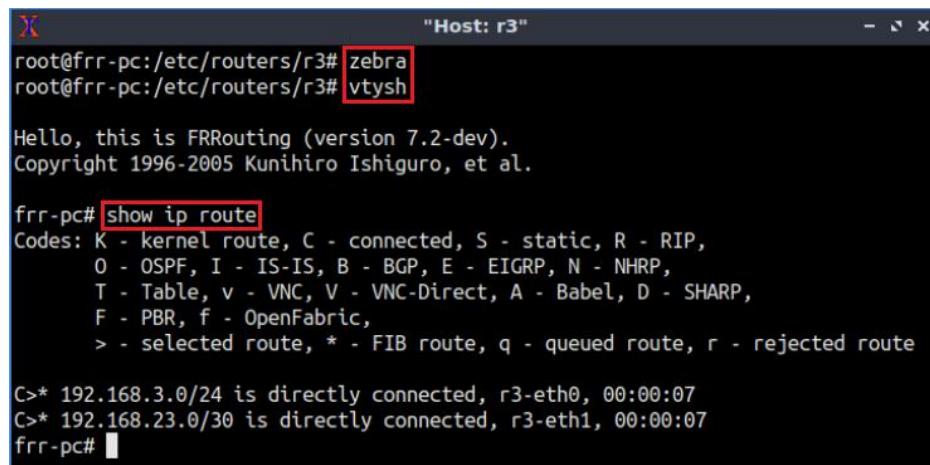
```
"Host: r2"
root@frr-pc:/etc/routers/r2# zebra
root@frr-pc:/etc/routers/r2# vtysh
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

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, 00:00:07
C>* 192.168.12.0/30 is directly connected, r2-eth1, 00:00:07
C>* 192.168.23.0/30 is directly connected, r2-eth2, 00:00:07
frr-pc#
```

Figure 21. Displaying routing table of router r2.

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



The terminal window shows the following output:

```
"Host: r3"
root@frr-pc:/etc/routers/r3# zebra
root@frr-pc:/etc/routers/r3# 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

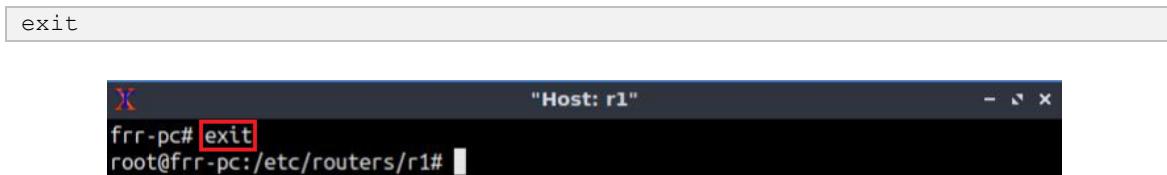
C>* 192.168.3.0/24 is directly connected, r3-eth0, 00:00:07
C>* 192.168.23.0/30 is directly connected, r3-eth1, 00:00:07
frr-pc#
```

Figure 22. Displaying routing table of router r3.

3 Configure EBGP on the routers

In this section, you will configure EBGP on the routers that are hosted in different ASes. You will assign BGP neighbors to allow the routers to exchange BGP routes. Furthermore, routers r1, r2, and r3 will advertise their Local Area Networks (LANs) via BGP so that the LANs are learned by peer routers.

Step 1. To configure BGP routing protocol, you need to enable the BGP daemon first. In router r1, type the following command to exit the vtysh session:



```
exit
```

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

Figure 23. Exiting the vtysh session.

Step 2. Type the following command on router r1 terminal to enable and to start BGP routing protocol.



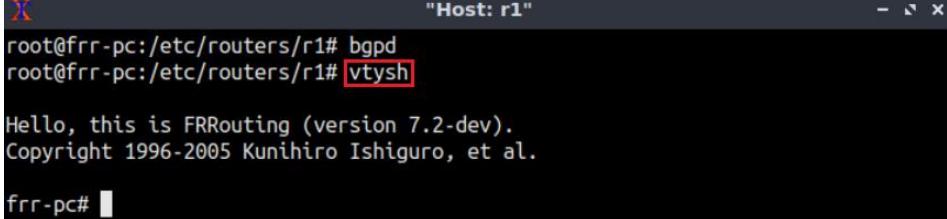
```
bgpd
```

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

Figure 24. Starting BGP daemon.

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

```
vtysh
```

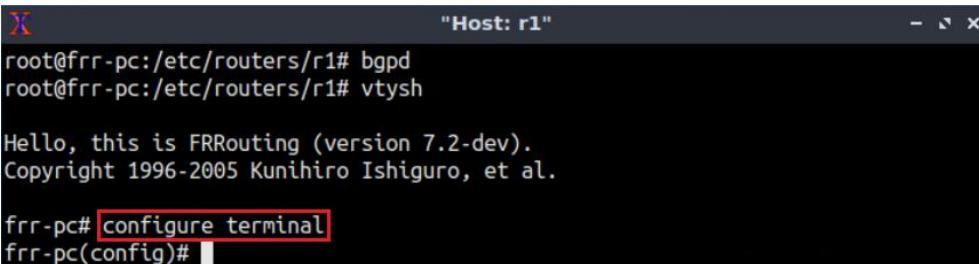


The terminal window shows the FRRouting version 7.2-dev. It starts with 'bgpd' and then enters 'vtysh'. The FRRouting copyright notice follows. The prompt 'frr-pc#' is shown at the bottom.

Figure 25. Starting vtysh on router r1.

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

```
configure terminal
```

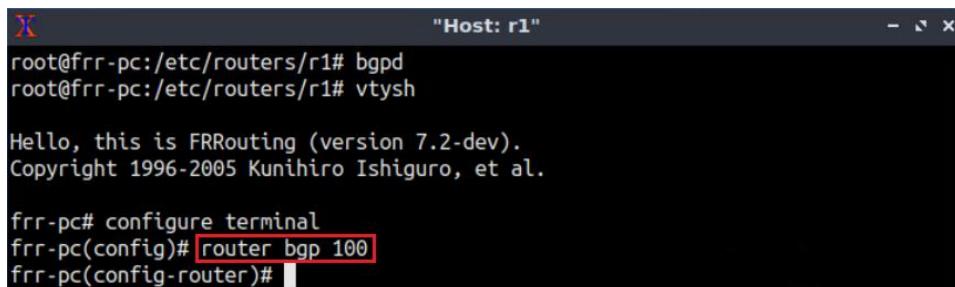


The terminal window shows the configuration mode command 'configure terminal' being entered. The FRRouting copyright notice follows. The prompt 'frr-pc(config)#' is shown at the bottom.

Figure 26. Enabling configuration mode on router r1.

Step 5. Router 1 is in AS 100. In order to configure BGP, type the following command:

```
router bgp 100
```

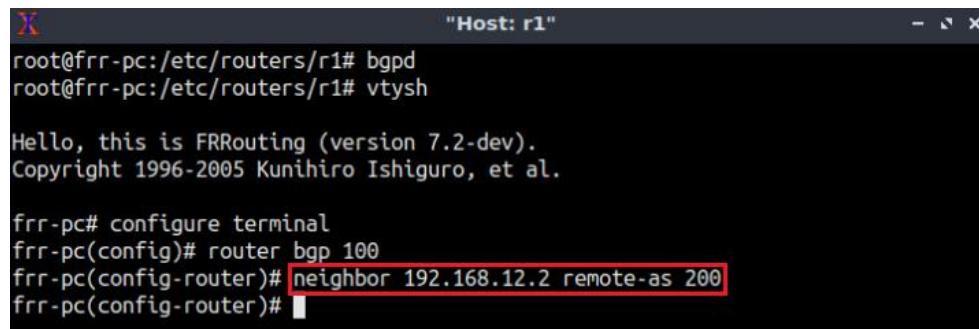


The terminal window shows the BGP configuration command 'router bgp 100' being entered. The FRRouting copyright notice follows. The prompt 'frr-pc(config-router)#' is shown at the bottom.

Figure 27. Configuring BGP on router r1.

Step 6. To configure a BGP neighbor to router r1 (AS 100), type the command shown below. This command specifies the neighbor IP address (192.168.12.2) and the AS number of the remote BGP peer (AS 200).

```
neighbor 192.168.12.2 remote-as 200
```



```
"Host: r1"
root@frrr-pc:/etc/routers/r1# bgpd
root@frrr-pc:/etc/routers/r1# vtysh

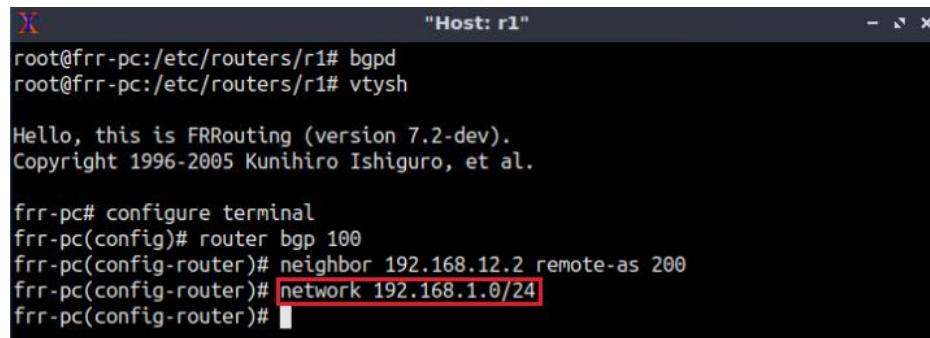
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frrr-pc# configure terminal
frrr-pc(config)# router bgp 100
frrr-pc(config-router)# neighbor 192.168.12.2 remote-as 200
frrr-pc(config-router)#[ ]
```

Figure 28. Assigning BGP neighbor to router r1.

Step 7. In this step, router r1 will advertise LAN 192.168.1.0/24 to its BGP peers. To do so, issue the following command:

```
network 192.168.1.0/24
```



```
"Host: r1"
root@frrr-pc:/etc/routers/r1# bgpd
root@frrr-pc:/etc/routers/r1# vtysh

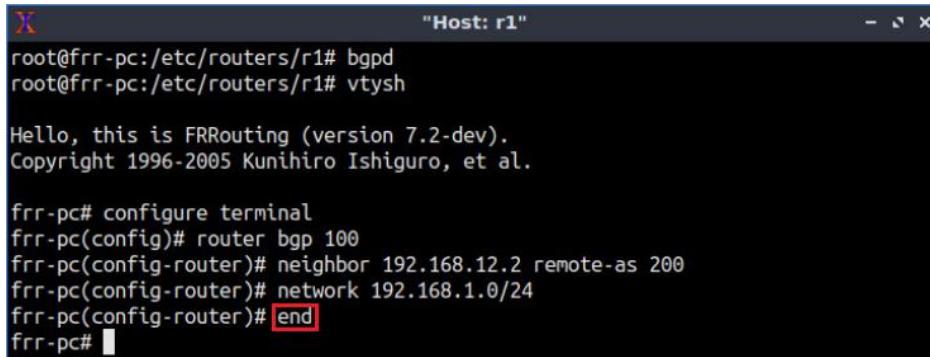
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frrr-pc# configure terminal
frrr-pc(config)# router bgp 100
frrr-pc(config-router)# neighbor 192.168.12.2 remote-as 200
frrr-pc(config-router)# network 192.168.1.0/24
frrr-pc(config-router)#[ ]
```

Figure 29. Advertising local network on router r1.

Step 8. Type the following command to exit from the configuration mode.

```
end
```



```
"Host: r1"
root@frrr-pc:/etc/routers/r1# bgpd
root@frrr-pc:/etc/routers/r1# vtysh

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

frrr-pc# configure terminal
frrr-pc(config)# router bgp 100
frrr-pc(config-router)# neighbor 192.168.12.2 remote-as 200
frrr-pc(config-router)# network 192.168.1.0/24
frrr-pc(config-router)# end
frrr-pc#[ ]
```

Figure 30. Exiting from configuration mode.

Step 9. Type the following command to verify BGP networks. You will observe the LAN network of router r1.

```
show ip bgp
```

```

frr-pc# show ip bgp
BGP table version is 1, local router ID is 192.168.12.1, vrf id 0
Default local pref 100, local AS 100
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*-> 192.168.1.0/24    0.0.0.0                  0        32768 i

Displayed 1 routes and 1 total paths
frr-pc#

```

Figure 31. Verifying BGP networks on router r1.

Step 10. Type the following command to verify BGP neighbors. You will verify that the neighbor IP address is 192.168.12.2. The corresponding AS number is 200.

```

show ip bgp neighbors

```

```

frr-pc# show ip bgp neighbors
BGP neighbor is [192.168.12.2], [remote AS 200], [local AS 100], [external link]
  BGP version 4, remote router ID 0.0.0.0, local router ID 192.168.12.1
  BGP state = Active
  Last read 00:04:15, Last write never
  Hold time is 180, keepalive interval is 60 seconds
  Message statistics:
    Inq depth is 0
    Outq depth is 0
              Sent      Rcvd
  Opens:          0          0
  Notifications: 0          0
  Updates:       0          0
  Keepalives:    0          0
  Route Refresh: 0          0
  Capability:   0          0
  Total:         0          0
  Minimum time between advertisement runs is 0 seconds

  For address family: IPv4 Unicast
    Not part of any update group
    Community attribute sent to this neighbor(all)
    0 accepted prefixes

```

Figure 32. Verifying BGP neighbors on router r1.

Step 11. Follow from step 1 to step 8 but with different metrics in order to configure BGP on router r2. All these steps are summarized in the following figure.

The terminal window shows the configuration of BGP on router r2. The user enters 'configure terminal' and configures BGP with AS 200, neighbors 192.168.12.1 and 192.168.23.2, and a network 192.168.2.0/24.

```
frr-pc# exit
root@frr-pc:/etc/routers/r2# bgpd
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 bgp 200
frr-pc(config-router)# neighbor 192.168.12.1 remote-as 100
frr-pc(config-router)# neighbor 192.168.23.2 remote-as 300
frr-pc(config-router)# network 192.168.2.0/24
frr-pc(config-router)# end
frr-pc#
```

Figure 33. Configuring BGP on router r2.

Step 12. Follow from step 1 to step 8 in order to configure BGP on router r3. All these steps are summarized in the following figure.

The terminal window shows the configuration of BGP on router r3. The user enters 'configure terminal' and configures BGP with AS 300, neighbors 192.168.23.1 and 192.168.3.0/24.

```
frr-pc# exit
root@frr-pc:/etc/routers/r3# bgpd
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 bgp 300
frr-pc(config-router)# neighbor 192.168.23.1 remote-as 200
frr-pc(config-router)# network 192.168.3.0/24
frr-pc(config-router)# end
frr-pc#
```

Figure 34. Configure BGP on router r3.

Step 13. In router r3 terminal, type the following command to verify the routing table of router r3. The LANs of router r1 (192.168.1.0/24) and router r2 (192.168.2.0/24) are advertised to router r3 through EBGP.

The terminal window shows the verification of the routing table on router r3. The user runs 'show ip route' and sees routes for 192.168.1.0/24 via r3-eth1 and 192.168.2.0/24 via r3-eth1, along with direct connections to 192.168.3.0/24 and 192.168.23.0/30.

```
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

B>* 192.168.1.0/24 [20/0] via 192.168.23.1, r3-eth1, 00:01:09
B>* 192.168.2.0/24 [20/0] via 192.168.23.1, r3-eth1, 00:01:09
C>* 192.168.3.0/24 is directly connected, r3-eth0, 00:05:22
C>* 192.168.23.0/30 is directly connected, r3-eth1, 00:05:22
frr-pc#
```

Figure 35. Verifying the routing table of router r3.

Step 14. On host h3 terminal, perform a connectivity test by running the command shown below. To stop the test, press **Ctrl+c**. The result will show a successful connectivity test.

```
ping 192.168.1.10
```

```
root@frr-pc:~# ping 192.168.1.10
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_seq=1 ttl=61 time=1.06 ms
64 bytes from 192.168.1.10: icmp_seq=2 ttl=61 time=0.090 ms
64 bytes from 192.168.1.10: icmp_seq=3 ttl=61 time=0.096 ms
64 bytes from 192.168.1.10: icmp_seq=4 ttl=61 time=0.092 ms
^C
--- 192.168.1.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 45ms
rtt min/avg/max/mdev = 0.090/0.333/1.056/0.417 ms
root@frr-pc:~#
```

Figure 36. Connectivity test using **ping** command.

4 Configure and verify MD5 authentication on the routers

In this section, you will employ MD5 algorithm to authenticate your BGP peer connection. You will configure BGP neighbor authentication on the routers so that each router authenticates the source of each routing update packet that it receives. This mechanism is accomplished by exchanging an authentication key (password) between the source and destination routers.

4.1 Configure MD5 authentication

Step 1. To enable router r1 configuration mode, issue the following command:

```
configure terminal
```

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

Figure 37. Enabling configuration mode on router r1.

Step 2. In order to configure BGP, type the following command:

```
router bgp 100
```

```
frr-pc# configure terminal
frr-pc(config)# router bgp 100
frr-pc(config-router)#
```

Figure 38. Configuring BGP on router r1.

Step 3. In router r1 terminal, type the following command to set a password (123) to the neighbor IP address 192.168.12.2.

```
neighbor 192.168.12.2 password 123
```

```
frr-pc# configure terminal  
frr-pc(config)# router bgp 100  
frr-pc(config-router)# neighbor 192.168.12.2 password 123  
frr-pc(config-router)#
```

Figure 39. Setting password for BGP peering with neighbor 192.168.12.2.

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

```
end
```

```
frr-pc# configure terminal  
frr-pc(config)# router bgp 100  
frr-pc(config-router)# neighbor 192.168.12.2 password 123  
frr-pc(config-router)# end  
frr-pc#
```

Figure 40. Exiting from configuration mode.

Step 5. Follow from step 1 to step 4 but with different metrics in order to configure BGP on router r2. Set password 123 for both the neighbors connected to router r2. All the steps are summarized in the following figure.

```
frr-pc# configure terminal  
frr-pc(config)# router bgp 200  
frr-pc(config-router)# neighbor 192.168.12.1 password 123  
frr-pc(config-router)# neighbor 192.168.23.2 password 123  
frr-pc(config-router)# end  
frr-pc#
```

Figure 41. Configuring BGP authentication on router r2.

Step 6. Follow from step 1 to step 4 but with different metrics in order to configure BGP on router r3. Set the password 345 for the neighbor with IP address 192.168.23.1 (router r2). The configured password on router r3 is different from the one configured on router r2 (123). All the steps are summarized in the following figure.

```
frr-pc# configure terminal  
frr-pc(config)# router bgp 300  
frr-pc(config-router)# neighbor 192.168.23.1 password 345  
frr-pc(config-router)# end  
frr-pc#
```

Figure 42. Configuring BGP authentication on router r3.

4.2 Verify MD5 authentication

Step 1. On host h3 terminal, perform a connectivity test by running the command shown below. To stop the test, press **Ctrl+c**. The results show that host h3 cannot reach host h1.

```
ping 192.168.1.10
```

```
root@frr-pc:~# ping 192.168.1.10
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.
From 192.168.3.1 icmp_seq=1 Destination Net Unreachable
From 192.168.3.1 icmp_seq=2 Destination Net Unreachable
From 192.168.3.1 icmp_seq=3 Destination Net Unreachable
^C
--- 192.168.1.10 ping statistics ---
3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 44ms

root@frr-pc:~#
```

Figure 43. Connectivity test using **ping** command.

Step 2. Type the following command to verify the routing table of router r3. You will notice that the routing table does not contain any route to the networks 192.168.1.0/24 and 192.168.2.0/24. The connection between routers r2 and r3 has dropped as the passwords exchanged between router r2 and router r3 are different.

```
show ip route
```

```
frr-pc# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       0 - 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>* 192.168.3.0/24 is directly connected, r3-eth0, 00:48:09
C>* 192.168.23.0/30 is directly connected, r3-eth1, 00:48:09
frr-pc#
```

Figure 44. Displaying the routing table of router r3.

Step 3. Type the following command to verify the routing table of router r2. The routing table does not have any route to the network 192.168.3.0/24. Router r2 has dropped the connection with router r3 as the password of router r2 did not match with the one configured on router r3.

```
show ip route
```

```
frrr-pc# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       0 - 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

B>* 192.168.1.0/24 [20/0] via 192.168.12.1, r2-eth1, 00:22:43
C>* 192.168.2.0/24 is directly connected, r2-eth0, 00:51:03
C>* 192.168.12.0/30 is directly connected, r2-eth1, 00:51:03
C>* 192.168.23.0/30 is directly connected, r2-eth2, 00:51:03
frrr-pc#
```

Figure 45. Displaying the routing table of router r2.

Step 4. In router r2 terminal, type the following command to verify BGP neighbors. Scroll down to verify the established connections. You will notice that two connections are established with router r2 and one connection is dropped due to BGP authentication.

```
show ip bgp neighbors
```

```
frrr-pc# show ip bgp neighbors
BGP neighbor is [192.168.12.1], remote AS 100, local AS 200, external link
  End-of-RIB send: IPv4 Unicast
  End-of-RIB received: IPv4 Unicast
  Message statistics:
    Inq depth is 0
    Outq depth is 0
          Sent      Rcvd
  Opens:          2          2
  Notifications: 0          2
  Updates:        9          9
  Keepalives:     49         49
  Route Refresh: 0          0
  Capability:    0          0
  Total:          60         62
  Minimum time between advertisement runs is 0 seconds

  For address family: IPv4 Unicast
    Update group 1, subgroup 1
    Packet Queue length 0
    Community attribute sent to this neighbor(all)
    1 accepted prefixes

[Connections established 2; dropped 1]
Last reset 00:28:03, No AFI/SAFI activated for peer
Local host: 192.168.12.2, Local port: 179
```

Figure 46. Verifying BGP neighbors of router r2.

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

References

1. A. Tanenbaum, D. Wetherall, "Computer networks", 5th Edition, Pearson, 2012.
2. J. Stewart III, BGP4 Inter-Domain Routing in the Internet, Addison-Wesley Longman Publishing Co., Inc., 1998.

3. Juniper networks, “BGP route authentication”, 2020. [Online]. Available: https://www.juniper.net/documentation/en_US/junos/topics/topic-map/bgp_security.html
4. Cisco, “Configuring authentication for BGP”, 2019. [Online]. Available: <https://community.cisco.com/t5/networking-documents/configuring-authentication-for-bgp/ta-p/3108287>
5. Cisco, “MD5 authentication between BGP peers configuration example”, 2010. [Online]. Available: <https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/112188-configure-md5-bgp-00.html#intro>



BORDER GATEWAY PROTOCOL

Lab 6: Configuring BGP with Default Routing

Document Version: **02-18-2020**



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	4
1.1 Multi-homed network.....	4
1.2 Default route	4
1.3 Floating static route	5
1.4 Access Control List.....	6
2 Lab topology.....	7
2.1 Lab settings.....	8
2.2 Open the topology and load the configuration	9
2.3 Load zebra daemon and verify configuration	12
3 Configure BGP on the routers.....	16
4 Configure route filters.....	21
4.1 Configure ACL.....	21
4.2 Verify configuration.....	22
5 Configure primary and backup routes using floating static routes	23
5.1 Configure default routes	23
5.2 Verify and test the default route	26
6 Propagate default route using BGP	28
References	30

Overview

This lab presents advanced Border Gateway Protocol (BGP) features. The goal of this lab is to create a multihomed network for Internet connectivity with fault tolerance. Thus, External BGP (EBGP) will be configured on all existing routers and route filters will be applied to permit or deny the route advertisement. Furthermore, floating static routes will be used to make one Internet Service Provider (ISP) as a primary provider and the other as a backup provider. Finally, a default route will be propagated from one router to another in different Autonomous Systems (ASes), where it will take effect on the delivered router without being configured manually.

Objectives

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

1. Explain the concept of default routing.
2. Configure and verify BGP between two ASes.
3. Apply route filters.
4. Set primary and backup routes using floating static routes.
5. Use BGP to propagate a default route.

Lab settings

The information in Table 1 provides the credentials to access Client1 machine.

Table 1. Credentials to access Client1 machine.

Device	Account	Password
Client1	admin	password

Lab roadmap

This lab is organized as follows:

1. Section 1: Introduction.
2. Section 2: Lab topology.
3. Section 3: Configure BGP on the routers.
4. Section 4: Configure route filters.
5. Section 5: Configure primary and backup routes using floating static routes.
6. Section 6: Propagate default route using BGP.

1 Introduction

1.1 Multi-homed network

BGP is an exterior gateway protocol designed to exchange routing and reachability information among ASes on the Internet. BGP is relevant to network administrators of large organizations which connect to one or more ISPs, as well as to ISPs who connect to other network providers. In terms of BGP, an AS is referred to as a routing domain, where all networked systems operate common routing protocols and are under the control of a single administration¹.

A single-homed network is one that is connected to the Internet through one ISP. In a single-homed network all the traffic that is destined to the Internet is sent to the ISP. If the link connecting the single-homed network with the ISP fails, all the traffic sent to the Internet will be affected. On the contrary, a multi-homed network is one that is connected to the Internet by two or more ISPs. Therefore, a multi-homed network provides additional bandwidth and redundancy⁷.

Consider Figure 1. The topology is multi-homed since the Customer is connected to two ISPs. The communication between the Customer and the ISPs is via EBGP since each network is in a different AS.

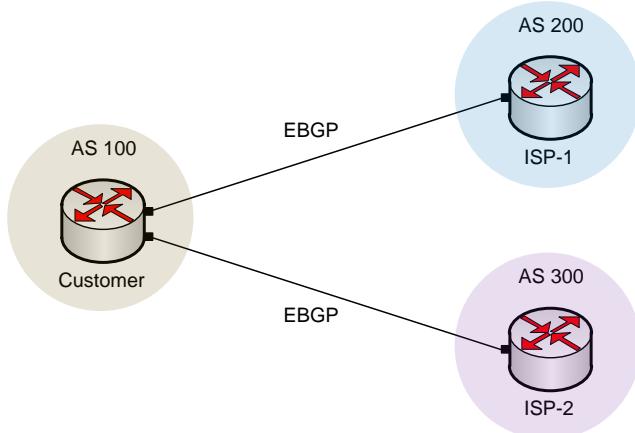


Figure 1. Multi-homed network.

1.2 Default route

Static routing is a nonadaptive routing algorithm, i.e., the routing decisions are taken regardless of any measurements or estimates of the current topology and traffic¹. Static routes are manually configured by the network administrator on a router's routing table.

A default route is a static route that takes effect when no other route is available for a destination IP address in the routing table¹.

A router receiving a packet inspects its routing table and checks if there is a match on the packet's destination IP address. If this is the case, the packet is forwarded according to the route information provided in the matched entry. Otherwise, the router forwards the packet to the default route if available².

Consider Figure 2. Router r2 is the border router that routes all the traffic transmitted to the Internet from Local Area Networks (LANs) 1 and 2. The Internet has an enormous number of network addresses that cannot fit into the routing table of router r2. Thus, a default route is configured so that the packets that arrive to router r2 from both LANs without a matched entry in its routing table are forwarded to the Internet.

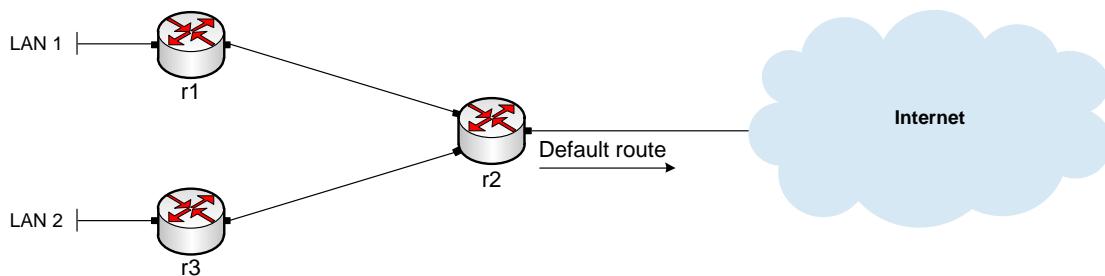


Figure 2. A default route is configured in router r2 to forward packets with unknown IP addresses to the Internet.

1.3 Floating static route

Administrative Distance is a feature used by routers in order to select the best path when there are two or more different routes to the same destination from different routing protocols. It defines how reliable or trustworthy the routing protocol is. The value of the administrative distance varies from 0 (most reliable) to 255 (least reliable/unknown)³. For example, a static route has a default administrative distance of 1.

Consider Figure 3. Router r1 has configured an administrative distance of 100 for path B (r1 to r2 to r3) and 110 for path A (r1 to r3). Thus, when router r1 wants to send packets to router r3, it will select path B since it has a lower administrative distance than path A.

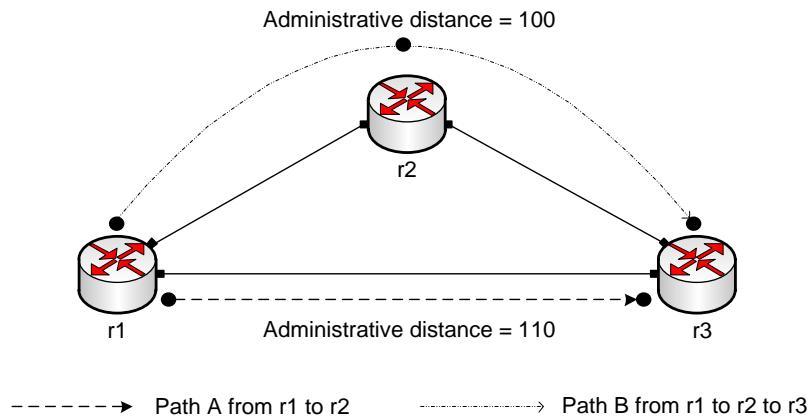


Figure 3. Path B is preferred when delivering packets from router r1 to router r3 since it has a lower administrative distance than path A.

Floating static route is used by routers to back up a dynamic route, i.e., the route that is learned from a dynamic routing protocol, such as BGP⁴. Floating static routes are configured to have a higher administrative distance than primary routes. Therefore, floating static routes are only used when primary routes are unavailable.

Consider Figure 4. There are two different paths from router r1 to router r2 (primary and floating static routes). The routing protocol configured in router r1 can deliver the packets destined to router r2 through the primary link. If this route fails (e.g., due to link failure), router r1 uses the floating static route as a backup.

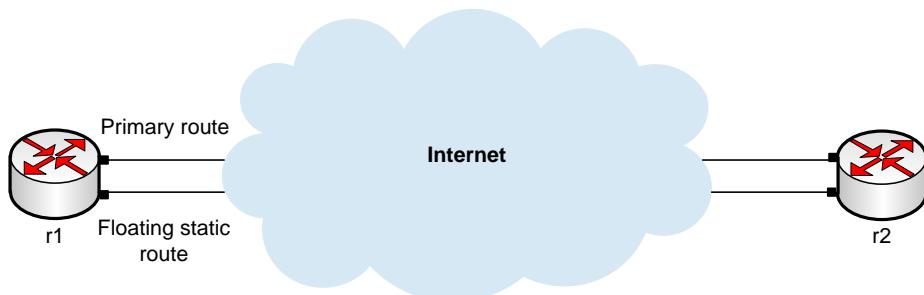


Figure 4. Floating static route is used when the primary route is unavailable.

1.4 Access Control List

An Access Control List (ACL) is a set of rules that perform packet filtering to control network traffic. They provide security measures by permitting or denying traffic, thus, restricting access to devices across a network⁵. ACLs are made up of one or more Access Control Entries (ACEs). An ACE is an entry in the ACL that specifies a permit or a deny rule. A permit rule allows authorized users to access specific resources, while a deny rule blocks any unwarranted attempts to reach these resources⁶.

Consider Figure 5. Router r1 is configured to allow host A to access the Internet by adding a permit ACE for this host. If host B tries to access the Internet, its IP address will not

match the ACE permit rule, hence, the packets won't be forwarded. More control over the network traffic can be achieved by configuring multiple ACEs.

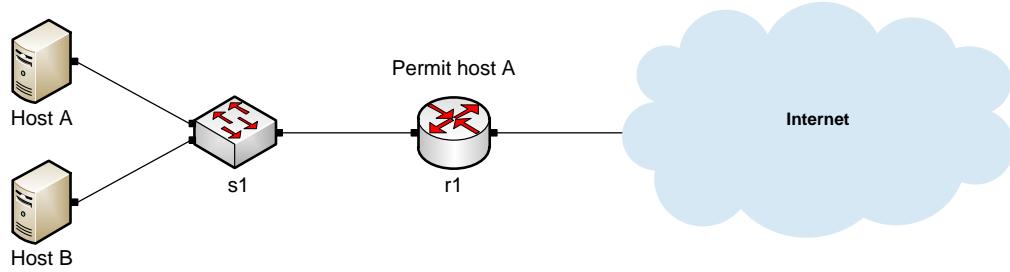


Figure 5. The ACL configured in router r1 allows only host A to access the Internet.

2 Lab topology

Consider Figure 6. The lab topology consists of three ASes, each identified by an Autonomous System Number (ASN). The ASNs assigned to ISP-1, the Campus network, and ISP-2 are 100, 200, and 300, respectively. ISP-1 and ISP-2 provide Internet connectivity to the Campus network. ISP-1 acts as the primary provider as it has a lower administrative distance than ISP-2. ISP-2 will act as a backup provider whenever the primary provider is not available. The communication between ASes is done via EBGP routing protocol.

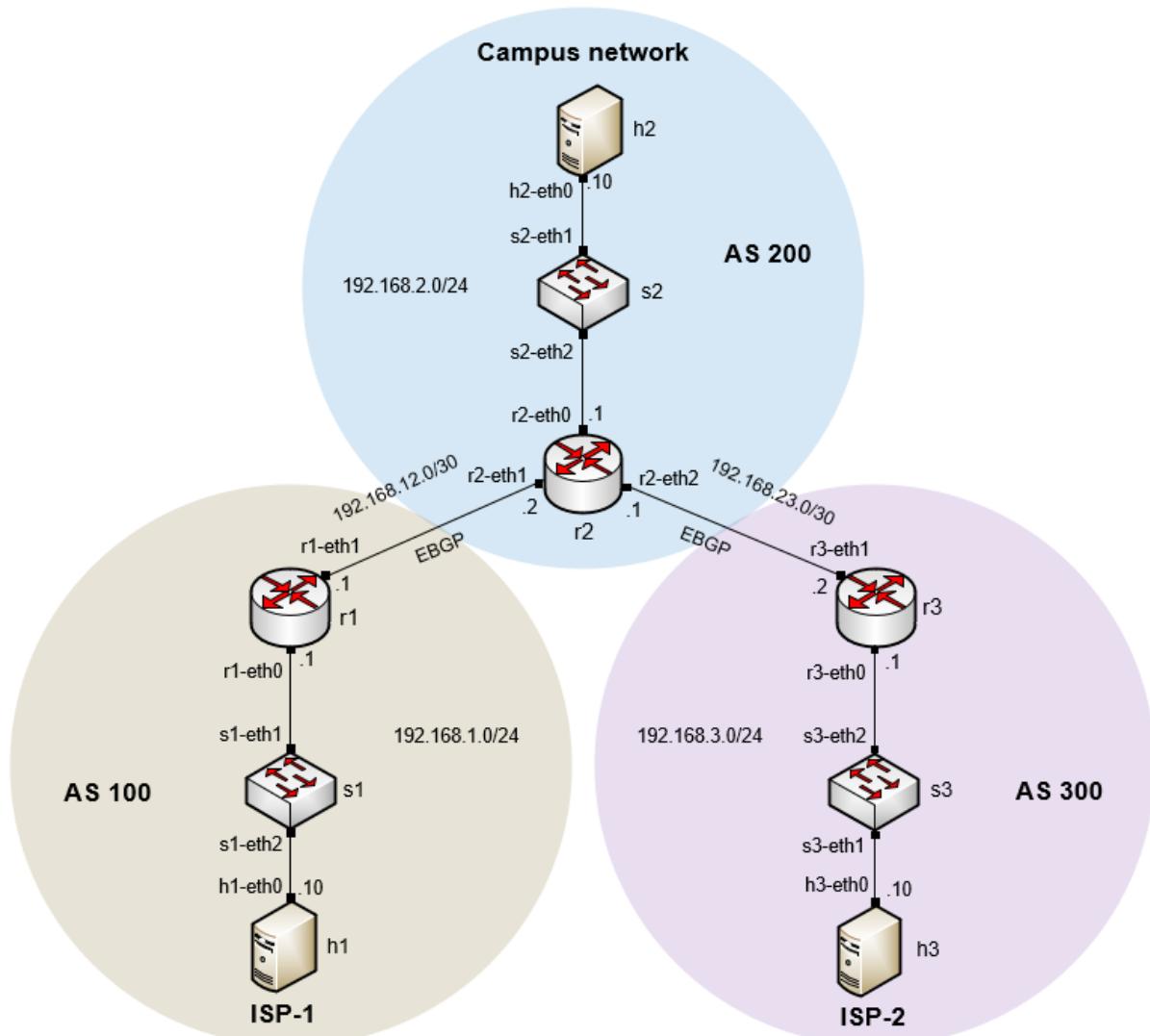


Figure 6. Lab topology.

2.1 Lab settings

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

Table 2. Topology information.

Device	Interface	IPv4 Address	Subnet	Default gateway
r1 (ISP-1)	r1-eth0	192.168.1.1	/24	N/A
	r1-eth1	192.168.12.1	/30	N/A
r2 (Campus network)	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

r3 (ISP-2)	r3-eth0	192.168.3.1	/24	N/A
	r3-eth1	192.168.23.2	/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
h3	h3-eth0	192.168.3.10	/24	192.168.3.1

2.2 Open the topology and load the configuration

Step 1. Start by launching Miniedit by clicking on Desktop's shortcut. When prompted for a password, type `password`.



Figure 7. MiniEdit shortcut.

Step 2. On Miniedit's menu bar, click on *File* then *open* to load the lab's topology. Locate the *Lab6.mn* topology file in the default directory, */home/frr/BGP_Labs/lab6* and click on *Open*.

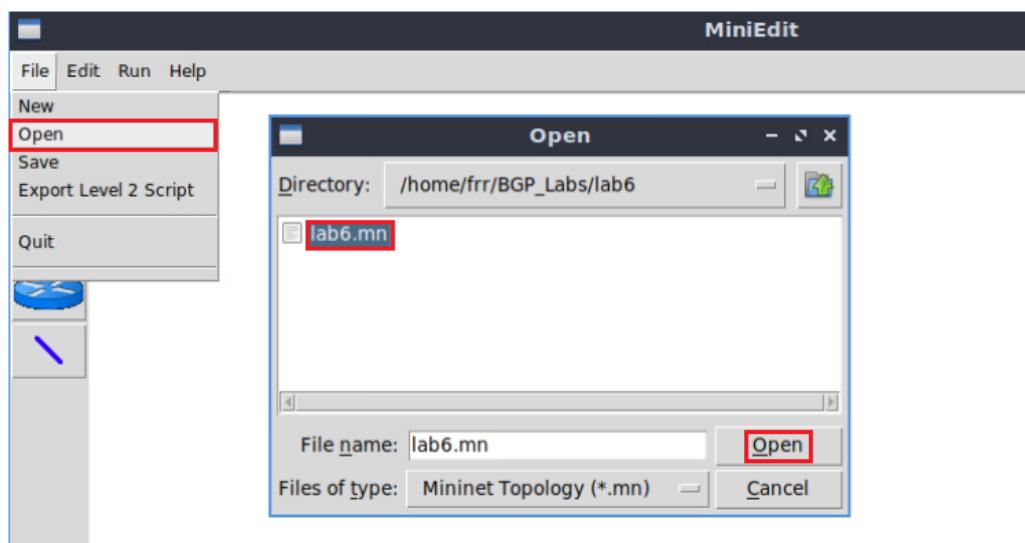


Figure 8. MiniEdit's Open dialog.

At this point the topology is loaded with all the required network components. You will execute a script that will load the configuration of the routers.

Step 3. Open the Linux terminal.

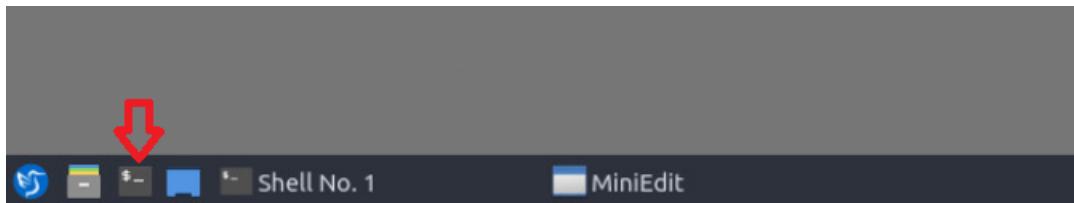


Figure 9. Opening Linux terminal.

Step 4. Click on the Linux's terminal and navigate into *BGP_Labs/lab6* 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 BGP_Labs/lab6
```

A screenshot of a terminal window titled "frr@frr-pc: ~/BGP_Labs/lab6". The command "cd BGP_Labs/lab6" is being typed into the terminal. The terminal window has a dark background with light-colored text.

Figure 10. Entering the BGP_Labs/lab6 directory.

Step 5. To execute the shell script, type the following command. The argument of the program corresponds to the configuration zip file that will be loaded in all the routers in the topology.

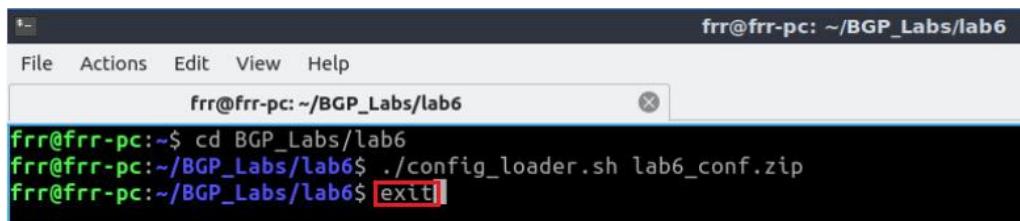
```
./config_loader.sh lab6_conf.zip
```

A screenshot of a terminal window titled "frr@frr-pc: ~/BGP_Labs/lab6". The command "./config_loader.sh lab6_conf.zip" is being typed into the terminal. The terminal window has a dark background with light-colored text.

Figure 11. Executing the shell script to load the configuration.

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

```
exit
```



```
frr@frr-pc: ~/BGP_Labs/lab6
File Actions Edit View Help
frr@frr-pc:~/BGP_Labs/lab6
frr@frr-pc:~/BGP_Labs/lab6$ ./config_loader.sh lab6_conf.zip
frr@frr-pc:~/BGP_Labs/lab6$ exit!
```

Figure 12. Exiting from the terminal.

Step 7. At this point hosts h1, h2 and h3 interfaces are configured. To proceed with the emulation, click on the *Run* button located in lower left-hand side.



Figure 13. Starting the emulation.

Step 8. Click on Mininet's terminal, i.e., the one launched when MiniEdit was started.

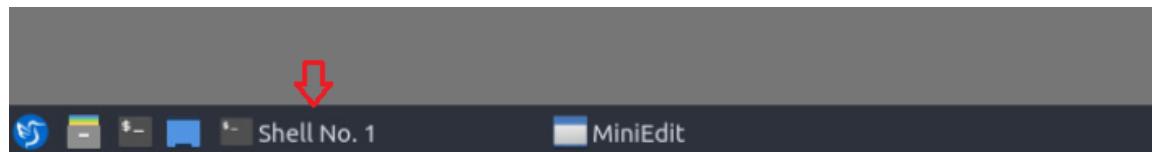
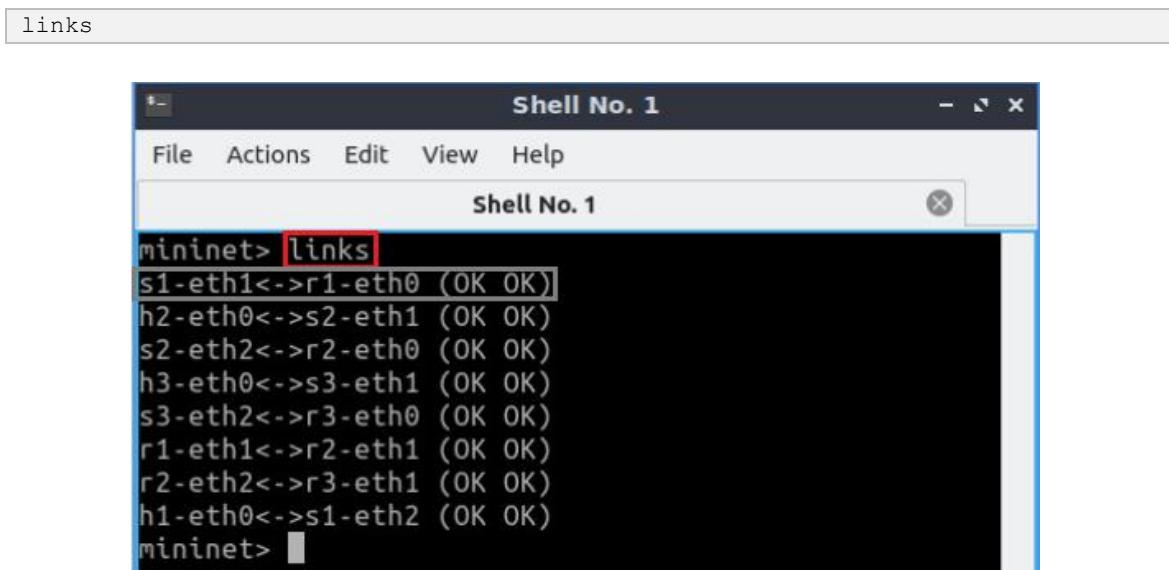


Figure 14. Opening Mininet's terminal.

Step 9. Issue the following command to display the interface names and connections.



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

Figure 15. Displaying network interfaces.

In Figure 15, the link displayed within the gray box indicates that interface eth1 of switch s1 connects to interface eth0 of router r1 (i.e., $s1\text{-}eth1 <-> r1\text{-}eth0$).

2.3 Load zebra daemon and verify configuration

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

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

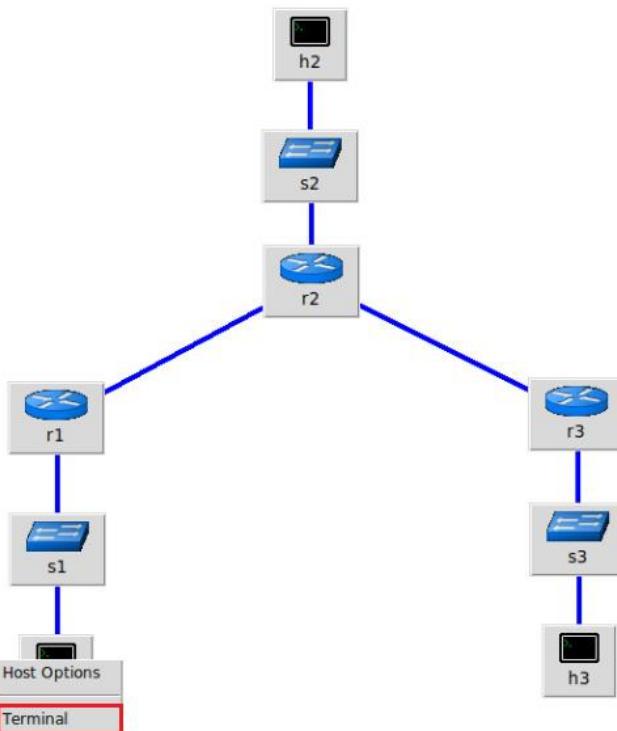
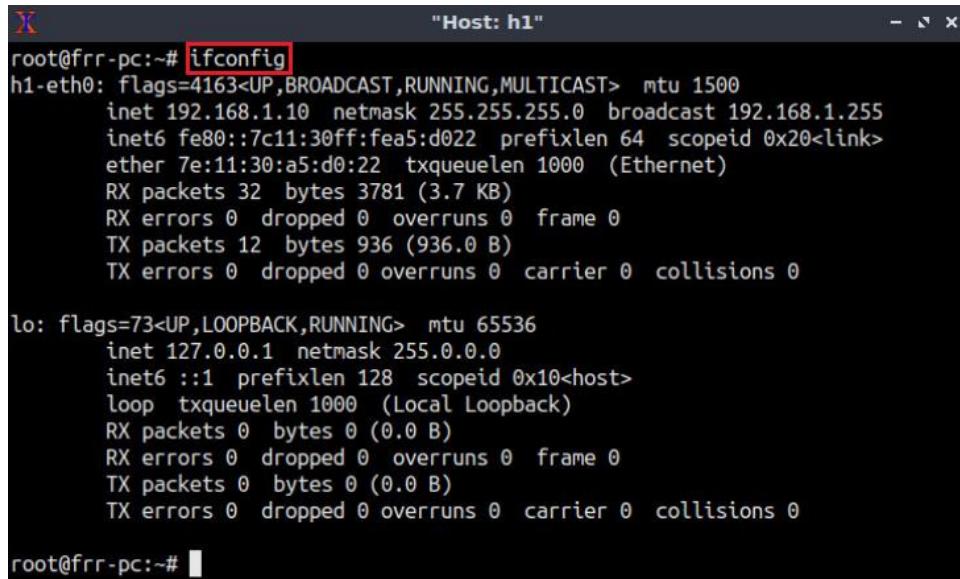


Figure 16. Opening a terminal on host h1.

Step 2. On 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.

```
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

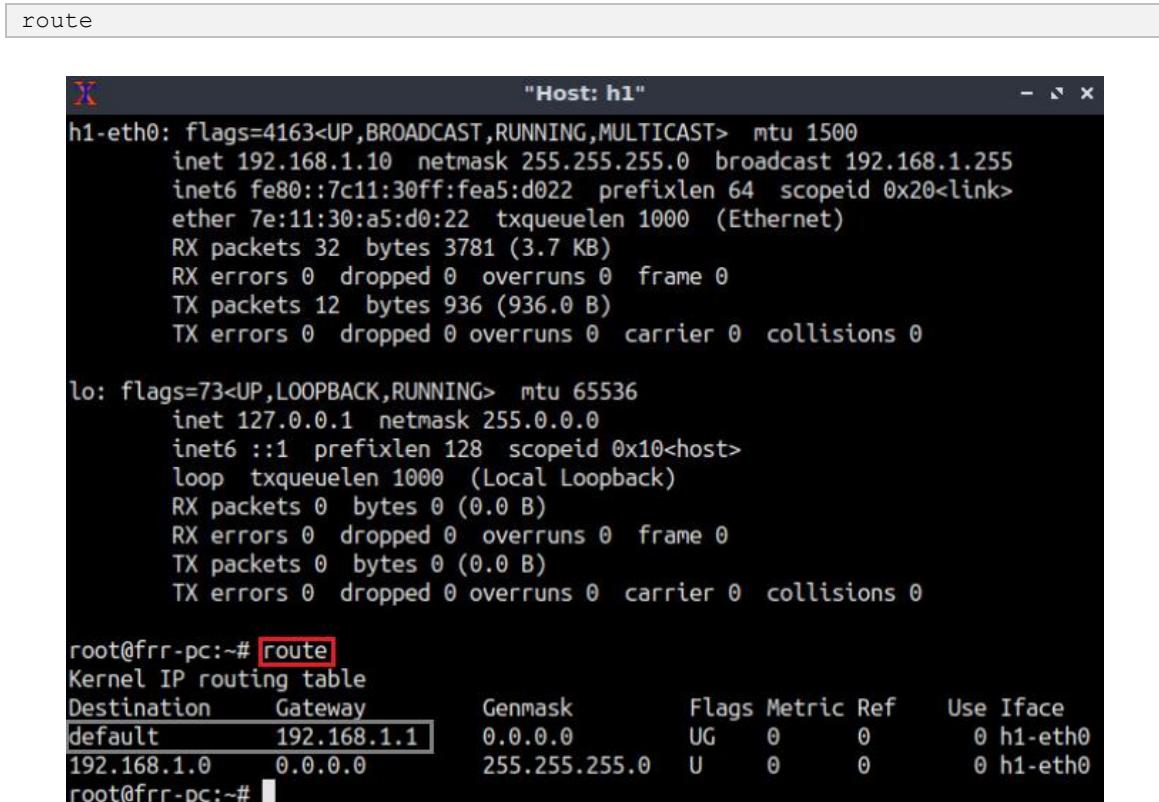
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
                inet6 ::1 prefixlen 128 scopeid 0x10<host>
                    loop txqueuelen 1000 (Local Loopback)
                        RX packets 0 bytes 0 (0.0 B)
                        RX errors 0 dropped 0 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 17. Output of `ifconfig` command.

Step 3. On 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:~# 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 18. Output of `route` command.

Step 4. In order to verify hosts h2 and h3, proceed similarly by repeating from step 1 to step 3 on hosts h2 and h3 terminals. Similar results should be observed.

Step 5. You will validate that the router interfaces are configured correctly according to Table 2. In order to verify router r1, hold right-click on router r1 and select Terminal.

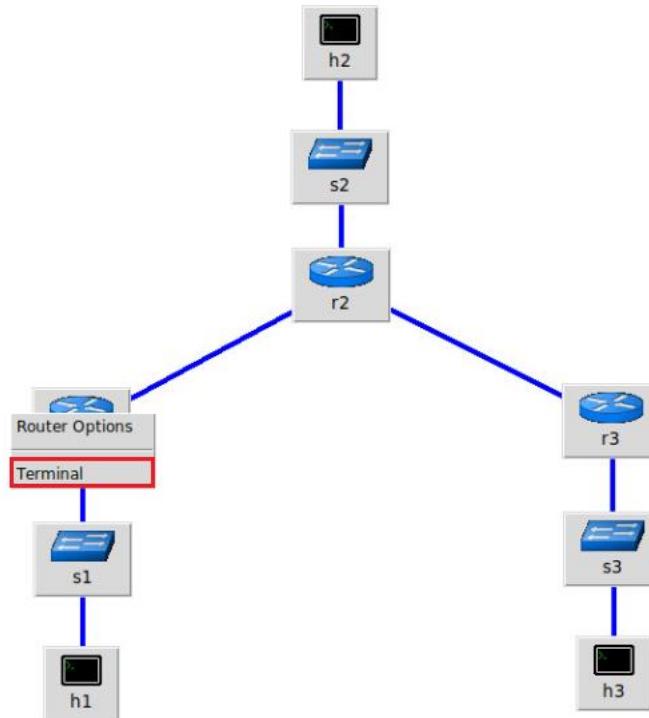


Figure 19. Opening a terminal on router r1.

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

```
zebra
```

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

Figure 20. Starting zebra daemon.

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

```
vtysh
```

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

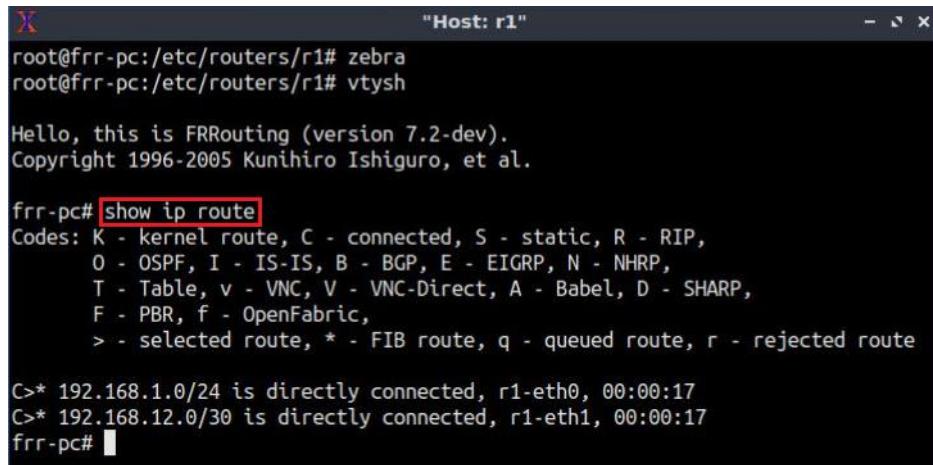
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc#
```

Figure 21. Starting vtysh 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 attached to routers r2 (192.168.2.0/24) and router r3 (192.168.3.0/24) 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

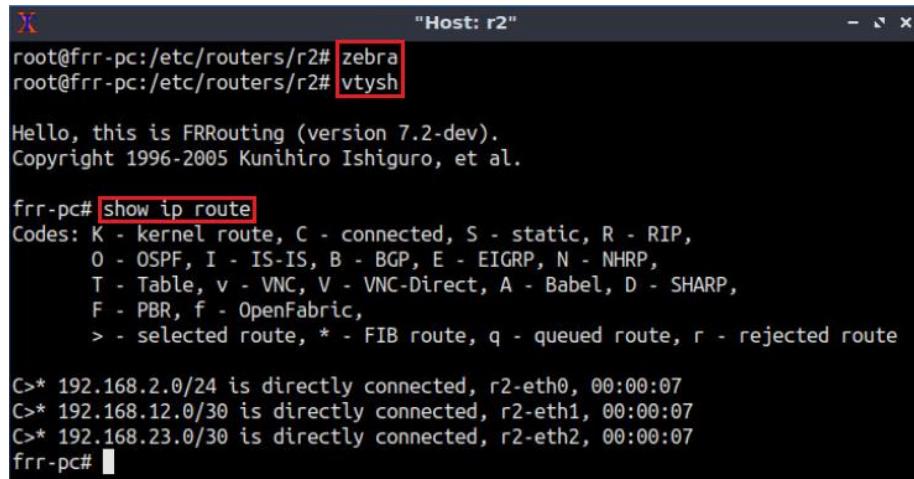
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

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, 00:00:17
C>* 192.168.12.0/30 is directly connected, r1-eth1, 00:00:17
frr-pc#
```

Figure 22. Displaying routing table of router r1.

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 networks directly connected networks of router r2.



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

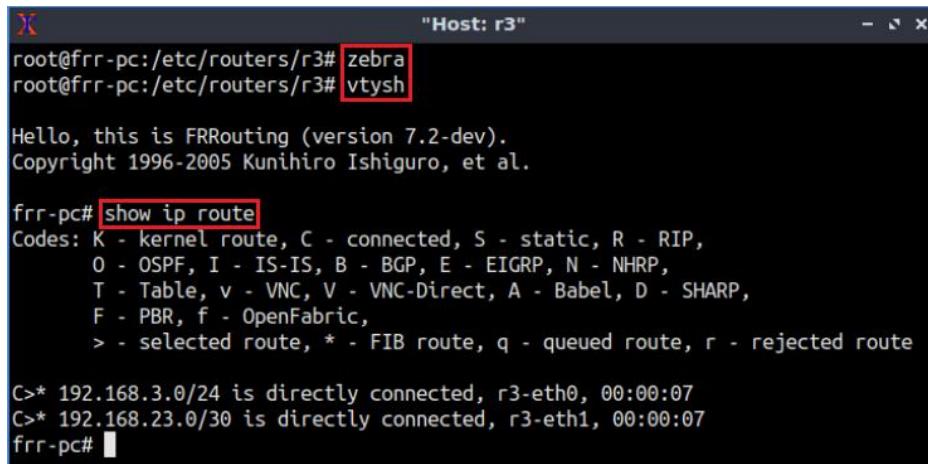
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

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, 00:00:07
C>* 192.168.12.0/30 is directly connected, r2-eth1, 00:00:07
C>* 192.168.23.0/30 is directly connected, r2-eth2, 00:00:07
frr-pc#
```

Figure 23. Displaying routing table of router r2.

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



```
"Host: r3"
root@frr-pc:/etc/routers/r3# zebra
root@frr-pc:/etc/routers/r3# 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>* 192.168.3.0/24 is directly connected, r3-eth0, 00:00:07
C>* 192.168.23.0/30 is directly connected, r3-eth1, 00:00:07
frr-pc#
```

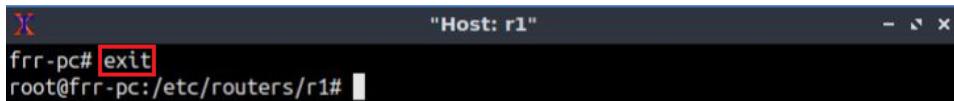
Figure 24. Displaying routing table of router r3.

3 Configure BGP on the routers

In this section, you will configure EBGP in the routers that are hosted in different ASes. You will assign BGP neighbors to allow the routers to exchange BGP routes. Furthermore, routers r1, r2, and r3 will advertise their LANs via BGP so that the LANs are learned by peer routers.

Step 1. To configure BGP routing protocol, you need to enable the BGP daemon first. In router r1, type the following command to exit the vtysh session:

```
exit
```

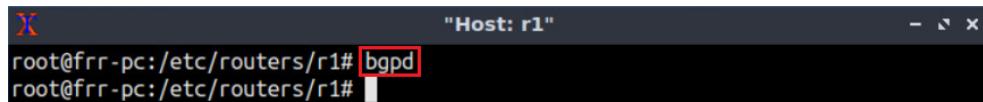


```
"Host: r1"
frr-pc# exit
root@frr-pc:/etc/routers/r1#
```

Figure 25. Exiting the vtysh session.

Step 2. Type the following command on router r1 terminal to enable and to start BGP routing protocol.

```
bgpd
```



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

Figure 26. Starting BGP daemon.

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

```
vtysh
```



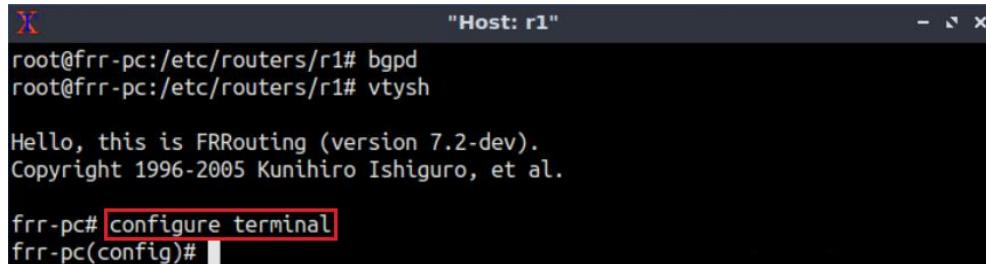
```
"Host: r1"
root@frr-pc:/etc/routers/r1# bgpd
root@frr-pc:/etc/routers/r1# vtysh
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc#
```

Figure 27. Starting vtysh on router r1.

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

```
configure terminal
```



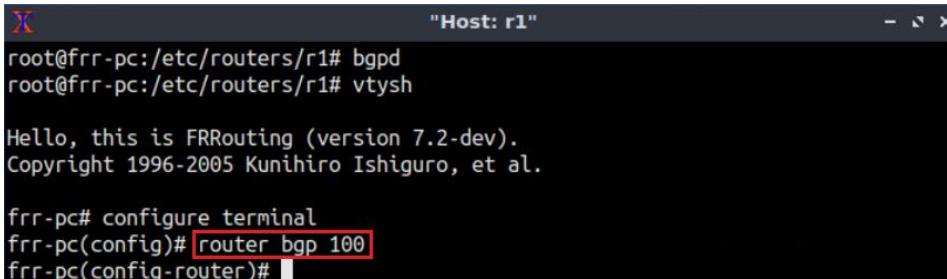
```
"Host: r1"
root@frr-pc:/etc/routers/r1# bgpd
root@frr-pc:/etc/routers/r1# vtysh
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

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

Figure 28. Enabling configuration mode on router r1.

Step 5. The ASN assigned for router r1 is 100. In order to configure BGP, type the following command:

```
router bgp 100
```



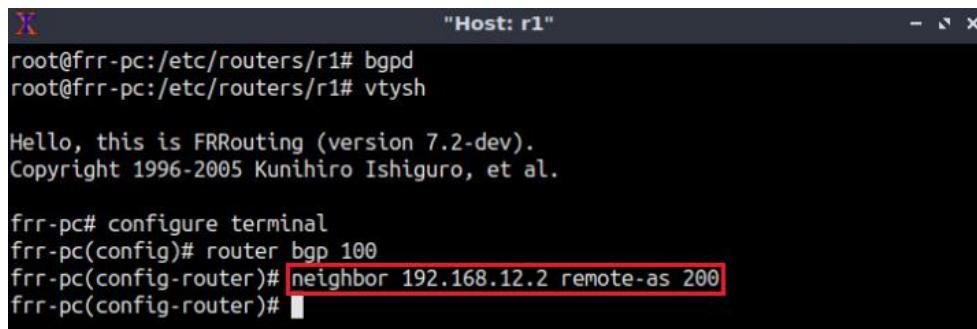
```
"Host: r1"
root@frr-pc:/etc/routers/r1# bgpd
root@frr-pc:/etc/routers/r1# vtysh
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# router bgp 100
frr-pc(config-router)#
```

Figure 29. Configuring BGP on router r1.

Step 6. To configure a BGP neighbor to router r1 (AS 100), type the command shown below. This command specifies the neighbor IP address (192.168.12.2) and the ASN of the remote BGP peer (AS 200).

```
neighbor 192.168.12.2 remote-as 200
```



```
"Host: r1"
root@frrr-pc:/etc/routers/r1# bgpd
root@frrr-pc:/etc/routers/r1# vtysh

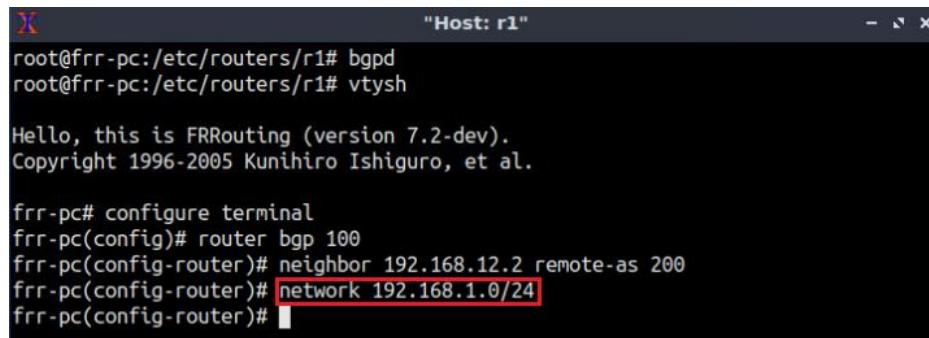
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frrr-pc# configure terminal
frrr-pc(config)# router bgp 100
frrr-pc(config-router)# neighbor 192.168.12.2 remote-as 200
frrr-pc(config-router)#[ ]
```

Figure 30. Assigning BGP neighbor to router r1.

Step 7. In this step, router r1 will advertise LAN 192.168.1.0/24 to its BGP peers. To do so, issue the following command:

```
network 192.168.1.0/24
```



```
"Host: r1"
root@frrr-pc:/etc/routers/r1# bgpd
root@frrr-pc:/etc/routers/r1# vtysh

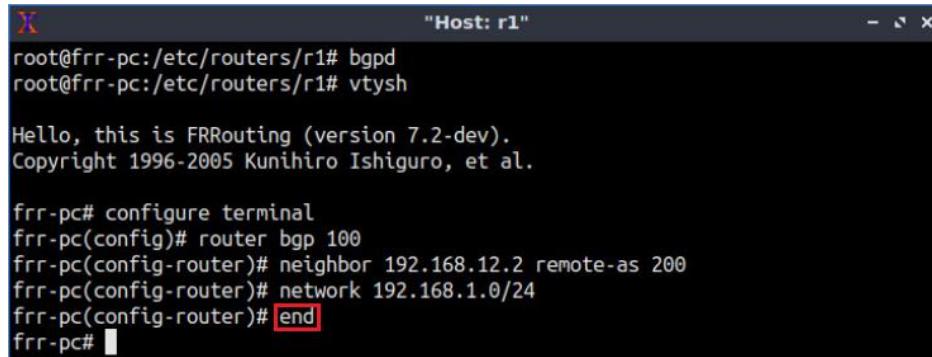
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frrr-pc# configure terminal
frrr-pc(config)# router bgp 100
frrr-pc(config-router)# neighbor 192.168.12.2 remote-as 200
frrr-pc(config-router)# network 192.168.1.0/24
frrr-pc(config-router)#[ ]
```

Figure 31. Advertising local network on router r1.

Step 8. Type the following command to exit from the configuration mode.

```
end
```



```
"Host: r1"
root@frrr-pc:/etc/routers/r1# bgpd
root@frrr-pc:/etc/routers/r1# vtysh

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

frrr-pc# configure terminal
frrr-pc(config)# router bgp 100
frrr-pc(config-router)# neighbor 192.168.12.2 remote-as 200
frrr-pc(config-router)# network 192.168.1.0/24
frrr-pc(config-router)# end
frrr-pc#[ ]
```

Figure 32. Exiting from configuration mode.

Step 9. Type the following command to verify BGP networks. You will observe the LAN network of router r1.

```
show ip bgp
```

```
frr-pc# show ip bgp
BGP table version is 1, local router ID is 192.168.12.1, vrf id 0
Default local pref 100, local AS 100
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*-> 192.168.1.0/24    0.0.0.0                  0        32768 i

Displayed 1 routes and 1 total paths
frr-pc#
```

Figure 33. Verifying BGP networks on router r1.

Step 10. Type the following command to verify BGP neighbors. You will verify that the neighbor IP address is 192.168.12.2. The corresponding ASN is 200.

```
show ip bgp neighbors
```

```
frr-pc# show ip bgp neighbors
BGP neighbor is 192.168.12.2, remote AS 200, local AS 100, external link
  BGP version 4, remote router ID 0.0.0.0, local router ID 192.168.12.1
  BGP state = Active
  Last read 00:04:15, Last write never
  Hold time is 180, keepalive interval is 60 seconds
  Message statistics:
    Inq depth is 0
    Outq depth is 0
              Sent      Rcvd
  Opens:          0          0
  Notifications: 0          0
  Updates:       0          0
  Keepalives:    0          0
  Route Refresh: 0          0
  Capability:   0          0
  Total:         0          0
  Minimum time between advertisement runs is 0 seconds

  For address family: IPv4 Unicast
    Not part of any update group
    Community attribute sent to this neighbor(all)
    0 accepted prefixes
```

Figure 34. Verifying BGP neighbors on router r1.

Step 11. Follow from step 1 to step 8 but with different metrics in order to configure BGP on router r2. All these steps are summarized in the following figure.

```
frr-pc# exit
root@frr-pc:/etc/routers/r2# bgpd
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 bgp 200
frr-pc(config-router)# neighbor 192.168.12.1 remote-as 100
frr-pc(config-router)# neighbor 192.168.23.2 remote-as 300
frr-pc(config-router)# network 192.168.2.0/24
frr-pc(config-router)# end
frr-pc#
```

Figure 35. Configuring BGP on router r2.

Step 12. Follow from step 1 to step 8 in order to configure BGP on router r3. All these steps are summarized in the following figure.

```
frr-pc# exit
root@frr-pc:/etc/routers/r3# bgpd
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 bgp 300
frr-pc(config-router)# neighbor 192.168.23.1 remote-as 200
frr-pc(config-router)# network 192.168.3.0/24
frr-pc(config-router)# end
frr-pc#
```

Figure 36. Configure BGP on router r3.

Step 13. In router r2 terminal, type the following command to verify the routing table of router r2. The LANs of router r1 (192.168.1.0/24) and router r3 (192.168.3.0/24) are advertised to router r2 through EBGP.

```
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

B>* [192.168.1.0/24] [20/0] via 192.168.12.1, r2-eth1, 00:13:28
C>* 192.168.2.0/24 is directly connected, r2-eth0, 00:16:04
B>* [192.168.3.0/24] [20/0] via 192.168.23.2, r2-eth2, 00:02:39
C>* 192.168.12.0/30 is directly connected, r2-eth1, 00:16:04
C>* 192.168.23.0/30 is directly connected, r2-eth2, 00:16:04
frr-pc#
```

Figure 37. Verifying the routing table of router r2.

4 Configure route filters

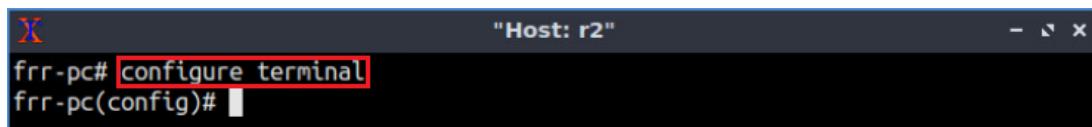
To ensure that the network operates efficiently, routing updates must be controlled and tuned. Information about networks must be sent where it is needed and filtered from where it is not needed.

In this lab, there is no need to connect the two ISPs. To do so, you will apply route filters by creating an ACL that controls the routing updates that are advertised by router r2. Thus, the two ISPs do not receive route information about each other's LANs.

4.1 Configure ACL

Step 1. In router r2 terminal, type the following command to enable the configuration mode:

```
configure terminal
```

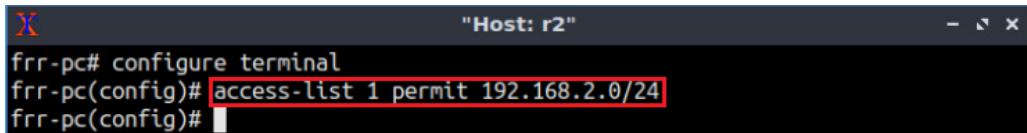


```
frr-pc# configure terminal
frr-pc(config)#
```

Figure 38. Enabling configuration mode on router r2.

Step 2. In this step, you will create an ACL that permits the network 192.168.2.0/24. An ACL must have a number within the range 1-99. In this lab, you will use 1 as the ACL number. Type the following command to configure the ACL in router r2.

```
access-list 1 permit 192.168.2.0/24
```

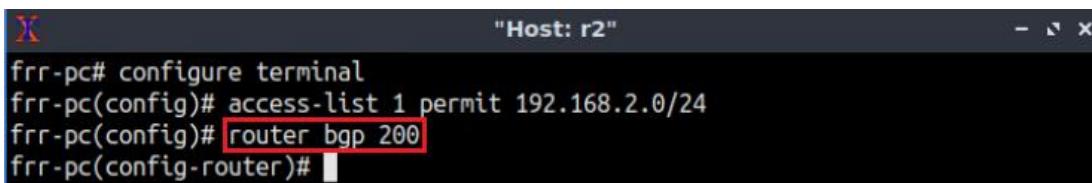


```
frr-pc# configure terminal
frr-pc(config)# access-list 1 permit 192.168.2.0/24
frr-pc(config)#
```

Figure 39. Creating an ACL that permits the LAN of router r2.

Step 3. You will filter the route updates that are advertised by router r2 to its neighbors using the configured ACL. Type the following command to enter BGP configuration mode:

```
router bgp 200
```

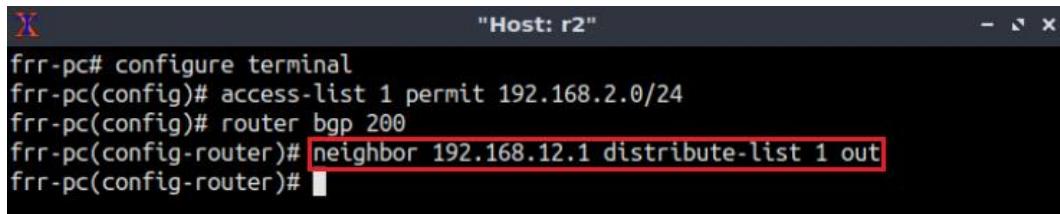


```
frr-pc# configure terminal
frr-pc(config)# access-list 1 permit 192.168.2.0/24
frr-pc(config)# router bgp 200
frr-pc(config-router)#
```

Figure 40. Configuring BGP on router r2.

Step 4. Router r2 must advertise only its LAN (192.168.2.0/24) to the neighbor with IP address 192.168.12.1 (router r1). To do so, type the below command. The `distribute-list 1 out` option will apply the ACL numbered 1 to filter all outgoing route updates.

```
neighbor 192.168.12.1 distribute-list 1 out
```

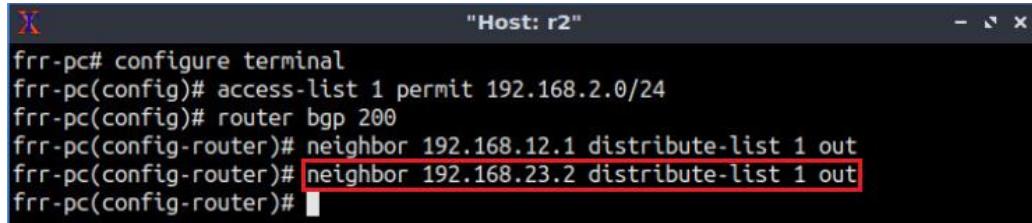


```
frr-pc# configure terminal
frr-pc(config)# access-list 1 permit 192.168.2.0/24
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.12.1 distribute-list 1 out
frr-pc(config-router)#
```

Figure 41. Applying the ACL to router r2 neighbor.

Step 5. Similarly, type the below command so that router r2 only advertises its LAN (192.168.2.0/24) to the neighbor with IP address 192.168.23.2 (router r3).

```
neighbor 192.168.23.2 distribute-list 1 out
```



```
frr-pc# configure terminal
frr-pc(config)# access-list 1 permit 192.168.2.0/24
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.12.1 distribute-list 1 out
frr-pc(config-router)# neighbor 192.168.23.2 distribute-list 1 out
frr-pc(config-router)#
```

Figure 42. Applying the ACL to router r2 neighbor.

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

```
end
```



```
frr-pc# configure terminal
frr-pc(config)# access-list 1 permit 192.168.2.0/24
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.12.1 distribute-list 1 out
frr-pc(config-router)# neighbor 192.168.23.2 distribute-list 1 out
frr-pc(config-router)# end
frr-pc#
```

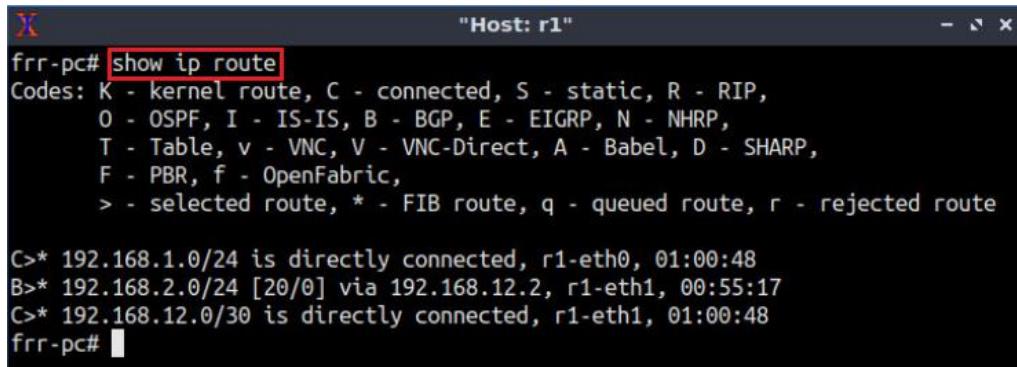
Figure 43. Exiting from configuration mode.

4.2 Verify configuration

In this section, you will verify the configured route filters by validating that routers r1 and r3 are not receiving any route information about each other's LANs.

Step 1. Type the following command to verify the routing table of router r1. The route to LAN 192.168.3.0/24 should not be in the table.

```
show ip route
```

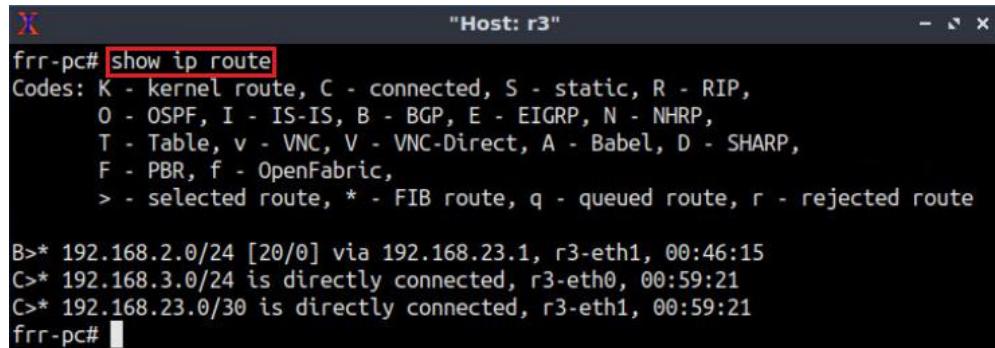


```
"Host: r1"
frr-pc# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       0 - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued route, r - rejected route
C>* 192.168.1.0/24 is directly connected, r1-eth0, 01:00:48
B>* 192.168.2.0/24 [20/0] via 192.168.12.2, r1-eth1, 00:55:17
C>* 192.168.12.0/30 is directly connected, r1-eth1, 01:00:48
frr-pc#
```

Figure 44. Displaying routing table of router r1.

Step 2. Type the following command to verify the routing table of router r3. The route to LAN 192.168.1.0/24 should not be in the table.

```
show ip route
```



```
"Host: r3"
frr-pc# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       0 - 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
B>* 192.168.2.0/24 [20/0] via 192.168.23.1, r3-eth1, 00:46:15
C>* 192.168.3.0/24 is directly connected, r3-eth0, 00:59:21
C>* 192.168.23.0/30 is directly connected, r3-eth1, 00:59:21
frr-pc#
```

Figure 45. Displaying routing table of router r3.

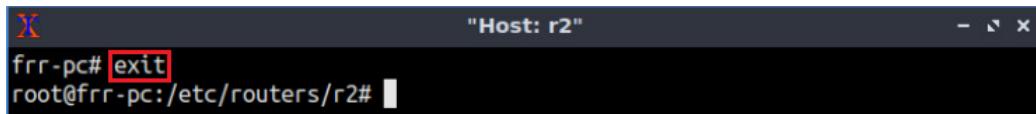
5 Configure primary and backup routes using floating static routes

In this section, you will configure floating static routes to reflect the policy that ISP-1 is the primary provider and ISP-2 is the backup provider. You will assign an administrative distance of 210 and 220 to the routes from the Customer to ISP-1 and from the Customer to ISP-2, respectively. The backup provider will be active whenever the link to the primary provider is unavailable. You will verify and test the floating static routes by creating an unadvertised loopback in router r1.

5.1 Configure default routes

Step 1. To configure static routing, you need to enable the static routing daemon first. In router r2, type the following command to exit the vtysh session:

```
exit
```

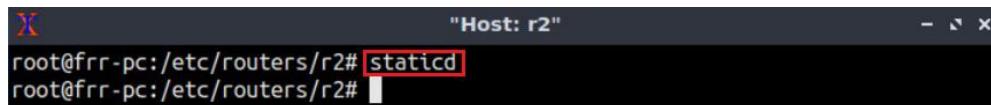


```
frr-pc# exit  
root@frr-pc:/etc/routers/r2#
```

Figure 46. Exiting from terminal on router r2.

Step 2. Type the following command on router r2 terminal to enable and start static routing protocol.

```
staticd
```

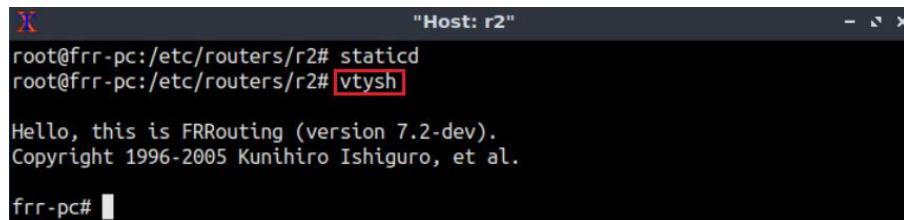


```
root@frr-pc:/etc/routers/r2# staticd  
root@frr-pc:/etc/routers/r2#
```

Figure 47. Starting static routing daemon.

Step 3. In order to enter to router r2 terminal, type the following command:

```
vtysh
```

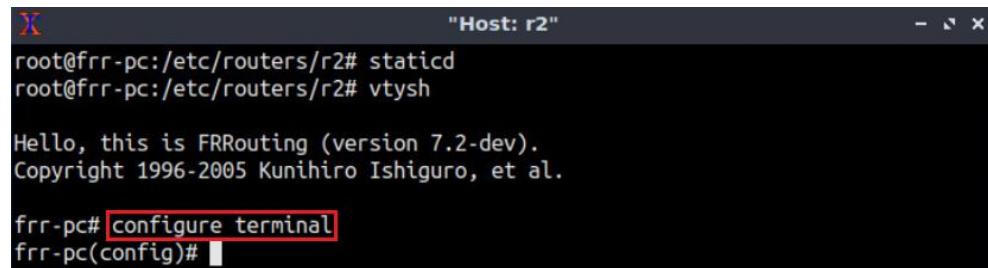


```
root@frr-pc:/etc/routers/r2# staticd  
root@frr-pc:/etc/routers/r2# vtysh  
  
Hello, this is FRRouting (version 7.2-dev).  
Copyright 1996-2005 Kunihiro Ishiguro, et al.  
  
frr-pc#
```

Figure 48. Starting vtysh on router r2.

Step 4. To enable router r2 configuration mode, issue the following command:

```
configure terminal
```



```
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 49. Enabling configuration mode on router r2.

Step 5. Type the following command to configure a floating static route for ISP-1 with administrative distance 210. In this step, the first `0.0.0.0` defines any network, and the `0.0.0.0` defines any subnet. All the traffic will be forwarded to router r1 through its IP address 192.168.12.1.

```
ip route 0.0.0.0 0.0.0.0 192.168.12.1 210
```

```
"Host: r2"
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 0.0.0.0 192.168.12.1 210
frr-pc(config)#

```

Figure 50. Configuring floating static route with administrative distance 210.

Step 6. Type the following command to configure a floating static route for ISP-2 with administrative distance 220. The traffic will always use the route with the lowest administrative distance to reach to a destination. ISP-2 will be acting as a backup since its route has a higher administrative distance than that of ISP-1.

```
ip route 0.0.0.0 0.0.0.0 192.168.23.2 220
```

```
"Host: r2"
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 0.0.0.0 192.168.12.1 210
frr-pc(config)# ip route 0.0.0.0 0.0.0.0 192.168.23.2 220
frr-pc(config)#

```

Figure 51. Configuring floating static route with distance administrative distance 220.

Step 7. Type the following command to exit from configuration mode.

```
exit
```

```
"Host: r2"
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.

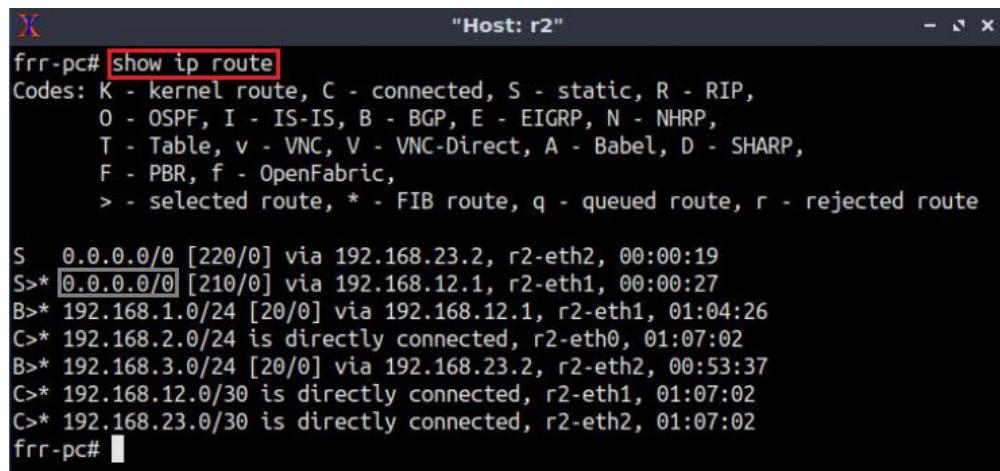
frr-pc# configure terminal
frr-pc(config)# ip route 0.0.0.0 0.0.0.0 192.168.12.1 210
frr-pc(config)# ip route 0.0.0.0 0.0.0.0 192.168.23.2 220
frr-pc(config)# exit
frr-pc#

```

Figure 52. Exiting from configuration mode.

Step 8. Type the following command to verify the routing table of router r2.

```
show ip route
```



```
"Host: r2"
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 [220/0] via 192.168.23.2, r2-eth2, 00:00:19
S>* 0.0.0.0/0 [210/0] via 192.168.12.1, r2-eth1, 00:00:27
B>* 192.168.1.0/24 [20/0] via 192.168.12.1, r2-eth1, 01:04:26
C>* 192.168.2.0/24 is directly connected, r2-eth0, 01:07:02
B>* 192.168.3.0/24 [20/0] via 192.168.23.2, r2-eth2, 00:53:37
C>* 192.168.12.0/30 is directly connected, r2-eth1, 01:07:02
C>* 192.168.23.0/30 is directly connected, r2-eth2, 01:07:02
frr-pc#"
```

Figure 53. Displaying routing table of router r2.

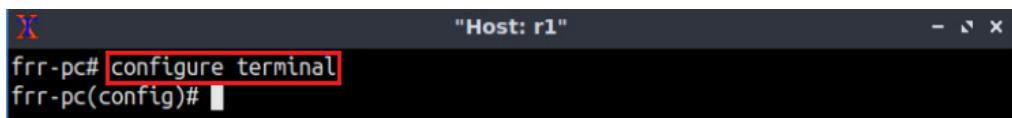
Consider Figure 53. The default routes should appear in the routing table (primary route has been shown within the gray box). The default route with the lower administrative distance will be used since it contains the characters `>*`. The default route with the higher administrative distance will be working only when the primary route is unavailable.

5.2 Verify and test the default route

In this section, you will verify if the configured floating static routes are working properly. Create a loopback address in router r1 and verify if router r2 can communicate with the loopback address. You will be using the loopback address because it has not been advertised to router r2. Router r2 will use the default route to reach the unadvertised network.

Step 1. Type the following command to get into the configuration mode on router r1.

```
configure terminal
```

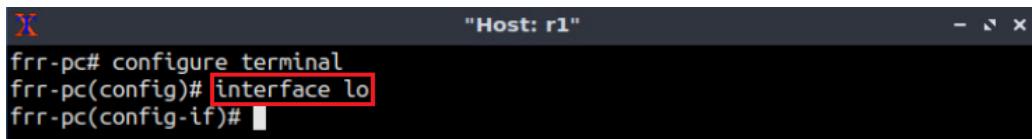


```
"Host: r1"
frr-pc# configure terminal
frr-pc(config)#"
```

Figure 54. Enabling configuration mode on router r1.

Step 2. In order to configure a loopback interface, type the following command:

```
interface lo
```



```
"Host: r1"
frr-pc# configure terminal
frr-pc(config)# interface lo
frr-pc(config-if)#"
```

Figure 55. Configuring loopback interface in router r1.

Step 3. To assign an IP address to `lo`, type the following command:

```
ip address 192.168.100.1/24
```

```
frr-pc# configure terminal
frr-pc(config)# interface lo
frr-pc(config-if)# ip address 192.168.100.1/24
frr-pc(config-if)#
```

Figure 56. Assigning an IP address for the loopback interface.

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

```
exit
```

```
frr-pc# configure terminal
frr-pc(config)# interface lo
frr-pc(config-if)# ip address 192.168.100.1/24
frr-pc(config-if)# end
frr-pc#
```

Figure 57. Exiting from configuration mode.

Step 5. In router r2 terminal, type the following command to verify the routing table of router r2. The IP address of router r1's loopback interface should not exist in the routing table.

```
show ip route
```

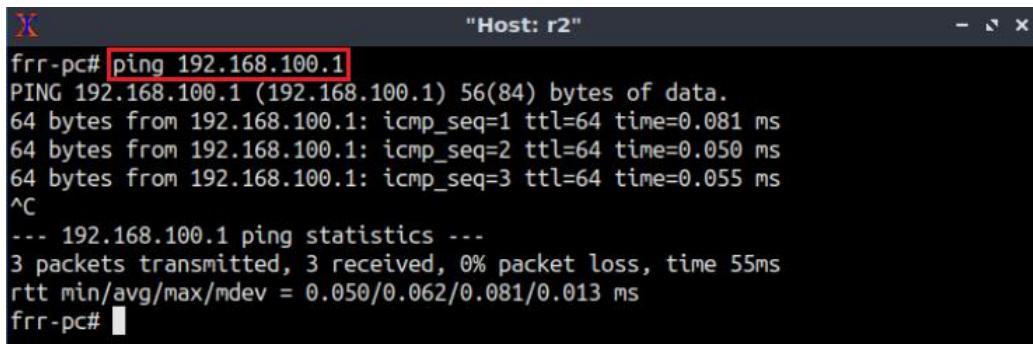
```
frr-pc# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       0 - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued route,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

S  0.0.0.0/0 [220/0] via 192.168.23.2, r2-eth2, 00:07:13
S>* 0.0.0.0/0 [210/0] via 192.168.12.1, r2-eth1, 00:07:21
B>* 192.168.1.0/24 [20/0] via 192.168.12.1, r2-eth1, 01:11:20
C>* 192.168.2.0/24 is directly connected, r2-eth0, 01:13:56
B>* 192.168.3.0/24 [20/0] via 192.168.23.2, r2-eth2, 01:00:31
C>* 192.168.12.0/30 is directly connected, r2-eth1, 01:13:56
C>* 192.168.23.0/30 is directly connected, r2-eth2, 01:13:56
frr-pc#
```

Figure 58. Displaying the routing table of router r2.

Step 6. Verify the default route using the `ping` command and check if router r2 can reach the IP address 192.168.100.1 of router r1's loopback interface. Type the following command to test the connectivity. To stop the test, press `Ctrl+d`. The figure below shows a successful connectivity test.

```
ping 192.168.100.1
```



```
frr-pc# ping 192.168.100.1
PING 192.168.100.1 (192.168.100.1) 56(84) bytes of data.
64 bytes from 192.168.100.1: icmp_seq=1 ttl=64 time=0.081 ms
64 bytes from 192.168.100.1: icmp_seq=2 ttl=64 time=0.050 ms
64 bytes from 192.168.100.1: icmp_seq=3 ttl=64 time=0.055 ms
^C
--- 192.168.100.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 55ms
rtt min/avg/max/mdev = 0.050/0.062/0.081/0.013 ms
frr-pc#
```

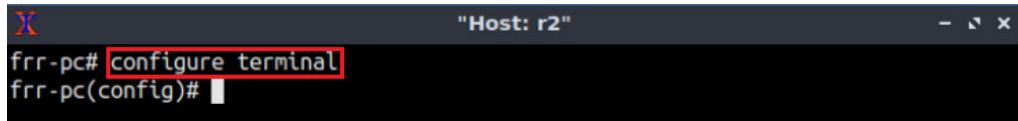
Figure 59. Connectivity test using `ping` command.

6 Propagate default route using BGP

In this section, you will configure BGP to advertise a default Route to BGP neighbors. Router r1 will generate and advertise a default route to the BGP peer router r2.

Step 1. In router r2 terminal, type the following command to enable the configuration mode.

```
configure terminal
```

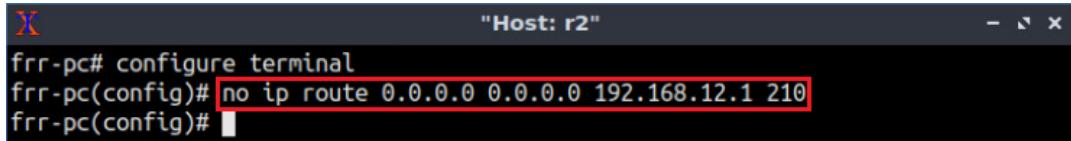


```
frr-pc# configure terminal
frr-pc(config)#
```

Figure 60. Enabling configuration mode on router r2.

Step 2. To remove the first configured default route, type the following command shown below.

```
no ip route 0.0.0.0 0.0.0.0 192.168.12.1 210
```

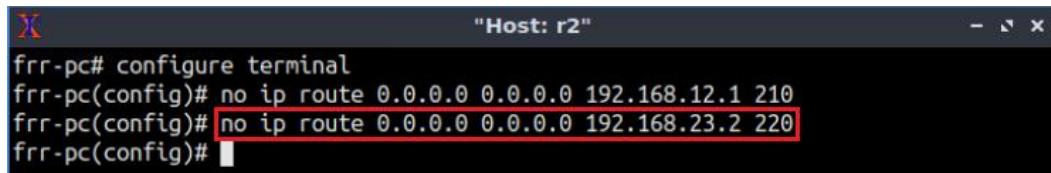


```
frr-pc# configure terminal
frr-pc(config)# no ip route 0.0.0.0 0.0.0.0 192.168.12.1 210
frr-pc(config)#
```

Figure 61. Removing a default route from router r2.

Step 3. To remove the second configured default route, type the following command shown below.

```
no ip route 0.0.0.0 0.0.0.0 192.168.23.2 220
```

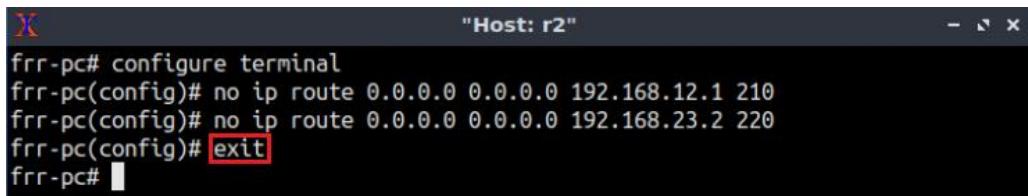


```
frr-pc# configure terminal
frr-pc(config)# no ip route 0.0.0.0 0.0.0.0 192.168.12.1 210
frr-pc(config)# no ip route 0.0.0.0 0.0.0.0 192.168.23.2 220
frr-pc(config)#
```

Figure 62. Removing a default route from router r2.

Step 4. Type the following command to exit from configuration mode. This directive will apply the configuration.

```
exit
```

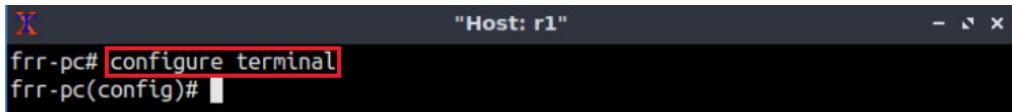


```
frr-pc# configure terminal
frr-pc(config)# no ip route 0.0.0.0 0.0.0.0 192.168.12.1 210
frr-pc(config)# no ip route 0.0.0.0 0.0.0.0 192.168.23.2 220
frr-pc(config)# exit
frr-pc#
```

Figure 63. Exiting from configuration mode.

Step 5. In router r1 terminal, type the following command to enable the configuration mode.

```
configure terminal
```

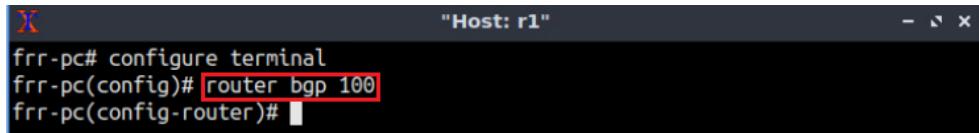


```
frr-pc# configure terminal
frr-pc(config)#
```

Figure 64. Enabling configuration mode on router r1.

Step 6. Type the following command to enter BGP configuration mode.

```
router bgp 100
```



```
frr-pc# configure terminal
frr-pc(config)# router bgp 100
frr-pc(config-router)#
```

Figure 65. Configuring BGP on router r1.

Step 7. Type the below command to generate and advertise a default route. The following command enables router r1 to advertise a default route to the neighbor IP address 192.168.12.2 (router r2).

```
neighbor 192.168.12.2 default-originate
```



```
frr-pc# configure terminal
frr-pc(config)# router bgp 100
frr-pc(config-router)# neighbor 192.168.12.2 default-originate
frr-pc(config-router)#
```

Figure 66. Creating default-route via BGP on router r1.

Step 8. Type the following command to exit from the configuration mode.

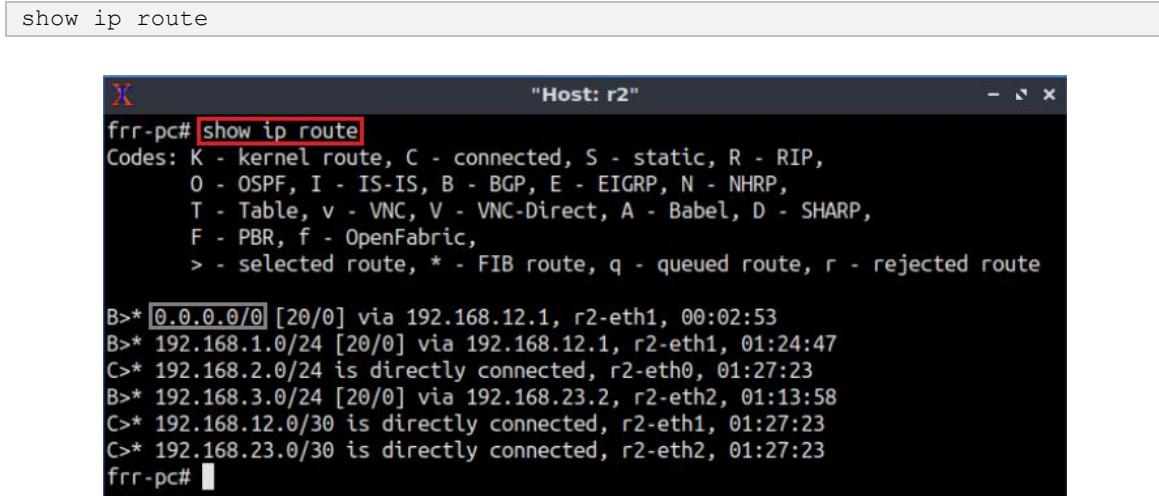
```
end
```



```
frr-pc# configure terminal
frr-pc(config)# router bgp 100
frr-pc(config-router)# neighbor 192.168.12.2 default-originate
frr-pc(config-router)# end
frr-pc#
```

Figure 67. Exiting from configuration mode.

Step 9. Type the following command to verify the routing table of router r2. The default route appears in the routing table (showed within the gray box).



```
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

B>* [0.0.0.0/0] [20/0] via 192.168.12.1, r2-eth1, 00:02:53
B>* 192.168.1.0/24 [20/0] via 192.168.12.1, r2-eth1, 01:24:47
C>* 192.168.2.0/24 is directly connected, r2-eth0, 01:27:23
B>* 192.168.3.0/24 [20/0] via 192.168.23.2, r2-eth2, 01:13:58
C>* 192.168.12.0/30 is directly connected, r2-eth1, 01:27:23
C>* 192.168.23.0/30 is directly connected, r2-eth2, 01:27:23
frr-pc#
```

Figure 68. Displaying routing table of router r2.

This concludes Lab 6. Stop the emulation and then exit out of MiniEdit.

References

1. A. Tanenbaum, D. Wetherall, “Computer networks”, 5th Edition, Pearson, 2012.
2. Cisco networking academy, “Cisco networking academy's introduction to routing dynamically”, 2014. [Online]. Available: <http://www.ciscopress.com/articles/article.asp?p=2180210&seqNum=7>
3. Cisco, “What is administrative distance”, 2013, [Online] Available: <https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/15986-admin-distance.html>
4. Cisco, “Cisco nexus 7000 series NX-OS System management configuration guide, release 5.x”, 2011. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/switches/datacenter/sw/5_x/nx-os/unicast/configuration/guide/l3_cli_nxos.pdf
5. Cisco, “Security configuration guide: access control lists, cisco IOS XERelease 3S”, 2015 [Online] Available: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_data_acl/configuration/xe-3s/sec-data-acl-xe-3s-book.pdf

6. Cisco, "ACL Rule", 2015. [Online]. Available:
https://www.cisco.com/c/dam/assets/sol/sb/WAP571_Emulators/WAP571_Emulator_v1-0-0-14/help/en/index.html#
7. A. Freedman, "Automated control of outbound transit links in a multi-homed BGP routing environment." U.S. Patent Application 09/887,655, 2002. [Online]. Available:
<https://patentimages.storage.googleapis.com/71/f1/f1/e337f09a79c859/US2002199016A1.pdf>



BORDER GATEWAY PROTOCOL

Lab 7: Using AS_PATH BGP attribute

Document Version: **01-23-2020**



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 Public and private ASN	3
1.2 AS_PATH attribute.....	4
1.3 Removing private ASN in BGP	5
1.4 Route filtering using AS_PATH attribute	5
2 Lab topology.....	6
2.1 Lab settings.....	7
2.2 Open topology and load the configuration.....	7
2.3 Load zebra daemon and Verify the Connectivity.....	10
3 Configure BGP on all routers	14
4 Remove the private ASN	19
5 Use the AS_PATH attribute to filter routes	21
5.1 Configure AS_PATH ACL	22
5.2 Verify Configuration	23
References	26

Overview

This lab discusses public and private Autonomous System Numbers (ASNs) that are assigned to Autonomous Systems (ASes) in Border Gateway Protocol (BGP). Additionally, the lab introduces BGP AS_PATH attribute and explains how to implement a policy that restricts network traffic using this attribute. In this lab, the terms BGP and External BGP (EBGP) will be used interchangeably since they will only be running between ASes.

Objectives

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

1. Explain the concept of public and private ASN.
2. Configure and verify BGP between two ASes.
3. Remove private ASNs from AS_PATH attribute.
4. Use the AS_PATH attribute to filter BGP routes.

Lab settings

The information in Table 1 provides the credentials to access Client1 machine.

Table 1. Credentials to access Client1 machine.

Device	Account	Password
Client1	admin	password

Lab roadmap

This lab is organized as follows:

1. Section 1: Introduction.
2. Section 2: Lab topology.
3. Section 3: Configure BGP on all routers.
4. Section 4: Remove the private ASN.
5. Section 5: Use the AS_PATH attribute to filter routes.

1 **Introduction**

1.1 **Public and private ASN**

The Internet consists of many independent administrative domains, referred to as ASes. ASes are operated by different organizations. BGP, also known as the interdomain routing protocol, is used to exchange routing information between ASes. In BGP, the path to a destination is described as a sequence of ASes that must be traversed to reach the destination¹.

Each AS is identified by an ASN that is either public or private. A public ASN is globally unique and can be advertised across the Internet; however, a private ASN is not globally unique and should not be advertised to external networks. Private ASNs range from 64512 to 65534, and from 4,200,000,000 to 4,294,967,294. All other ASNs are public and available for use on the Internet except for few reserved numbers².

A public ASN is required only when an AS is originating routes that are visible on the Internet. However, a private ASN should be used when an AS is only exchanging routes via BGP with a single Internet Service Provider (ISP)³.

Consider Figure 1. The AS of the customer is assigned a private ASN (64512) since the customer is connected to one ISP via BGP. The ISP has a public ASN (100) since it originates the routes that are visible on the Internet.

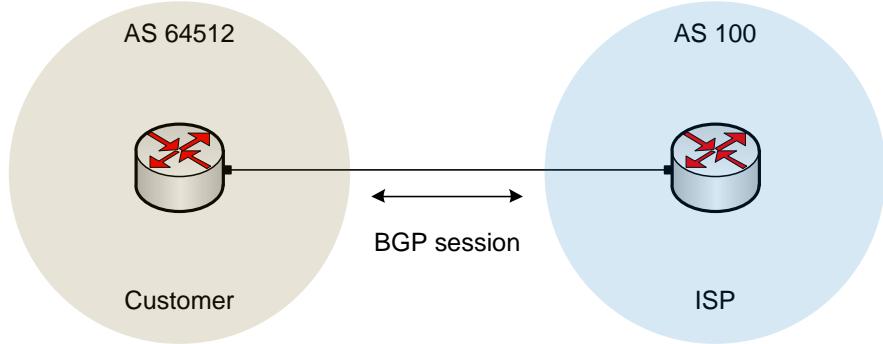


Figure 1. Customer Network has a private ASN and it exchanges BGP routes with the ISP that has a public ASN.

1.2 AS_PATH attribute

In BGP, when a router advertises a network across a BGP session, i.e., between two routers running BGP, it includes a number of BGP attributes⁴. These attributes help BGP select the best path when there are multiple paths to the same destination⁵.

The AS_PATH attribute is a list of all ASes that a specific route passes through to reach a specified network. When a router is advertising a BGP route, the AS_PATH attribute is first created empty. Each time the route is advertised from one AS to another, the AS_PATH attribute is modified to prepend the ASN of the router that advertised the route⁴.

Consider Figure 2. Every router prepends its own ASN to the AS_PATH attribute before it advertises the route to another AS. Eventually, router r4 receives the route advertisement with the AS_PATH attribute in the form of {300,200,100}.

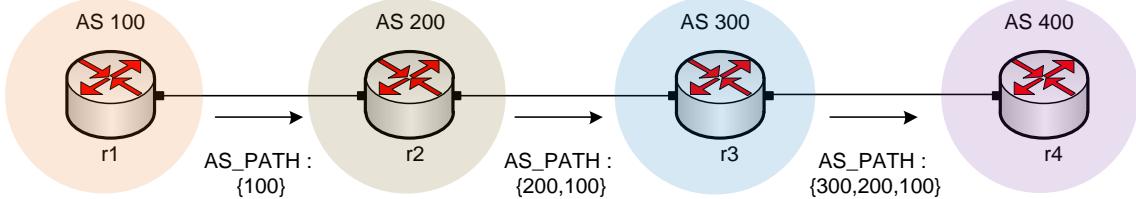


Figure 2. AS_PATH attribute prepending during route advertisement from one AS to another.

Routers use the AS_PATH attribute to detect and prevent loops. For example, a router drops any route in which its own ASN is part of the AS_PATH attribute⁴.

1.3 Removing private ASN in BGP

Private ASNs are not globally unique, hence, they cannot be leaked to the Internet. To achieve this goal, routers must strip the private ASNs from the AS_PATH attribute list before the routes are advertised to the Internet⁶.

Consider Figure 3. ISP-1 strips the private ASN 64512 from the AS_PATH attribute of all route advertisements originated by the customer. Thus, ISP-2 receives the route advertisement with the AS_PATH attribute containing only the ASN of ISP-1 (100).

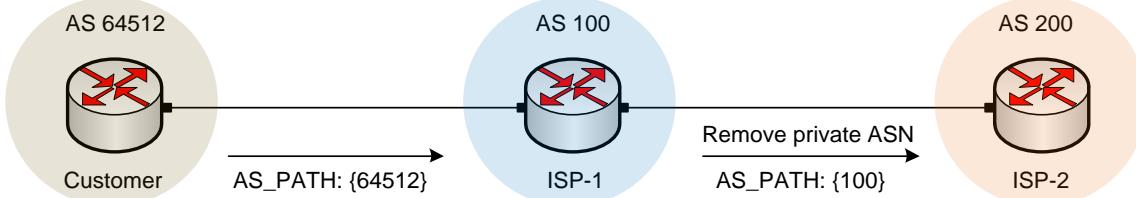


Figure 3. ISP-1 removes the private ASN of the customer from AS_PATH attribute before advertising the route to ISP-2.

1.4 Route filtering using AS_PATH attribute

An Access Control List (ACL) is a set of rules that perform packet filtering to control network traffic⁷. Routers can create ACLs to filter incoming or outgoing routes based on their AS_PATH attributes (AS_PATH ACL). Several scenarios may require filtering and selection of routing information based on the content of the AS_PATH attribute carried with each BGP route⁸.

For example, an AS can only allow local route advertisements, i.e., the routes that originate from the AS itself, by permitting those with the empty AS_PATH attribute only.

Consider Figure 4. Router r3 is configured with an ACL that only permits the routes originating from AS 200 to be advertised to AS 400. Thus, any route that does not have the ASN 200 at the end of its AS_PATH attribute will not be advertised to AS 400.

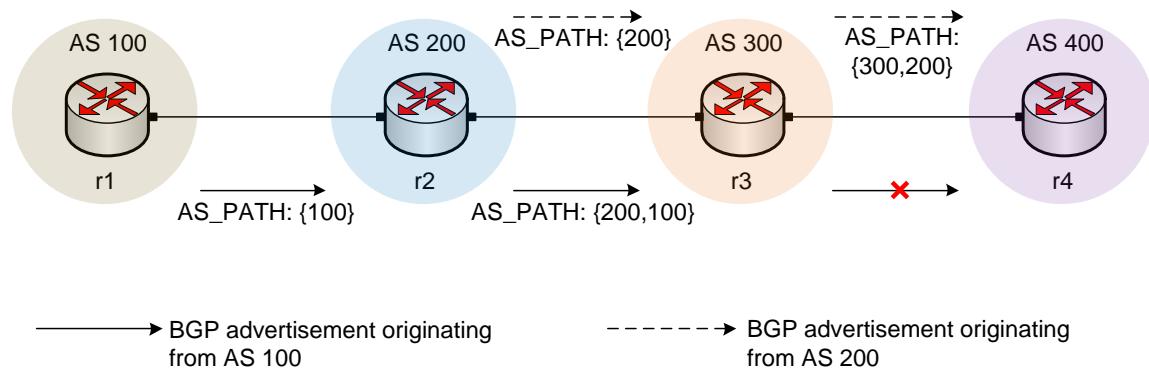


Figure 4. Router r3 is configured with an ACL that prevents route advertisements not sourced from AS 200.

2 Lab topology

Consider Figure 5. The lab topology consists of three ASes, each identified by an ASN that is either public or private. The ASNs assigned to the Campus network, the ISP, and the Customer are 100, 200, and 65000, respectively. The ISP must remove the private ASN of the Customer before it advertises it to the Campus network. Furthermore, the ISP will create an ACL so that the Customer does not receive route information from the Campus network. The ISP communicates with the Customer and the Campus network via EBGP routing protocol.

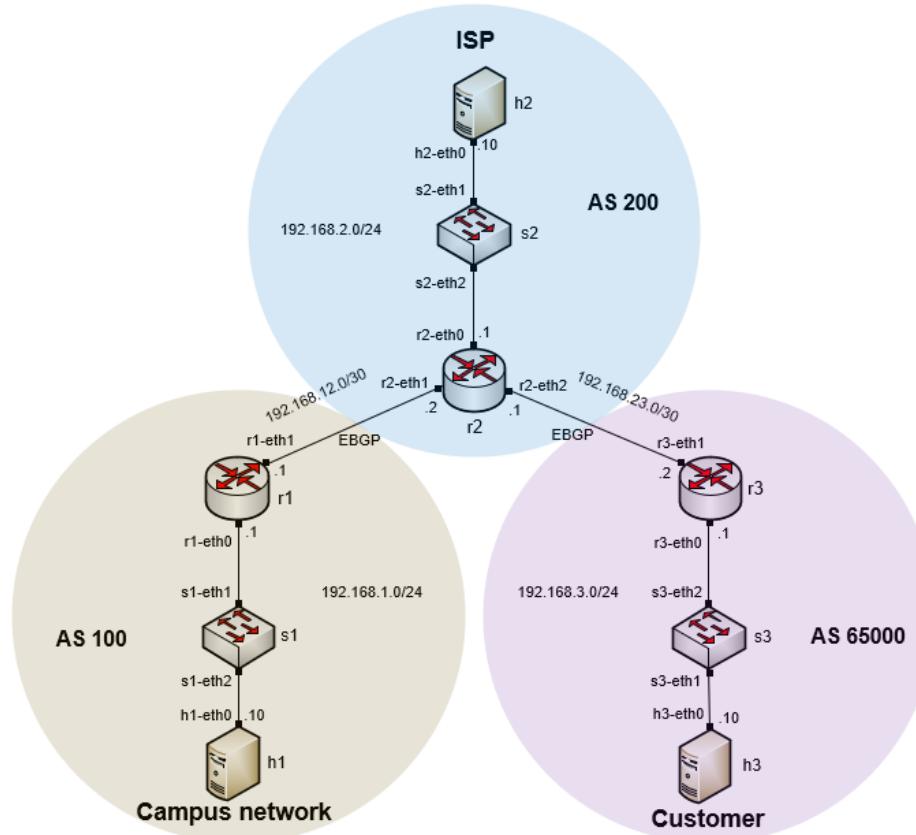


Figure 5. Lab topology.

2.1 Lab settings

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

Table 2. Topology information.

Device	Interface	IPV4 Address	Subnet	Default gateway
r1 (Campus network)	r1-eth0	192.168.1.1	/24	N/A
	r1-eth1	192.168.12.1	/30	N/A
r2 (ISP)	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
r3 (Customer)	r3-eth0	192.168.3.1	/24	N/A
	r3-eth1	192.168.23.2	/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
h3	h3-eth0	192.168.3.10	/24	192.168.3.1

2.2 Open topology and load the configuration

Step 1. Start by launching Minedit by clicking on Desktop's shortcut. When prompted for a password, type `password`.



Figure 6. MiniEdit shortcut.

Step 2. On Minedit's menu bar, click on *File* then *open* to load the lab's topology. Locate the *Lab7.mn* topology file in the default directory, */home/frr/BGP_Labs/lab7* and click on *Open*.

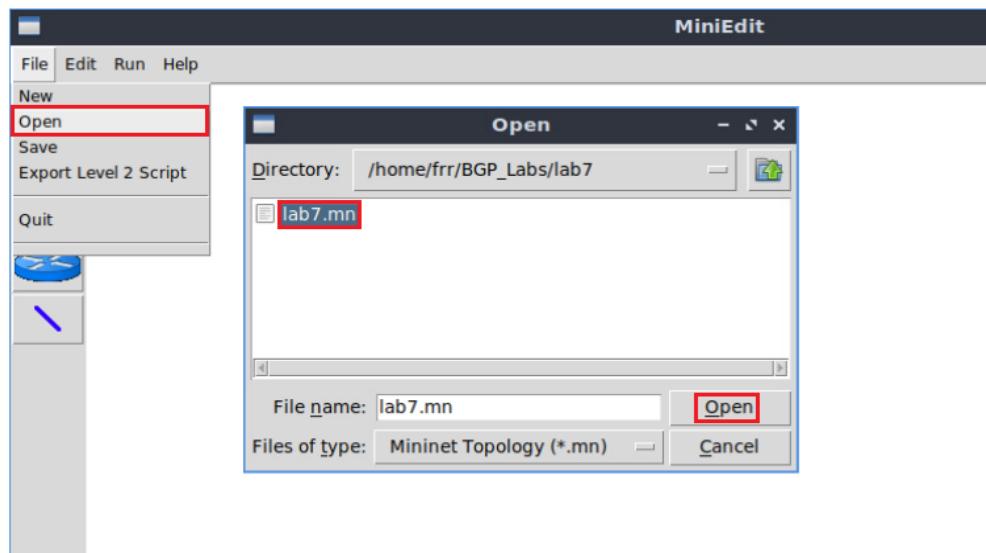


Figure 7. MiniEdit's Open dialog.

At this point the topology is loaded with all the required network components. You will execute a script that will load the configuration of the routers.

Step 3. Open the Linux terminal.

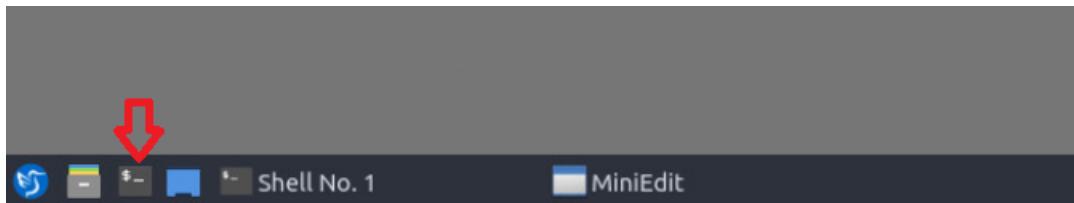


Figure 8. Opening Linux terminal.

Step 4. Click on the Linux's terminal and navigate into *BGP_Labs/lab7* 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 BGP_Labs/lab7
```

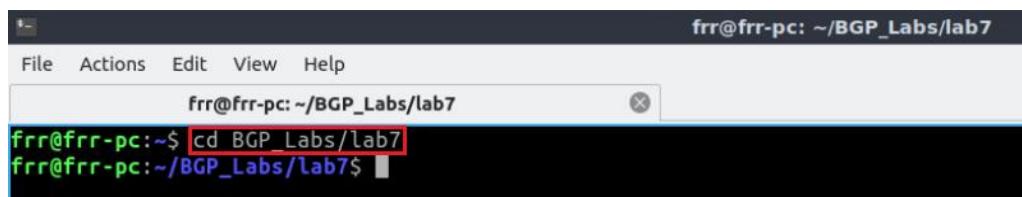
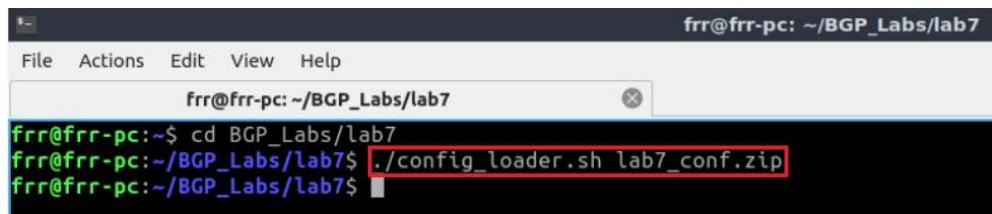


Figure 9. Entering to the *BGP_Labs/lab7* directory.

Step 5. To execute the shell script, type the following command. The argument of the program corresponds to the configuration zip file that will be loaded in all the routers in the topology.

```
./config_loader.sh lab7_conf.zip
```

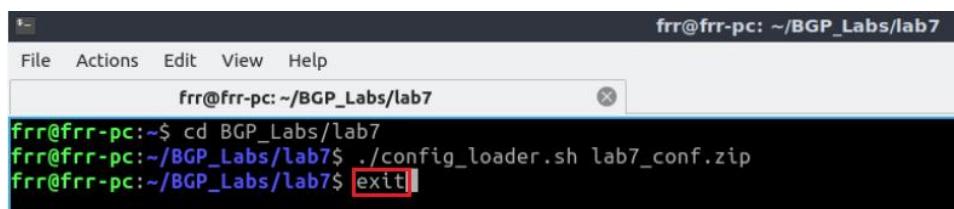


A terminal window titled "frr@frr-pc: ~/BGP_Labs/lab7". The command "cd BGP_Labs/lab7" is entered, followed by the command "./config_loader.sh lab7_conf.zip". The entire command line is highlighted with a red box.

Figure 10. Executing the shell script to load the configuration.

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

```
exit
```



A terminal window titled "frr@frr-pc: ~/BGP_Labs/lab7". The command "cd BGP_Labs/lab7" is entered, followed by the command "./config_loader.sh lab7_conf.zip", and finally the command "exit". The entire command line is highlighted with a red box.

Figure 11. Exiting from the terminal.

Step 7. At this point hosts h1, h2 and h3 interfaces are configured. To proceed with the emulation, click on the *Run* button located in lower left-hand side.



Figure 12. Starting the emulation.

Step 8. Click on Mininet's terminal, i.e., the one launched when MiniEdit was started.

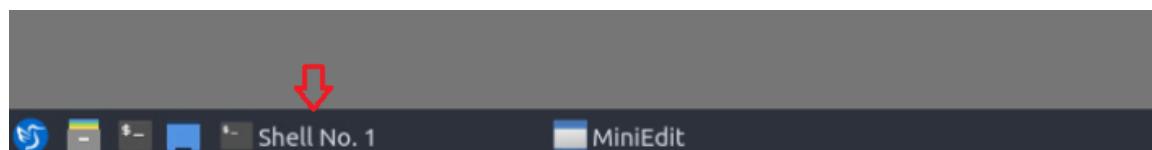


Figure 13. Opening Mininet's terminal.

Step 9. Issue the following command to display the interface names and connections.

```
links
```

```
File Actions Edit View Help
Shell No. 1
mininet> links
s1-eth1<->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)
r1-eth1<->r2-eth1 (OK OK)
r2-eth2<->r3-eth1 (OK OK)
h1-eth0<->s1-eth2 (OK OK)
mininet>
```

Figure 14. Displaying network interfaces.

In Figure 14, the link displayed within the gray box indicates that interface eth1 of switch s1 connects to interface eth0 of router r1 (i.e., $s1\text{-}eth1 <-> r1\text{-}eth0$).

2.3 Load zebra daemon and Verify the Connectivity

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

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.

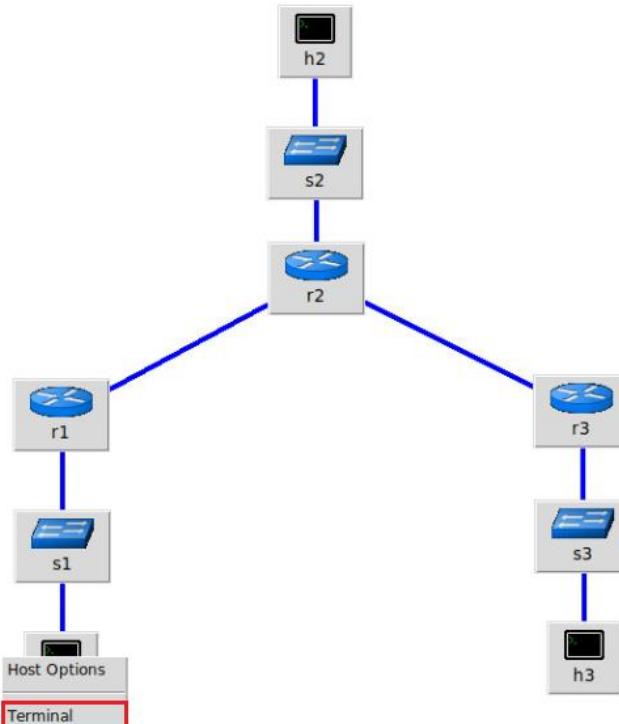


Figure 15. Opening a terminal on host h1.

Step 2. On 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.

```
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 16. Output of `ifconfig` command.

Step 3. On 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:~# 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 17. Output of `route` command.

Step 4. In order to verify hosts h2 and h3, proceed similarly by repeating from step 1 to step 3 on hosts h2 and h3 terminals. Similar results should be observed.

Step 5. You will validate that the router interfaces are configured correctly according to Table 2. In order to verify router r1, hold right-click on router r1 and select Terminal.

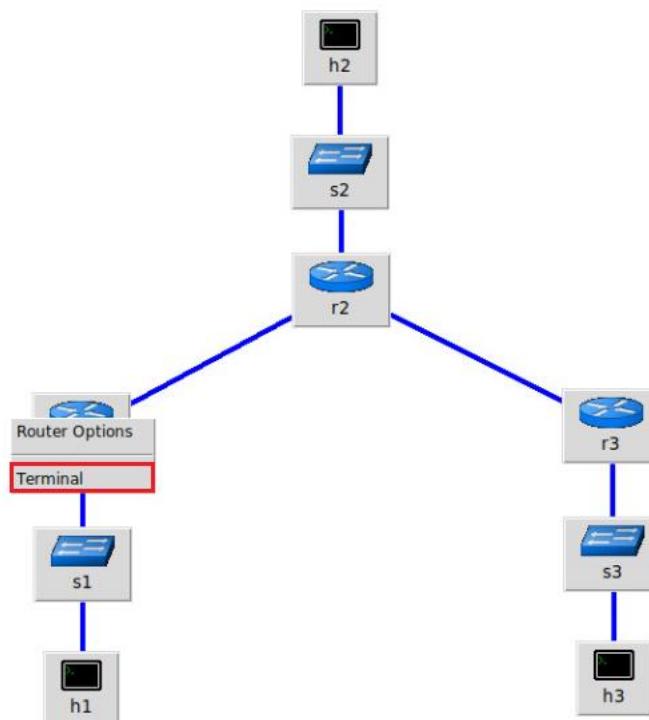


Figure 18. Opening a terminal on router r1.

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

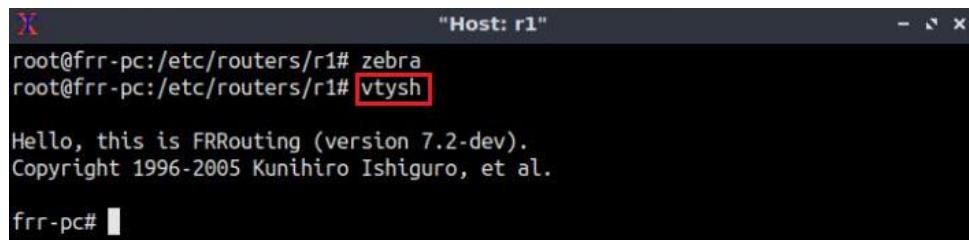
```
zebra
```

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

Figure 19. Starting zebra daemon.

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

```
vtysh
```

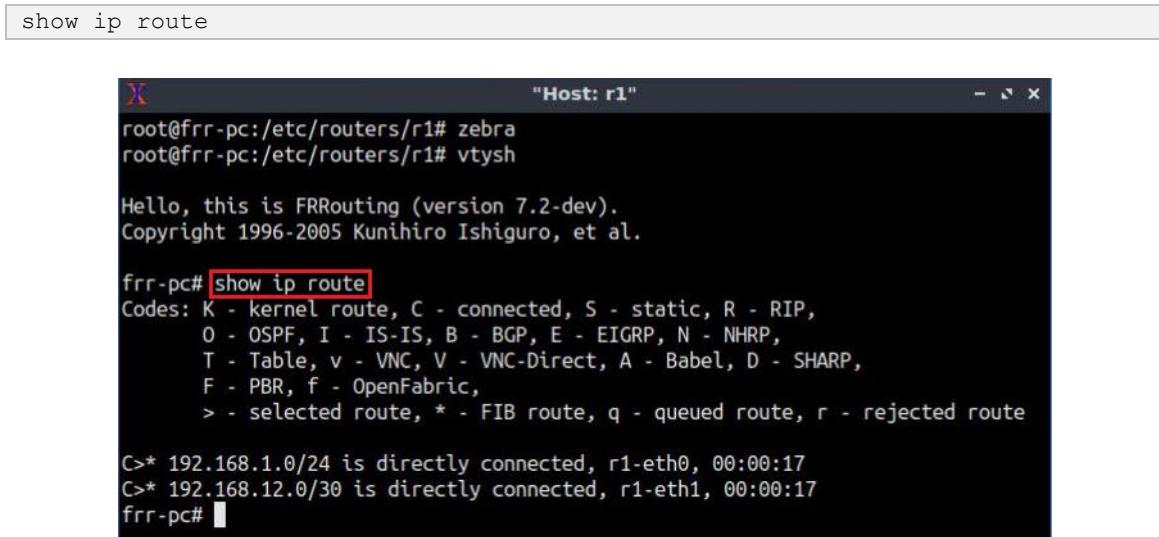


```
"Host: r1"
root@frr-pc:/etc/routers/r1# zebra
root@frr-pc:/etc/routers/r1# vtysh
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc#
```

Figure 20. 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 networks attached to routers r2 (192.168.2.0/24) and router r3 (192.168.3.0/24) as there is no routing protocol configured yet.



```
show ip route
```



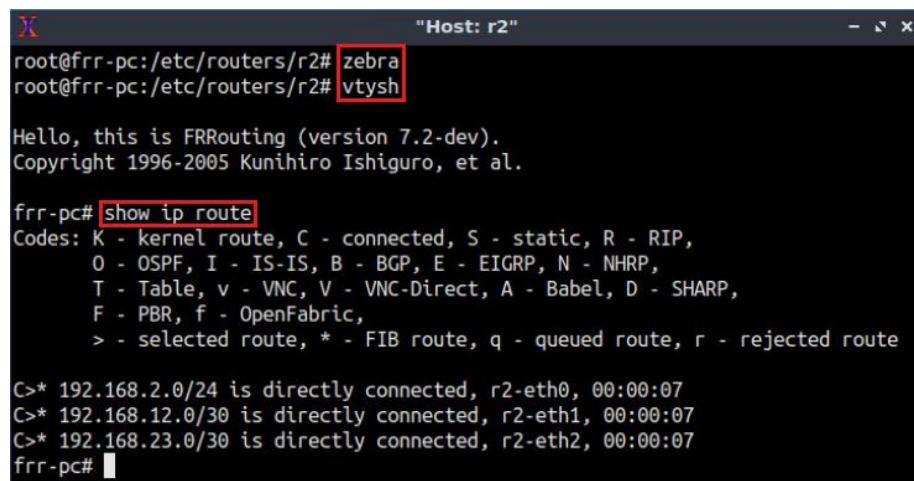
```
"Host: r1"
root@frr-pc:/etc/routers/r1# zebra
root@frr-pc:/etc/routers/r1# vtysh
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       0 - 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, 00:00:17
C>* 192.168.12.0/30 is directly connected, r1-eth1, 00:00:17
frr-pc#
```

Figure 21. Displaying routing table of router r1.

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,
       0 - 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, 00:00:07
C>* 192.168.12.0/30 is directly connected, r2-eth1, 00:00:07
C>* 192.168.23.0/30 is directly connected, r2-eth2, 00:00:07
frr-pc#
```

Figure 22. Displaying routing table of router r2.

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

```

root@frr-pc:/etc/routers/r3# zebra
root@frr-pc:/etc/routers/r3# 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>* 192.168.3.0/24 is directly connected, r3-eth0, 00:00:07
C>* 192.168.23.0/30 is directly connected, r3-eth1, 00:00:07
frr-pc# 

```

Figure 23. Displaying routing table of router r3.

3 Configure BGP on all routers

In this section, you will configure EBGP on the routers that are hosted in different ASes. You will assign BGP neighbors to allow the routers to exchange BGP routes. Furthermore, routers r1, r2, and r3 will advertise their LANs via BGP so that the LANs are learned by peer routers.

Step 1. To configure BGP routing protocol, you need to enable the BGP daemon first. On router r1, type the following command to exit the vtysh session:

```
exit
```

```

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

```

Figure 24. Exiting the vtysh session.

Step 2. Type the following command on router r1 terminal to enable and to start BGP routing protocol.

```
bgpd
```

```

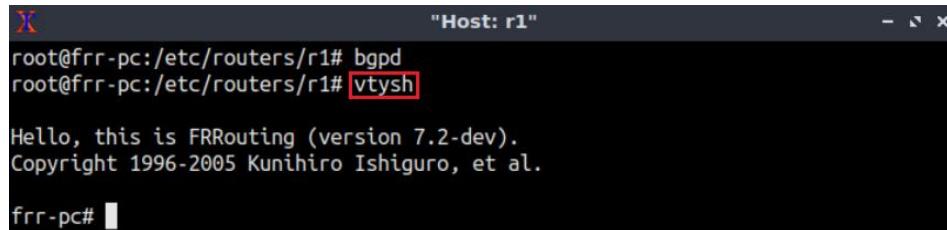
root@frr-pc:/etc/routers/r1# bgpd
root@frr-pc:/etc/routers/r1# 

```

Figure 25. Starting BGP daemon.

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

```
vtysh
```

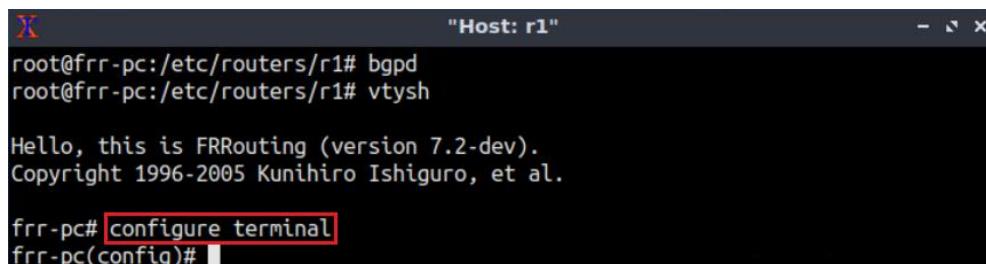


A terminal window titled "Host: r1". The command "vtysh" is entered at the root prompt "root@frr-pc:/etc/routers/r1#". The output shows the FRRouting version and copyright information, followed by the prompt "frr-pc#".

Figure 26. Starting vtysh on router r1.

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

```
configure terminal
```

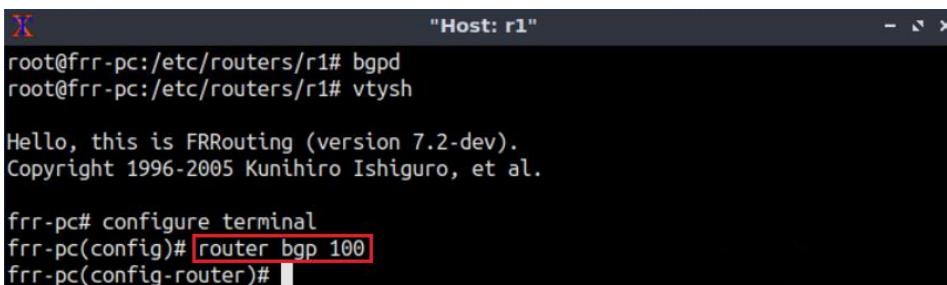


A terminal window titled "Host: r1". The command "configure terminal" is entered at the "frr-pc#" prompt. The output shows the FRRouting version and copyright information, followed by the prompt "frr-pc(config)#".

Figure 27. Enabling configuration mode on router r1.

Step 5. The ASN assigned for router r1 is 100. In order to configure BGP, type the following command:

```
router bgp 100
```

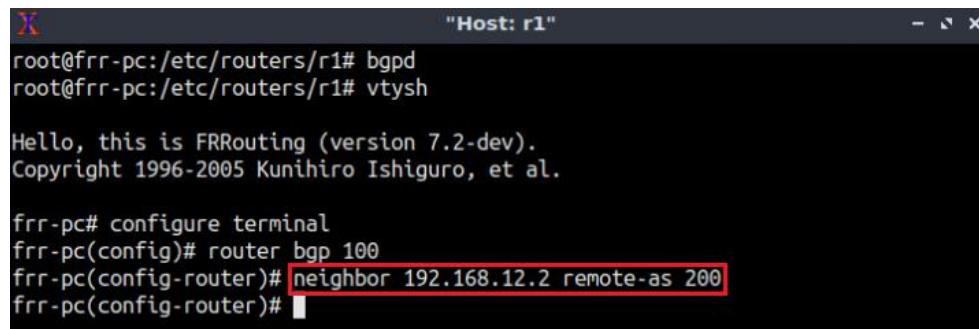


A terminal window titled "Host: r1". The command "router bgp 100" is entered at the "frr-pc(config-router)" prompt. The output shows the FRRouting version and copyright information, followed by the prompt "frr-pc(config-router)#".

Figure 28. Configuring BGP on router r1.

Step 6. To configure a BGP neighbor to router r1 (AS 100), type the command shown below. This command specifies the neighbor IP address (192.168.12.2) and the ASN of the remote BGP peer (AS 200).

```
neighbor 192.168.12.2 remote-as 200
```



```
"Host: r1"
root@frrr-pc:/etc/routers/r1# bgpd
root@frrr-pc:/etc/routers/r1# vtysh

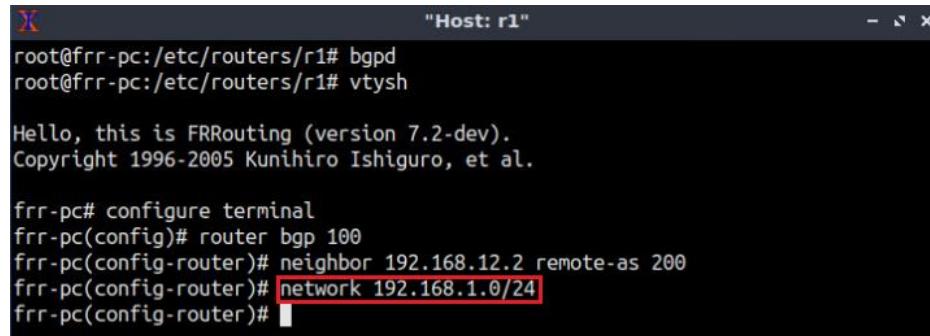
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frrr-pc# configure terminal
frrr-pc(config)# router bgp 100
frrr-pc(config-router)# neighbor 192.168.12.2 remote-as 200
frrr-pc(config-router)#[ ]
```

Figure 29. Assigning BGP neighbor to router r1.

Step 7. In this step, router r1 will advertise the LAN 192.168.1.0/24 to its BGP peers. To do so, issue the following command:

```
network 192.168.1.0/24
```



```
"Host: r1"
root@frrr-pc:/etc/routers/r1# bgpd
root@frrr-pc:/etc/routers/r1# vtysh

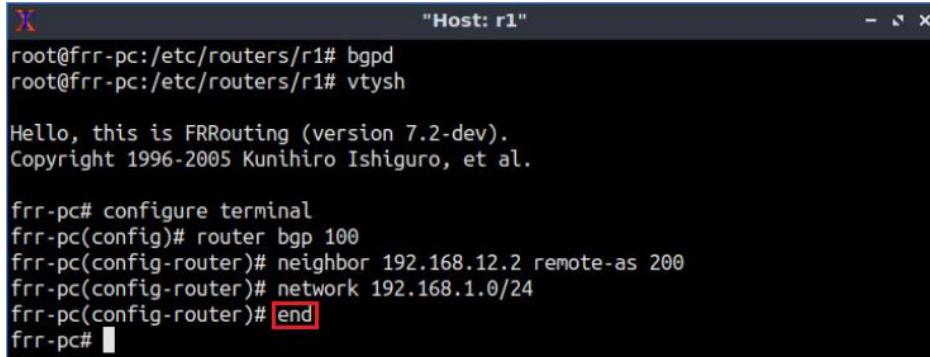
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frrr-pc# configure terminal
frrr-pc(config)# router bgp 100
frrr-pc(config-router)# neighbor 192.168.12.2 remote-as 200
frrr-pc(config-router)# network 192.168.1.0/24
frrr-pc(config-router)#[ ]
```

Figure 30. Advertising local network on router r1.

Step 8. Type the following command to exit from configuration mode.

```
end
```



```
"Host: r1"
root@frrr-pc:/etc/routers/r1# bgpd
root@frrr-pc:/etc/routers/r1# vtysh

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

frrr-pc# configure terminal
frrr-pc(config)# router bgp 100
frrr-pc(config-router)# neighbor 192.168.12.2 remote-as 200
frrr-pc(config-router)# network 192.168.1.0/24
frrr-pc(config-router)# end
frrr-pc#[ ]
```

Figure 31. Exiting from configuration mode.

Step 9. Type the following command to verify BGP networks. You will observe the LAN network of router r1.

```
show ip bgp
```

```
frr-pc# show ip bgp
BGP table version is 1, local router ID is 192.168.12.1, vrf id 0
Default local pref 100, local AS 100
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*-> 192.168.1.0/24    0.0.0.0                  0        32768 i

Displayed 1 routes and 1 total paths
frr-pc#
```

Figure 32. Verifying BGP networks on router r1.

Step 10. Type the following command to verify BGP neighbors. You will verify that the neighbor IP address is 192.168.12.2. The corresponding ASN is 200.

show ip bgp neighbors

```
frr-pc# show ip bgp neighbors
BGP neighbor is [192.168.12.2], remote AS 200, local AS 100, external link
  BGP version 4, remote router ID 0.0.0.0, local router ID 192.168.12.1
  BGP state = Active
  Last read 00:04:15, Last write never
  Hold time is 180, keepalive interval is 60 seconds
  Message statistics:
    Inq depth is 0
    Outq depth is 0
              Sent      Rcvd
  Opens:          0          0
  Notifications: 0          0
  Updates:        0          0
  Keepalives:     0          0
  Route Refresh: 0          0
  Capability:    0          0
  Total:          0          0
  Minimum time between advertisement runs is 0 seconds

  For address family: IPv4 Unicast
  Not part of any update group
  Community attribute sent to this neighbor(all)
  0 accepted prefixes
```

Figure 33. Verifying BGP neighbors on router r1.

Step 11. Follow from step 1 to step 8 but with different metrics in order to configure BGP on router r2. All these steps are summarized in the following figure.

The terminal window shows the configuration of BGP on router r2. The configuration includes defining a BGP router ID (200), specifying neighbors (192.168.12.1 and 192.168.23.2) with their respective remote AS numbers (100 and 65000), and advertising a local network (192.168.2.0/24). The configuration is enclosed in a red box.

```
frr-pc# exit  
root@frr-pc:/etc/routers/r2# bgpd  
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 bgp 200  
frr-pc(config-router)# neighbor 192.168.12.1 remote-as 100  
frr-pc(config-router)# neighbor 192.168.23.2 remote-as 65000  
frr-pc(config-router)# network 192.168.2.0/24  
frr-pc(config-router)# end  
frr-pc#
```

Figure 34. Configuring BGP on router r2.

Step 12. Follow from step 1 to step 8 but with different metrics in order to configure BGP on router r3. All these steps are summarized in the following figure.

The terminal window shows the configuration of BGP on router r3. The configuration includes defining a BGP router ID (65000), specifying a neighbor (192.168.23.1) with its remote AS number (200), and advertising a local network (192.168.3.0/24). The configuration is enclosed in a red box.

```
frr-pc# exit  
root@frr-pc:/etc/routers/r3# bgpd  
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 bgp 65000  
frr-pc(config-router)# neighbor 192.168.23.1 remote-as 200  
frr-pc(config-router)# network 192.168.3.0/24  
frr-pc(config-router)# end  
frr-pc#
```

Figure 35. Configuring BGP on router r3.

Step 13. In router r2 terminal, type the following command to verify the routing table of router r2. The LANs of router r1 (192.168.1.0/24) and router r3 (192.168.3.0/24) are advertised to router r2 through EBGP.

The terminal window shows the output of the 'show ip route' command on router r2. It lists several routes, including the direct connections to r1 and r3, and the EBGP routes learned from r1 and r3. The output is enclosed in a red box.

```
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  
  
B>* 192.168.1.0/24 [20/0] via 192.168.12.1, r2-eth1, 00:13:28  
C>* 192.168.2.0/24 is directly connected, r2-eth0, 00:16:04  
B>* 192.168.3.0/24 [20/0] via 192.168.23.2, r2-eth2, 00:02:39  
C>* 192.168.12.0/30 is directly connected, r2-eth1, 00:16:04  
C>* 192.168.23.0/30 is directly connected, r2-eth2, 00:16:04  
frr-pc#
```

Figure 36. Verifying the routing table of router r2.

4 Remove the private ASN

BGP private ASNs are not globally unique. If a BGP router receives a route in which its own ASN is part of the AS_PATH attribute, it does not accept the route. ISP needs to ensure they remove private ASN from BGP updates to EBGP peers when announcing routing information across the Internet.

At this point, router r1 can't reach the LAN of router r3 (192.168.3.0/24), since the private ASN exists in the advertised AS_PATH attribute. In this section, you will configure the ISP so that it does not advertise the private ASN of the customer.

Step 1. In router r1 terminal, perform a connectivity test by running the command shown below. To stop the test, press **Ctrl+c**. The result will show a successful connectivity test between router r1 and host h2.

```
ping 192.168.2.10
```

```
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=0.444 ms
64 bytes from 192.168.2.10: icmp_seq=2 ttl=63 time=0.075 ms
64 bytes from 192.168.2.10: icmp_seq=3 ttl=63 time=0.088 ms
^C
--- 192.168.2.10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 29ms
rtt min/avg/max/mdev = 0.075/0.202/0.444/0.171 ms
frr-pc#
```

Figure 37. Connectivity test using **ping** command.

Step 2. Test the connectivity between router r1 and host h3 using **ping** command as specified below. To stop the test, press **Ctrl+c**. Router r1 cannot reach host h3 since the private ASN (65000) is part of the AS_PATH attribute of this route (192.168.3.0/24).

```
ping 192.168.3.10
```

```
frr-pc# ping 192.168.3.10
PING 192.168.3.10 (192.168.3.10) 56(84) bytes of data.
^C
--- 192.168.3.10 ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 126ms
frr-pc#
```

Figure 38. Connectivity test using **ping** command.

Step 3. Type the following command to verify the BGP table of router r1. ASN 65000 is listed in the path to network 192.168.3.0/24. If router r1 wants to communicate with host h3 through 192.168.12.2, router r3 will discard the route as its own ASN is a part of the AS_PATH attribute. The private ASN should be removed in order to communicate with router r3.

```
show ip bgp
```

The terminal window shows the output of the 'show ip bgp' command. It displays the BGP table version, local router ID, and various status codes. The routes listed are:

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 192.168.1.0/24	0.0.0.0	0		32768	i
*> 192.168.2.0/24	192.168.12.2	0		0 200	i
*> [192.168.3.0/24]	[192.168.12.2]			0 200	[65000] i

Displayed 3 routes and 3 total paths

Figure 39. Verifying BGP table of router r1.

Step 4. In router r2 terminal, type the following command to enable the configuration mode:

```
configure terminal
```

The terminal window shows the 'configure terminal' command being entered. The prompt changes to 'frr-pc(config)#'. The command is highlighted with a red box.

Figure 40. Enabling configuration mode on router r2.

Step 5. Type the following command to enable BGP configuration mode.

```
router bgp 200
```

The terminal window shows the 'router bgp 200' command being entered. The prompt changes to 'frr-pc(config-router)#'. The command is highlighted with a red box.

Figure 41. Entering to BGP configuration mode.

Step 6. Type the following command to remove the private ASN from the BGP routes that are exchanged with router r1.

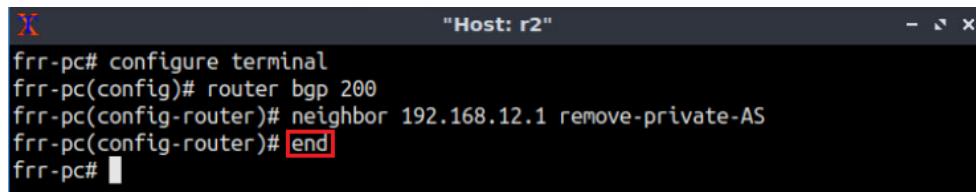
```
neighbor 192.168.12.1 remove-private-AS
```

The terminal window shows the 'neighbor 192.168.12.1 remove-private-AS' command being entered. The command is highlighted with a red box. The prompt changes to 'frr-pc(config-router)#'.

Figure 42. Removing private AS from r1 route.

Step 7. Type the following command to exit from configuration mode.

```
end
```



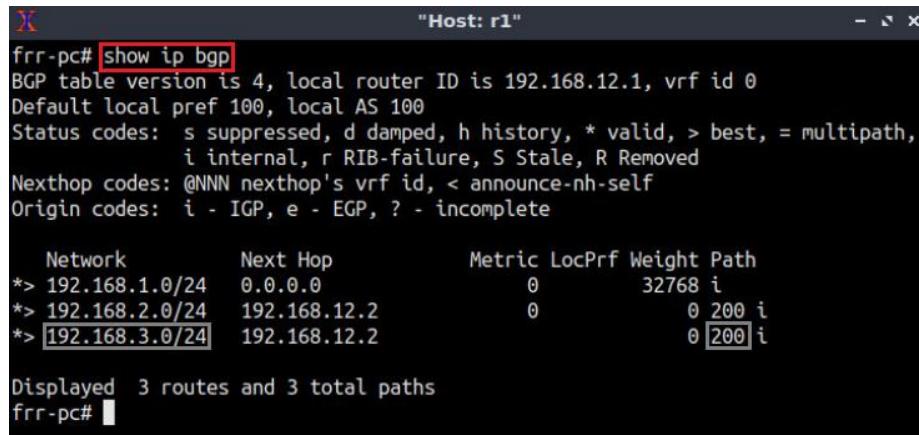
A terminal window titled "Host: r2". The command history shows:

```
frr-pc# configure terminal
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.12.1 remove-private-AS
frr-pc(config-router)# end
frr-pc#
```

Figure 43. Exiting from configuration mode.

Step 8. Type the following command to verify the BGP table of router r1. The path to network 192.168.3.0/24 will include AS 200 only. The private ASN (65000) is no longer included in the AS_PATH attribute.

```
show ip bgp
```



A terminal window titled "Host: r1". The command history shows:

```
frr-pc# show ip bgp
```

The output displays the BGP table:

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 192.168.1.0/24	0.0.0.0	0		32768	i
*> 192.168.2.0/24	192.168.12.2	0		0	200 i
*> [192.168.3.0/24]	192.168.12.2	0		0	[200]i

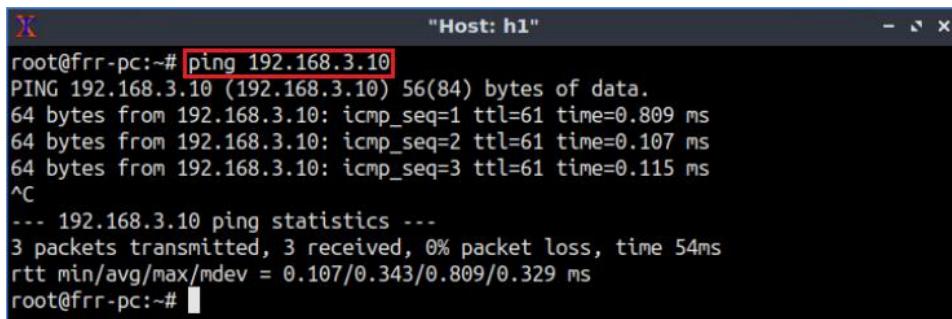
Displayed 3 routes and 3 total paths

```
frr-pc#
```

Figure 44. Displaying BGP table of router r1.

Step 9. Test the connectivity between the end-hosts using the `ping` command. On host h1, type the command specified below. This command tests the connectivity between host h1 and host h3. To stop the test, press `Ctrl+c`. The figure below shows a successful connectivity test.

```
ping 192.168.3.10
```



A terminal window titled "Host: h1". The command history shows:

```
root@frr-pc:~# ping 192.168.3.10
```

The output shows the ping results:

```
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=61 time=0.809 ms
64 bytes from 192.168.3.10: icmp_seq=2 ttl=61 time=0.107 ms
64 bytes from 192.168.3.10: icmp_seq=3 ttl=61 time=0.115 ms
^C
--- 192.168.3.10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 54ms
rtt min/avg/max/mdev = 0.107/0.343/0.809/0.329 ms
```

```
root@frr-pc:~#
```

Figure 45. Connectivity test using `ping` command.

5 Use the AS_PATH attribute to filter routes

In this section, you will filter the advertised routes based on their AS_PATH attribute. In this case, the Customer (AS 65000) does not need to receive routing updates from the Campus network (AS 100). You will configure the ISP so that it does not advertise any route that originates from AS 100 to AS 65000.

AS_PATH ACLs can filter the advertised routes based on their AS_PATH attribute using regular expressions. Regular expressions are used to search for a substring within a text; for example, to search for a specific ASN in an AS_PATH attribute list.

5.1 Configure AS_PATH ACL

Step 1. In router r2 terminal, type the following command to enable the configuration mode:

```
configure terminal
```

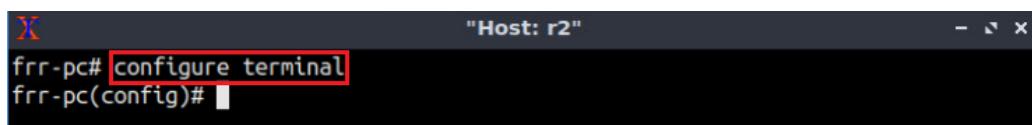


Figure 46. Enabling configuration mode on router r2.

Step 2. In this step, you will configure an AS_PATH ACL so that it does not advertise the updates coming from AS 100. Type the following command to configure an ACL to match BGP routes with an AS_PATH attribute that both begins and ends with the number 100. An ACL number can be selected within the range 1-99. You will use 1 as the ACL number in this lab. Use `deny` so that router r2 does not advertise any update coming from AS 100 to router r3. The character `^` indicates that the AS_PATH must begin with the given number 100. The `$` character indicates that the AS_PATH attribute must also end with 100. Essentially, this statement matches only paths that are sourced from AS 100.

```
bgp as-path access-list 1 deny ^100$
```

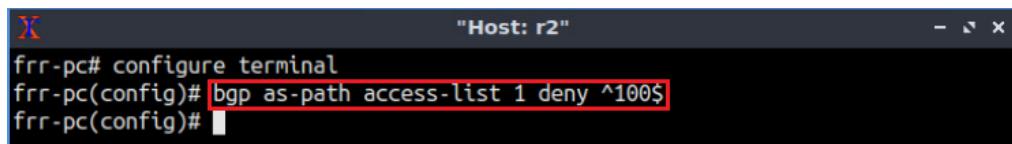
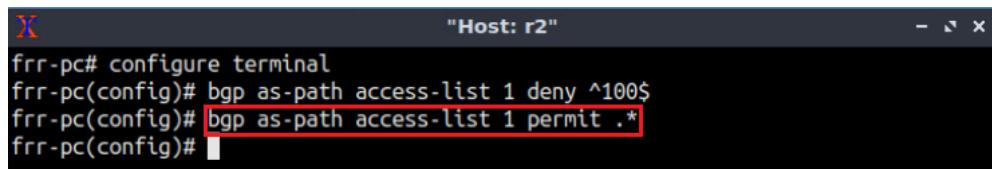


Figure 47. Configuring an AS_PATH ACL on router r2.

Step 5. Type the command shown below. The characters `.*` matches any value of the AS_PATH attribute, which in effect permits any update that has not been denied by the previous ACL statement.

```
bgp as-path access-list 1 permit .*
```

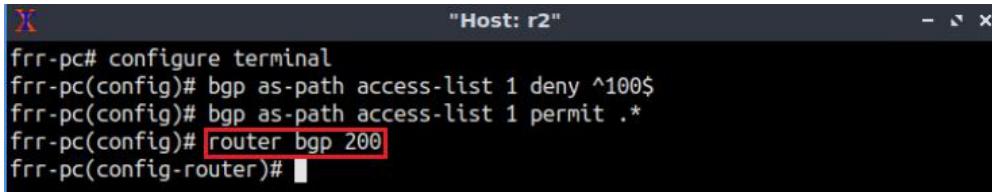


```
frr-pc# configure terminal  
frr-pc(config)# bgp as-path access-list 1 deny ^100$  
frr-pc(config)# bgp as-path access-list 1 permit .*  
frr-pc(config)#[ ]
```

Figure 48. Configuring access-list on router r2.

Step 6. Type the following command to enter BGP configuration mode:

```
router bgp 200
```

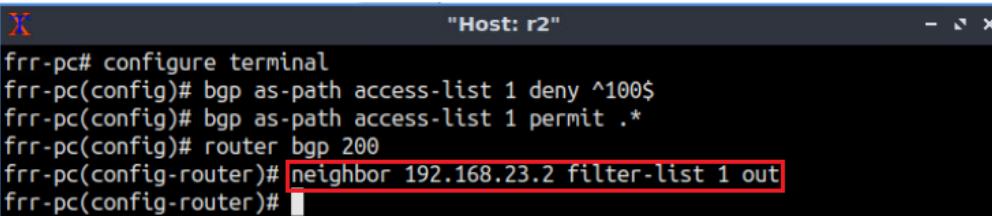


```
frr-pc# configure terminal  
frr-pc(config)# bgp as-path access-list 1 deny ^100$  
frr-pc(config)# bgp as-path access-list 1 permit .*  
frr-pc(config)# router bgp 200  
frr-pc(config-router)#[ ]
```

Figure 49. Configuring BGP on router r2.

Step 7. Set up the configured ACL (ACL 1) to filter the BGP routes that are sent to the router r3 neighbor (192.168.23.2). To do so, type the following command:

```
neighbor 192.168.23.2 filter-list 1 out
```

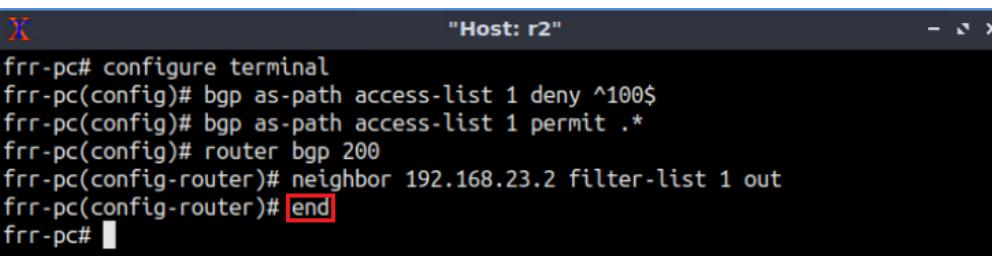


```
frr-pc# configure terminal  
frr-pc(config)# bgp as-path access-list 1 deny ^100$  
frr-pc(config)# bgp as-path access-list 1 permit .*  
frr-pc(config)# router bgp 200  
frr-pc(config-router)# neighbor 192.168.23.2 filter-list 1 out  
frr-pc(config-router)#[ ]
```

Figure 50. Configuring BGP filter-list on router r2.

Step 8. Type the following command to exit from configuration mode.

```
end
```



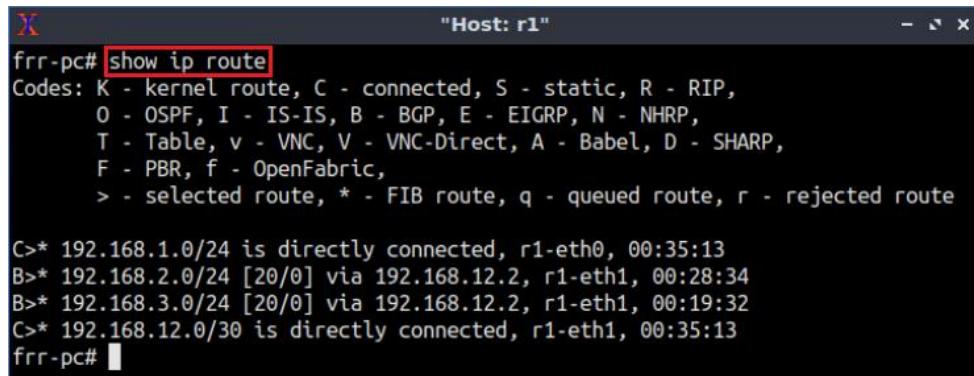
```
frr-pc# configure terminal  
frr-pc(config)# bgp as-path access-list 1 deny ^100$  
frr-pc(config)# bgp as-path access-list 1 permit .*  
frr-pc(config)# router bgp 200  
frr-pc(config-router)# neighbor 192.168.23.2 filter-list 1 out  
frr-pc(config-router)# end  
frr-pc#[ ]
```

Figure 51. Ending the configuration on router r2.

5.2 Verify Configuration

Step 1. Type the following command to verify the routing table of router r1. The routing table has a route to router r3 network (192.168.3.0/24). Router r2 applied the ACL to router r3 only, thus, router r1 will keep receiving routing updates from router r3.

```
show ip route
```



A terminal window titled "Host: r1" displaying the output of the "show ip route" command. The output shows several routes, including a direct connection to 192.168.12.0/30 via r1-eth1 at 00:35:13, and routes to 192.168.1.0/24, 192.168.2.0/24, and 192.168.3.0/24 via r1-eth1 at 00:28:34, 00:19:32, and 00:35:13 respectively. The command "show ip route" is highlighted with a red box.

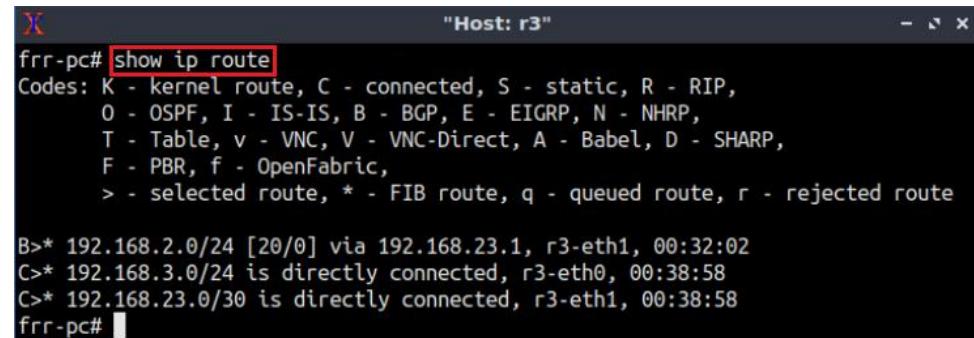
```
frr-pc# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       0 - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued route,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

C>* 192.168.1.0/24 is directly connected, r1-eth0, 00:35:13
B>* 192.168.2.0/24 [20/0] via 192.168.12.2, r1-eth1, 00:28:34
B>* 192.168.3.0/24 [20/0] via 192.168.12.2, r1-eth1, 00:19:32
C>* 192.168.12.0/30 is directly connected, r1-eth1, 00:35:13
frr-pc#
```

Figure 52. Displaying the routing table of router r1.

Step 2. Type the following command to verify the routing table of router r3. The routing table of router r3 should not have a route to network 192.168.1.0/24, since router r2 does not advertise any routing update to router r3 that are sent from AS 100.

```
show ip route
```



A terminal window titled "Host: r3" displaying the output of the "show ip route" command. The output shows a direct connection to 192.168.2.0/24 via r3-eth1 at 00:32:02, and routes to 192.168.3.0/24 and 192.168.23.0/30 via r3-eth0 at 00:38:58. The command "show ip route" is highlighted with a red box.

```
frr-pc# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       0 - 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

B>* 192.168.2.0/24 [20/0] via 192.168.23.1, r3-eth1, 00:32:02
C>* 192.168.3.0/24 is directly connected, r3-eth0, 00:38:58
C>* 192.168.23.0/30 is directly connected, r3-eth1, 00:38:58
frr-pc#
```

Figure 53. Displaying the routing table of router r3.

Step 3. To verify that the filter is working properly, type the following command. It will display routes that match the specified regular expression. The network 192.168.1.0/24 should appear in the list (shown within the gray box).

```
show ip bgp regexp ^100$
```

```
frr-pc# show ip bgp regexp ^100$  
BGP table version is 3, local router ID is 192.168.23.1, vrf id 0  
Default local pref 100, local AS 200  
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,  
i internal, r RIB-failure, S Stale, R Removed  
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self  
Origin codes: i - IGP, e - EGP, ? - incomplete  
  
Network Next Hop Metric LocPrf Weight Path  
*> 192.168.1.0/24 192.168.12.1 0 0 100 i  
  
Displayed 1 routes and 3 total paths  
frr-pc#
```

Figure 54. Verifying BGP filter on router r2.

Figure 54 displays the routes that match the specified regular expression. The network 192.168.1.0/24 should appear in the list.

Step 4. On host h1 terminal, perform a connectivity between host h1 and host h2 by issuing the command shown below. To stop the test, press **Ctrl+c**. The result will show a successful connectivity test.

```
ping 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=0.766 ms  
64 bytes from 192.168.2.10: icmp_seq=2 ttl=62 time=0.092 ms  
64 bytes from 192.168.2.10: icmp_seq=3 ttl=62 time=0.098 ms  
^C  
--- 192.168.2.10 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 16ms  
rtt min/avg/max/mdev = 0.092/0.318/0.766/0.316 ms  
root@frr-pc:~#
```

Figure 55. Connectivity test using **ping** command.

Step 5. Test connectivity between host h1 and host h3 using the **ping** command. To stop test, press **Ctrl+c**.

```
ping 192.168.3.10
```

```
root@frr-pc:~# ping 192.168.3.10  
PING 192.168.3.10 (192.168.3.10) 56(84) bytes of data.  
^C  
--- 192.168.3.10 ping statistics ---  
5 packets transmitted, 0 received, 100% packet loss, time 98ms  
root@frr-pc:~#
```

Figure 56. Connectivity test using **ping** command.

Consider figure 56, host h1 cannot reach host h3 due to the configured route filter.

This concludes Lab 7. Stop the emulation and then exit out of MiniEdit.

References

1. G. Huston, "Exploring Autonomous System Numbers", 2005, [Online] Available: <http://wattle.apnic.net/ispcol/2005-08/as.pdf>
2. IANA, "Special-Purpose Autonomous System (AS) Numbers", 2015, [Online] Available: <https://www.iana.org/assignments/iana-as-numbers-special-registry/iana-as-numbers-special-registry.xhtml>
3. APNIC, "Autonomous System numbers - FAQs", 2020, [Online] Available: <https://www.apnic.net/get-ip/faqs/asn/>
4. J. Kurose, K. Ross, "Computer networking, a top-down approach," 7th Edition, Pearson, 2017.
5. Cisco, "BGP Best Path Selection Algorithm", 2016, [Online] Available: <https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/13753-25.html#anc2>
6. Cisco, "Removing Private Autonomous System Numbers in BGP", [Online] Available: <https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/13756-32.html>
7. Cisco, "Security Configuration Guide: Access Control Lists, Cisco IOS XERelease 3S", 2015 [Online] Available: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_data_acl/configuration/xe-3s/sec-data-acl-xe-3s-book.pdf
8. Cisco, "Implementing Cisco IP Routing (ROUTE) Foundation Learning Guide", Pearson, 2015.



BORDER GATEWAY PROTOCOL

Lab 8: Configuring IBGP and EBGP Sessions, Local Preference, and MED

Document Version: **2-19-2020**



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 Intradomain and Interdomain routing protocols.....	4
1.2 LOCAL_PREF attribute	4
1.3 MED attribute.....	5
2 Lab topology.....	6
2.1 Lab settings.....	7
2.2 Open topology and load the configuration.....	8
2.3 Load zebra daemon and Verify Connectivity	11
3 Configure OSPF on router r2 and router r3	15
4 Configure BGP on all routers	19
4.1 Configure EBGP on router r1.....	19
4.2 Configure EBGP and IBGP on router r2 and router r3	23
4.3 Advertise networks on router r2 and router r3 through OSPF	26
5 Set and verify BGP LOCAL_PREF on router r2 and router r3	29
6 Set and verify BGP MED on router r2 and router r3.....	33
6.1 Set BGP MED on router r2 and router r3	33
6.2 Verify Configuration	36
7 Verify the secondary link	37
References	39

Overview

This lab discusses the Local Preference (LOCAL_PREF) and Multi Exit Discriminator (MED) attributes used in the Border Gateway Protocol (BGP). In this lab, BGP and Open Shortest Path First (OSPF) will be configured as the Exterior Gateway Protocol (EGP) and the Interior Gateway Protocol (IGP), respectively. BGP is referred to as External BGP (EBGP) when it is running between different Autonomous Systems (ASes), whereas it is referred to as Internal BGP (IBGP) when it is running within an AS.

Objectives

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

1. Configure OSPF as the EGP.
2. Configure and verify EBGP and IBGP.
3. Set the BGP LOCAL_PREF attribute to set route preferences.
4. Use the BGP MED attribute to establish a traffic filter.

Lab settings

The information in Table 1 provides the credentials to access Client1 machine.

Table 1. Credentials to access Client1 machine.

Device	Account	Password
Client1	admin	password

Lab roadmap

This lab is organized as follows:

1. Section 1: Introduction.
2. Section 2: Lab topology.
3. Section 3: Configure OSPF on router r2 and router r3.
4. Section 4: Configure IBGP on router r2 and router r3.
5. Section 5: Configure and verify EBGP on all routers.
6. Section 6: Set and verify BGP LOCAL_PREF on router r2 and router r3.
7. Section 7: Set and verify BGP MED on router r2 and router r3.

1. Introduction

1.1 Intradomain and Interdomain routing protocols

The Internet consists of many independent administrative domains, referred to as ASes. ASes are operated by different organizations, which can run their own internal routing protocols. A routing protocol that runs within an AS is referred to as intradomain routing protocol. One of the most widely used intradomain protocols is OSPF. Since an AS may be large and nontrivial to manage, OSPF allows an AS to be divided into numbered areas¹. An area is a logical collection of networks, routers, and links. All routers in the same area have detailed information of the topology within their area².

A routing protocol that runs between ASes is referred to as interdomain routing protocol. ASes may use different intradomain routing protocols; however, they must use the same interdomain routing protocol, i.e., BGP. BGP allows the enforcement of different routing policies on the traffic from one AS to another. For example, a security policy can prevent the dissemination of routing information from one AS to another¹.

BGP is referred to as External BGP (EBGP) when it is running between different ASes, whereas it is referred to as Internal BGP (IBGP) when it is running within an AS¹. IBGP is usually used to distribute the routes learned using EBGP among the routers within the same AS¹.

Consider Figure 1. The intradomain routing protocol within AS 100 is OSPF, and the interdomain routing protocol between AS 100 and AS 200 is BGP (EBGP). Routers within the same AS advertise their EBGP learned routes among each other through IBGP.

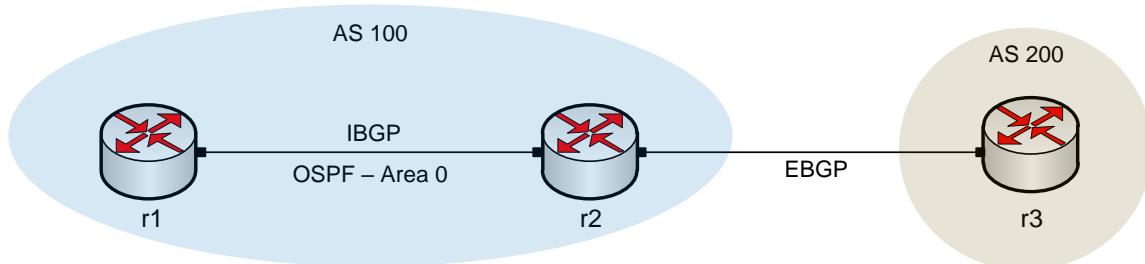


Figure 1. Routers that exchange information within the same AS use OSPF and IBGP, while routers that exchange information between different ASes use EBGP.

1.2 LOCAL_PREF attribute

In BGP, when a router advertises an IP address across a BGP session (i.e., between two routers running BGP), it comes with some BGP attributes³. These attributes help BGP select the best path when there are multiple paths to the same destination⁴.

A router sending IBGP update packets, will include the LOCAL_PREF attribute. This attribute indicates the degree of preference for one BGP route over the others, when an AS has multiple routes to another AS. The LOCAL_PREF attribute is always advertised to IBGP neighbors and never advertised to EBGP peers, hence, it is exchanged only among

routers within the same AS. The BGP route with the highest LOCAL_PREF value is preferred⁵.

Consider Figure 2. All the packets that are sent from AS 100 to AS 200 prefer the route from router r2 to router r3 since it has a higher LOCAL_PREF value (120) than the route from router r1 to router r3 (110).

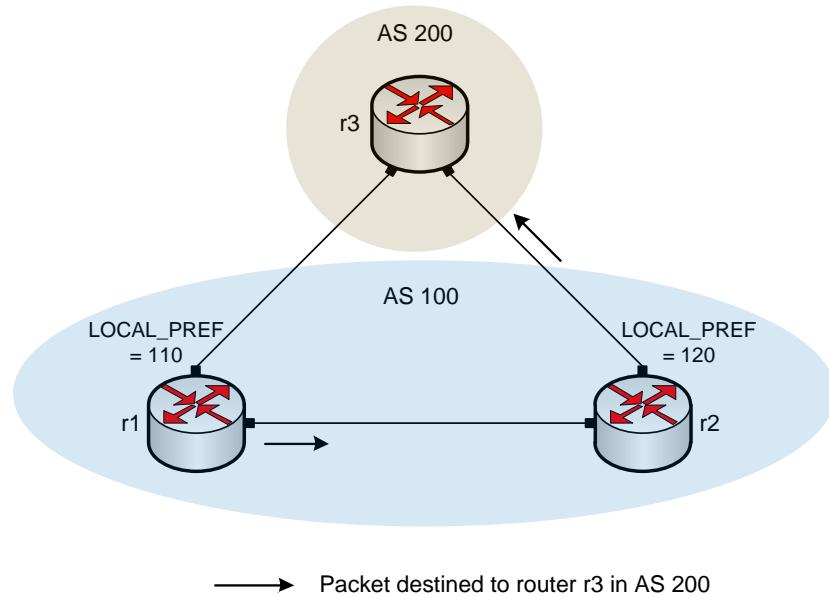


Figure 2. Packets sent from AS 100 to AS 200 prefer to exit AS 100 through router r2 instead of router r1 due to the configured LOCAL_PREF attribute.

1.3 MED attribute

The MED attribute indicates to external neighbors the preferred path into an AS if there are multiple entry points into the same AS. The BGP route with the lowest MED value is preferred⁶.

Consider Figure 3. All the packets that are sent from AS 200 to AS 100 prefer the route from router r3 to router r1 since it has a lower MED value (10) than the route from router r3 to router r2 (20).

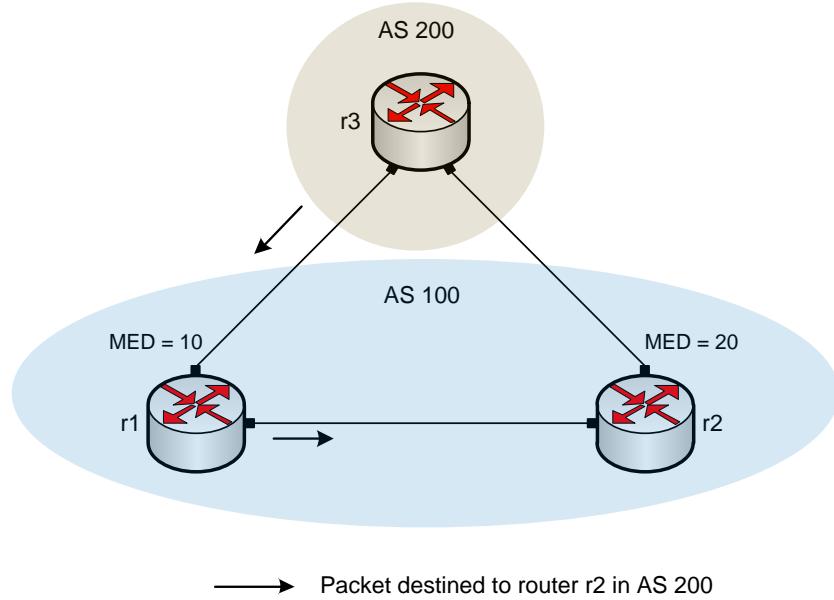


Figure 3. Packets sent from AS 200 to AS 100 prefer to enter AS 100 through router r1 instead of router r2 due to the configured MED attribute.

The MED is sent to EBGP peers; those peer routers propagate the MED attribute within their AS, but do not pass it on to the next AS. The MED value will be set back to 0 in case the same update is passed on to another AS⁷.

Consider Figure 4. The MED attribute of the route advertised by router r1 will propagate to all IBGP routers within AS 200, i.e., to routers r2 and r3. The MED attribute of this route (route advertised by router r1) will be set to zero if it is advertised beyond AS 200.

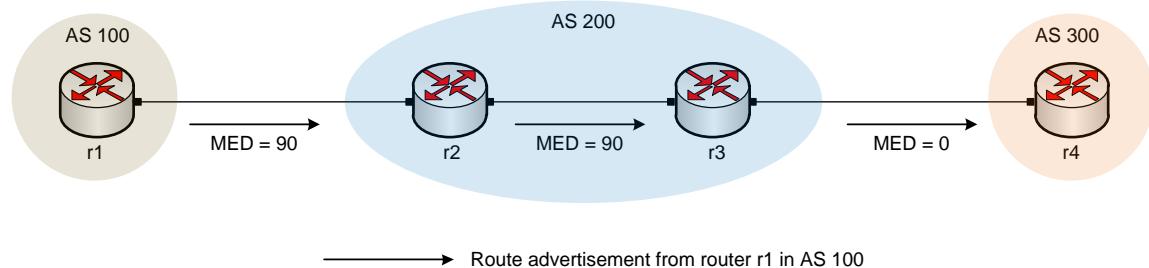


Figure 4. The value of the MED attribute is advertised to EBGP peers, whereas this value will be set to zero if it gets advertised outside the AS of those EBGP peers.

The MED attribute influences inbound traffic to an AS, i.e., the information coming-in to a network, whereas LOCAL_PREF attribute influences outbound traffic from an AS, i.e., the information going-out of a network⁶.

2 Lab topology

Consider Figure 5. The lab topology consists of two ASes, each identified by an Autonomous System Number (ASN). The Internet Service Provider (ISP), i.e., router r1,

provides Internet service to the Campus (routers r2 and r3). The ISP is connected to the Campus through two links to allow for redundancy. The secondary link should only be used if the primary link is not available. The ISP communicates with the Campus via EBGP routing protocol, and the routers within the Campus network communicate using IBGP and OSPF.

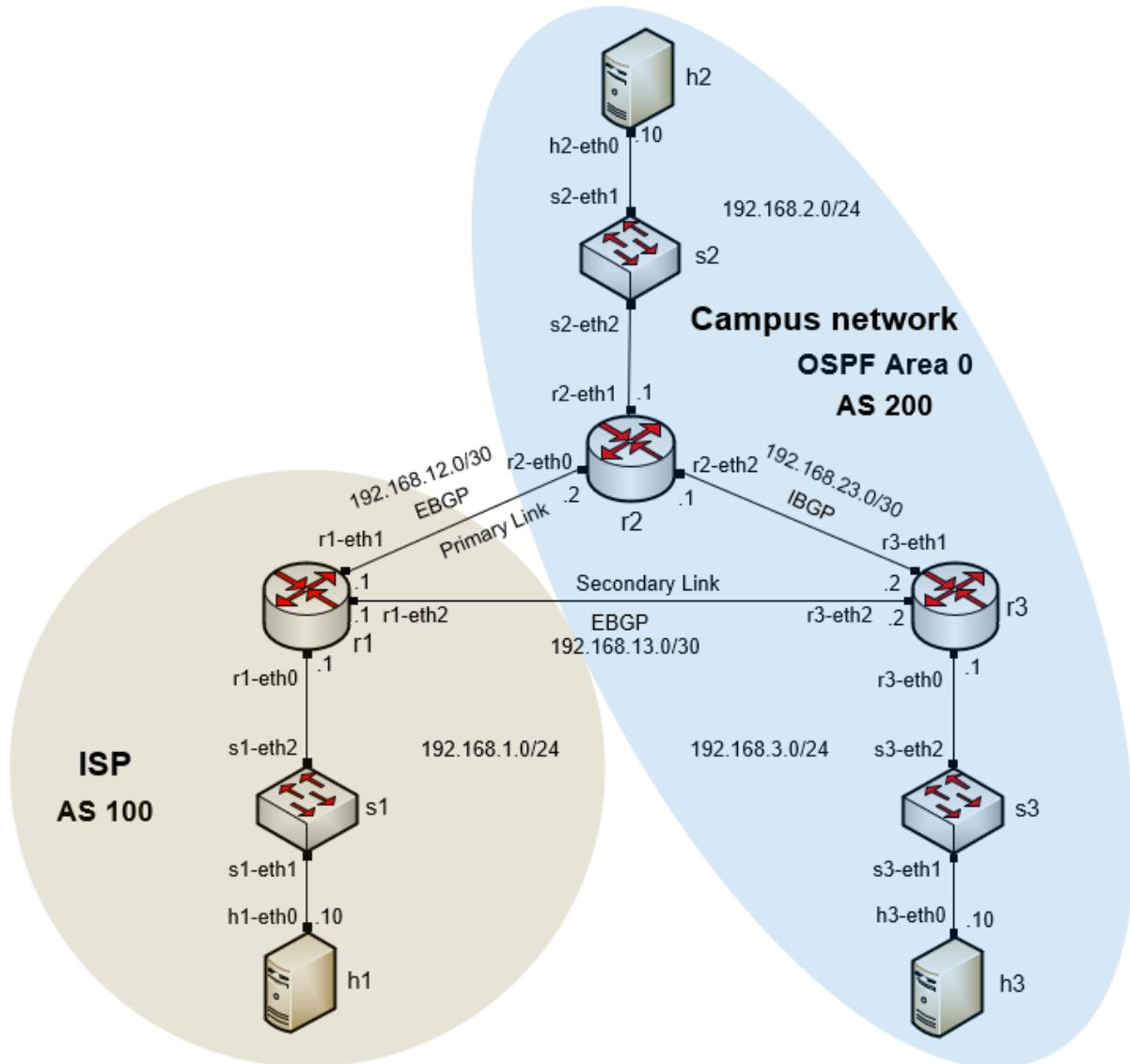


Figure 5. Lab topology.

2.1 Lab settings

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

Table 2. Topology information.

Device	Interface	IPV4 Address	Subnet	Default gateway
r1 (ISP)	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
r2 (Campus)	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
r3 (Campus)	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
h1	h1-eth0	192.168.1.10	/24	192.168.1.1
h2	h2-eth0	192.168.2.10	/24	192.168.2.1
h3	h3-eth0	192.168.3.10	/24	192.168.3.1

2.2 Open topology and load the configuration

Step 1. Start by launching Miniedit by clicking on Desktop's shortcut. When prompted for a password, type `password`.



Figure 6. MiniEdit shortcut.

Step 2. On Miniedit's menu bar, click on *File* then *open* to load the lab's topology. Locate the *Lab8.mn* topology file in the default directory, */home/frr/BGP_Labs/lab8* and click on *Open*.

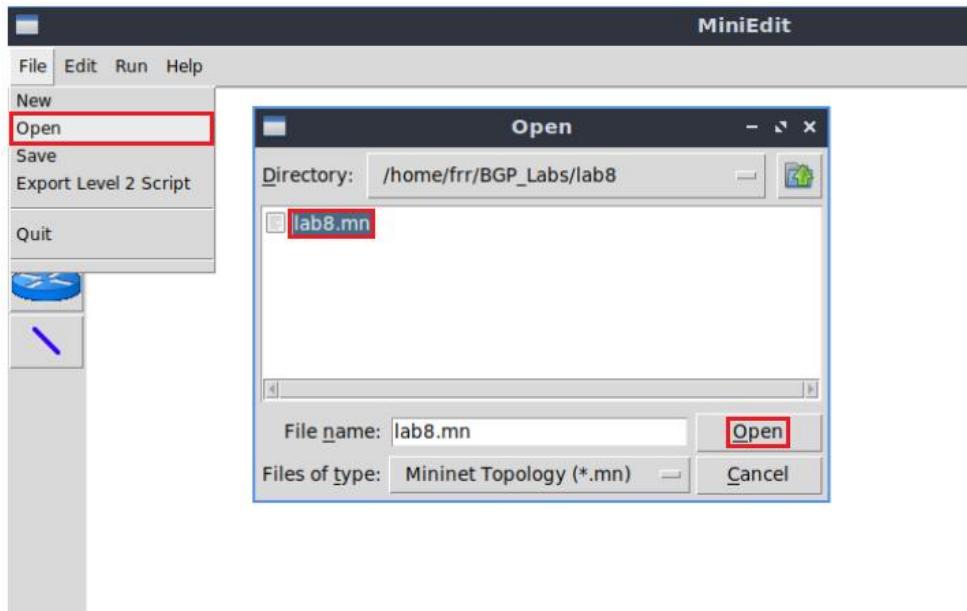


Figure 7. MiniEdit's Open dialog.

At this point the topology is loaded with all the required network components. You will execute a script that will load the configuration of the routers.

Step 3. Open the Linux terminal.

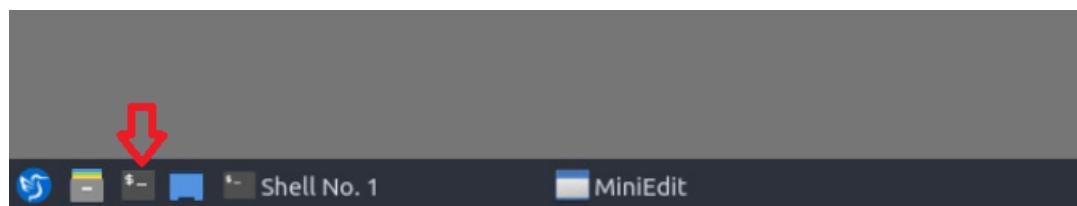


Figure 8. Opening Linux terminal.

Step 4. Click on the Linux's terminal and navigate into *BGP_Labs/lab8* 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.

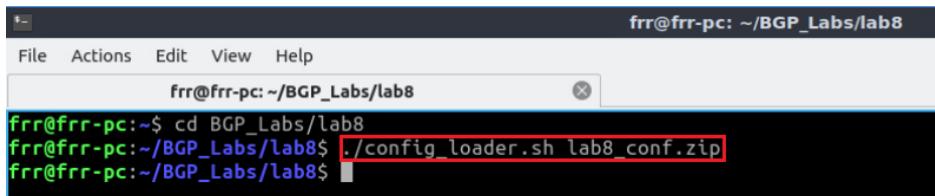
```
cd BGP_Labs/lab8
```

A screenshot of a Linux terminal window. The title bar says 'frr@frr-pc: ~/BGP_Labs/lab8'. The window has a menu bar with 'File', 'Actions', 'Edit', 'View', and 'Help'. Below the menu is a terminal prompt: 'frr@frr-pc: ~\$ cd BGP_Labs/lab8'. The command 'cd BGP_Labs/lab8' is highlighted with a red box. The prompt then changes to 'frr@frr-pc: ~/BGP_Labs/lab8\$'.

Figure 9. Entering the *BGP_Labs/lab8* directory.

Step 5. To execute the shell script, type the following command. The argument of the program corresponds to the configuration zip file that will be loaded in all the routers in the topology.

```
./config_loader.sh lab8_conf.zip
```

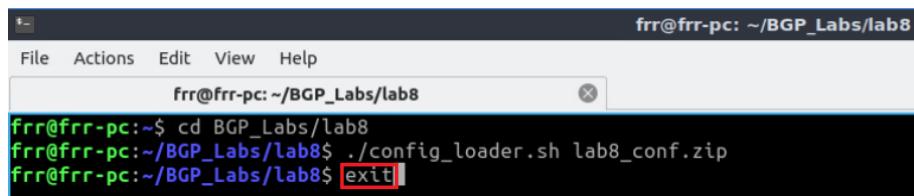


A terminal window titled "frr@frr-pc: ~/BGP_Labs/lab8". The command "cd BGP_Labs/lab8" is entered. The command "frr@frr-pc:~/BGP_Labs/lab8\$./config_loader.sh lab8_conf.zip" is highlighted with a red box. The command "frr@frr-pc:~/BGP_Labs/lab8\$" is also visible at the bottom.

Figure 10. Executing the shell script to load the configuration.

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

```
exit
```



A terminal window titled "frr@frr-pc: ~/BGP_Labs/lab8". The command "cd BGP_Labs/lab8" is entered. The command "frr@frr-pc:~/BGP_Labs/lab8\$./config_loader.sh lab8_conf.zip" is highlighted with a red box. The command "frr@frr-pc:~/BGP_Labs/lab8\$ exit" is highlighted with a red box.

Figure 11. Exiting from the terminal.

Step 7. At this point hosts h1, h2 and h3 interfaces are configured. To proceed with the emulation, click on the *Run* button located in lower left-hand side.



Figure 12. Starting the emulation.

Step 8. Click on Mininet's terminal, i.e., the one launched when MiniEdit was started.

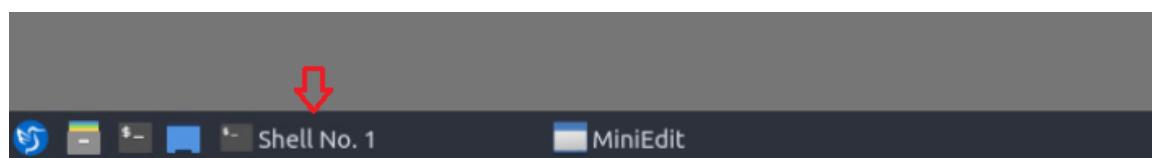
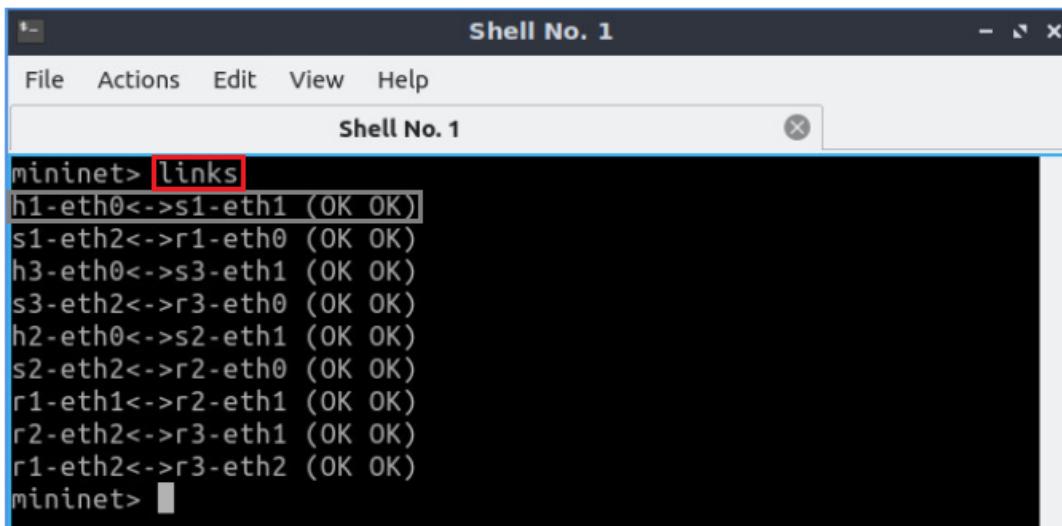


Figure 13. Opening Mininet's terminal.

Step 9. 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)
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-eth1<->r2-eth1 (OK OK)
r2-eth2<->r3-eth1 (OK OK)
r1-eth2<->r3-eth2 (OK OK)
mininet>
```

Figure 14. Displaying network interfaces.

In Figure 14, the link displayed within the gray box indicates that interface *eth0* of host h1 connects to interface *eth1* of switch s1 (i.e., $h1\text{-}eth0 \leftrightarrow s1\text{-}eth1$).

2.3 Load zebra daemon and Verify Connectivity

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

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.

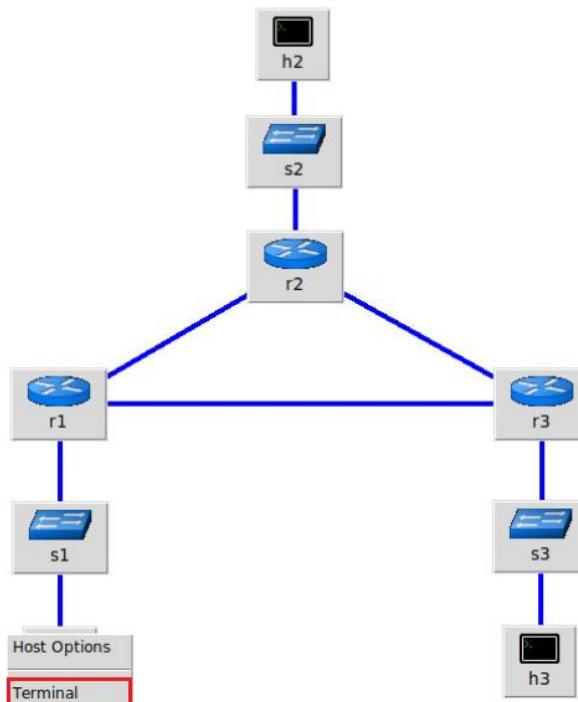
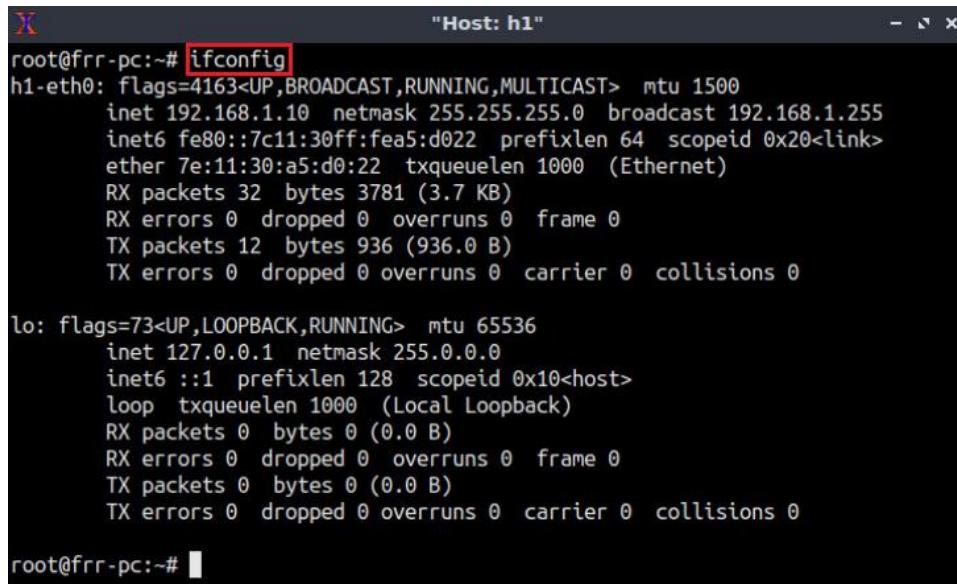


Figure 15. Opening a terminal on host h1.

Step 2. On 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
```



The terminal window shows the output of the `ifconfig` command. It displays two network interfaces: `h1-eth0` and `lo`. The `h1-eth0` interface has an IP address of 192.168.1.10 and a subnet mask of 255.255.255.0. The `lo` interface has an IP address of 127.0.0.1 and a subnet mask of 255.0.0.0. Both interfaces show no errors or dropped packets.

```
"Host: h1"
root@frrr-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@frrr-pc:~#
```

Figure 16. Output of `ifconfig` command.

Step 3. On 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 17. Output of `route` command.

Step 4. In order to verify hosts h2 and h3, proceed similarly by repeating from step 1 to step 3 in hosts h2 and h3 terminals. Similar results should be observed.

Step 5. You will validate that the router interfaces are configured correctly according to Table 2. In order to verify router r1, hold right-click on router r1 and select Terminal.

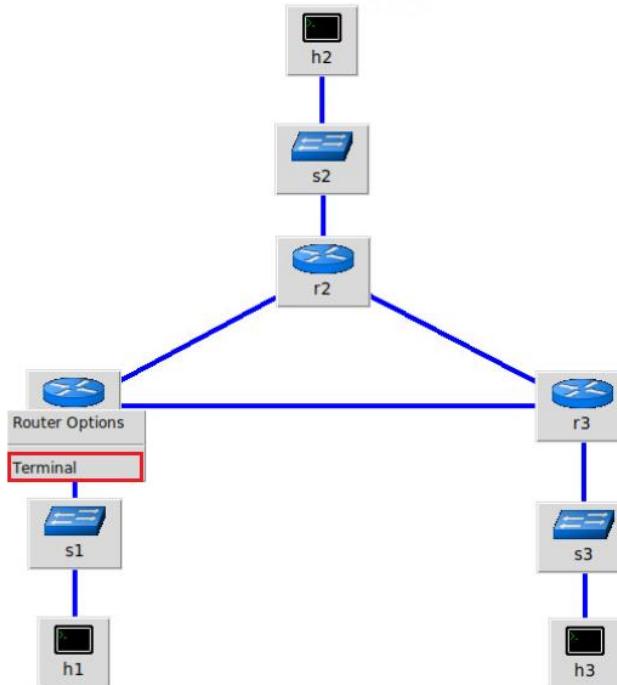
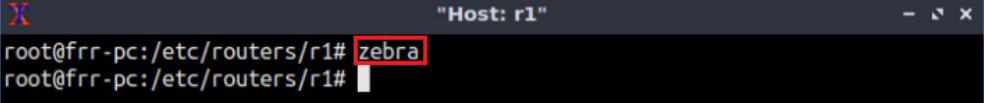


Figure 18. Opening terminal on router r1.

Step 6. In this step, you will start zebra daemon, which is a multi-server routing software that provides TCP/IP based routing protocols. The configuration will not be working if you

do not enable zebra daemon initially. In order to start the zebra, type the following command:

```
zebra
```

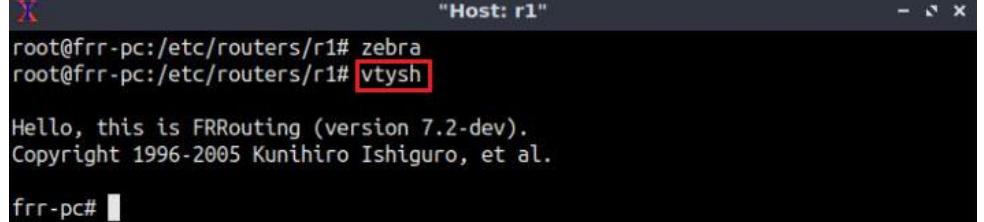


```
"Host: r1"
root@frr-pc:/etc/routers/r1# zebra
root@frr-pc:/etc/routers/r1#
```

Figure 19. Starting zebra daemon.

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

```
vtysh
```



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

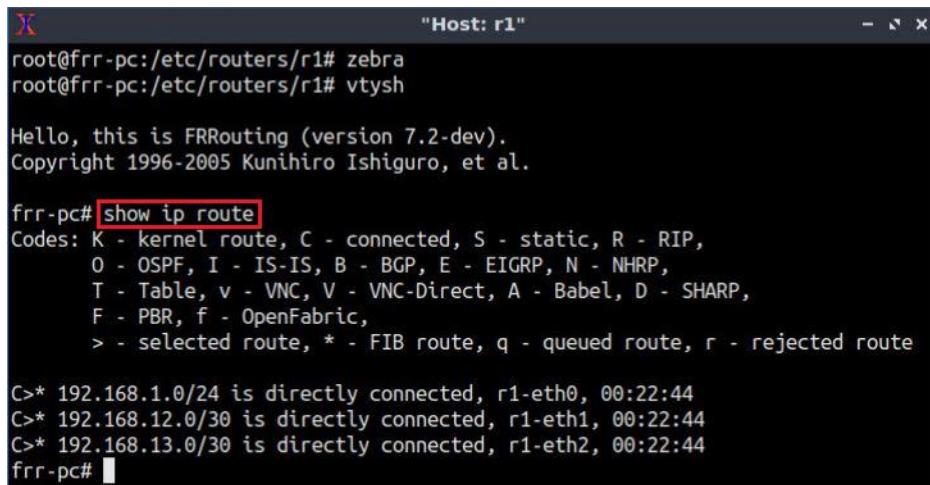
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc#
```

Figure 20. 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 networks attached to routers r2 (192.168.2.0/24) or router r3 (192.168.3.0/24) 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, 00:22:44
C>* 192.168.12.0/30 is directly connected, r1-eth1, 00:22:44
C>* 192.168.13.0/30 is directly connected, r1-eth2, 00:22:44
frr-pc#
```

Figure 21. Displaying routing table of router r1.

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.

```

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
C>* 192.168.23.0/30 is directly connected, r2-eth2, 00:00:08
frr-pc#

```

Figure 22. Displaying routing table of router r2.

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

```

root@frr-pc:/etc/routers/r3# zebra
root@frr-pc:/etc/routers/r3# 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>* 192.168.3.0/24 is directly connected, r3-eth0, 00:00:07
C>* 192.168.13.0/30 is directly connected, r3-eth2, 00:00:07
C>* 192.168.23.0/30 is directly connected, r3-eth1, 00:00:07
frr-pc#

```

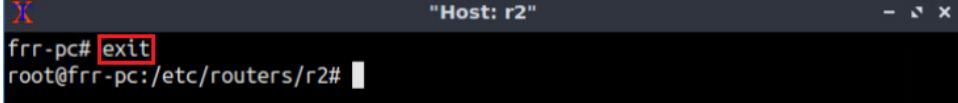
Figure 23. Displaying routing table of router r3.

3 Configure OSPF on router r2 and router r3

In this section, you will configure OSPF routing protocol on router r2 and router r3. First, you will enable the OSPF daemon on routers r3 and r4. Second, you will establish a single area OSPF, which is classified as area 0 or backbone area. Finally, the networks 192.168.2.0/24, 192.168.3.0/24, and 192.168.23.0/30 will be advertised in area 0 between router r2 and router r3.

Step 1. To configure OSPF routing protocol, you need to enable the OSPF daemon first. In router r2, type the following command to exit the vtysh session.

```
exit
```

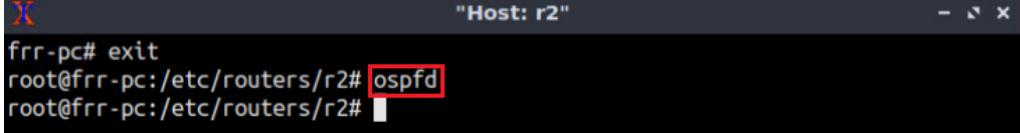


The terminal window shows the command 'exit' being typed at the root prompt 'frr-pc#'. The window title is "Host: r2".

Figure 24. Exiting the vtysh session.

Step 2. Type the following command on router r2 terminal to enable OSPF daemon.

```
ospfd
```



The terminal window shows the command 'ospfd' being typed at the root prompt 'frr-pc#'. The window title is "Host: r2".

Figure 25. Starting OSPF daemon.

Step 3. In order to enter to router r2 terminal, issue the following command:

```
vtysh
```



The terminal window shows the command 'vtysh' being typed at the root prompt 'frr-pc#'. The window title is "Host: r2". The output includes FRRouting version information and a prompt 'frr-pc#'. The 'vtysh' command is highlighted with a red box.

Figure 26. Starting vtysh on router r2.

Step 4. To enable router r2 configuration mode, issue the following command:

```
configure terminal
```



The terminal window shows the command 'configure terminal' being typed at the configuration prompt 'frr-pc(config)#'. The window title is "Host: r2". The output includes FRRouting version information and a prompt 'frr-pc(config)#'. The 'configure terminal' command is highlighted with a red box.

Figure 27. Enabling configuration mode on router r2.

Step 5. In order to configure OSPF routing protocol, type the command shown below. This command will enable OSPF configuration mode where you can advertise the networks directly connected to the router r2.

```
router ospf
```

The terminal window shows the following session:

```
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)#
```

Figure 28. Configuring OSPF on router r3.

Step 6. In this step, type the following command to enable the interface *r2-eth2*, corresponding to the network 192.168.23.0/30, to participate in the routing process. This network is associated with area 0.

```
network 192.168.23.0/30 area 0
```

The terminal window shows the following session:

```
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)# network 192.168.23.0/30 area 0
frr-pc(config-router)#
```

Figure 29. Enabling the interface corresponding to the network 192.168.23.0/30 to participate in the OSPF routing process.

Step 7. Similarly, type the following command in router r2 terminal to enable the interface *r2-eth0* to participate in the OSPF routing process.

```
network 192.168.2.0/24 area 0
```

```
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)# network 192.168.23.0/30 area 0
frr-pc(config-router)# network 192.168.2.0/24 area 0
frr-pc(config-router)#[
```

Figure 30. Enabling the interface corresponding to 192.168.2.0/24 to participate in the OSPF routing process.

Step 8. Type the following command to exit from the configuration mode.

```
end
```

```
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)# 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 31. Exiting from configuration mode.

Step 9. 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, on router r3 terminal issue the commands depicted below.

```
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)# network 192.168.23.0/30 area 0
frr-pc(config-router)# network 192.168.3.0/24 area 0
frr-pc(config-router)# end
frr-pc#[
```

Figure 32. Configuring OSPF on router r3.

Step 10. Type the following command to verify the routing table of router r3.

```
show ip route
```

The terminal window shows the output of the 'show ip route' command. The output includes a legend of route codes and a list of routes. Router r3 has direct connections to networks 192.168.3.0/24, 192.168.13.0/30, and 192.168.23.0/30 via its respective interfaces (r3-eth0, r3-eth2, r3-eth1). It also has an OSPF route to network 192.168.2.0/24 via interface r3-eth1 with an administrative distance of 110. The output is as follows:

```
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 192.168.23.1, r3-eth1, 00:00:48
0  192.168.3.0/24 [110/10] is directly connected, r3-eth0, 00:00:33
C>* 192.168.3.0/24 is directly connected, r3-eth0, 00:03:37
C>* 192.168.13.0/30 is directly connected, r3-eth2, 00:03:37
0  192.168.23.0/30 [110/10] is directly connected, r3-eth1, 00:00:58
C>* 192.168.23.0/30 is directly connected, r3-eth1, 00:03:37
frr-pc#
```

Figure 33. Verifying the routing table of router r3.

Router r3 reaches the network 192.168.2.0/24 via the IP address 192.168.23.1. Networks 192.168.3.0/24 and 192.168.23.0/30 have two available paths from router r3. The administrative distance (AD) of the paths advertised through OSPF is 110. The AD is a value used by routers to select the best path when there are multiple available routes to the same destination. A smaller AD is always preferable to the routers. The characters **>*** indicates that the following path is used to reach a specific network. Router r3 prefers directly connected networks over OSPF since the former has a lower AD than the latter.

4 Configure BGP on all routers

In this section, you will configure BGP on all routers. Routers r2 and r3 communicate with router r1 through EBGP, while router r2 communicates with router r3 through IBGP. You will assign BGP neighbors to allow the routers to exchange BGP routes. Furthermore, routers r1, r2, and r3 will advertise their LANs via BGP so that the LANs are learned by peer routers.

4.1 Configure EBGP on router r1

Step 1. To configure BGP routing protocol, you need to enable the BGP daemon first. In router r1 terminal, type the following command to exit the vtysh session:

```
exit
```

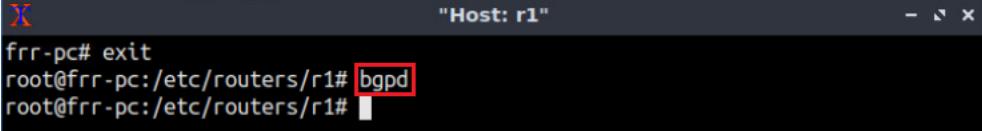
The terminal window shows the exit command being typed in the vtysh session of router r1. The session is running on host r1. The command 'exit' is highlighted with a red box. The output is as follows:

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

Figure 34. Exiting the vtysh session.

Step 2. Type the following command on r1 terminal to enable and start BGP routing protocol.

```
bgpd
```

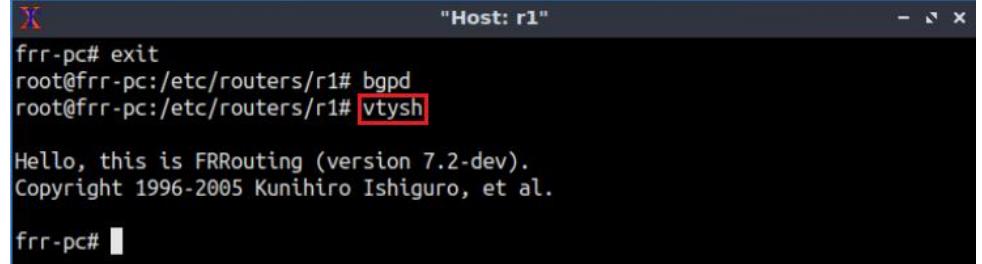


```
frr-pc# exit
root@frr-pc:/etc/routers/r1# bgpd
root@frr-pc:/etc/routers/r1#
```

Figure 35. Starting BGP daemon.

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

```
vtysh
```



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

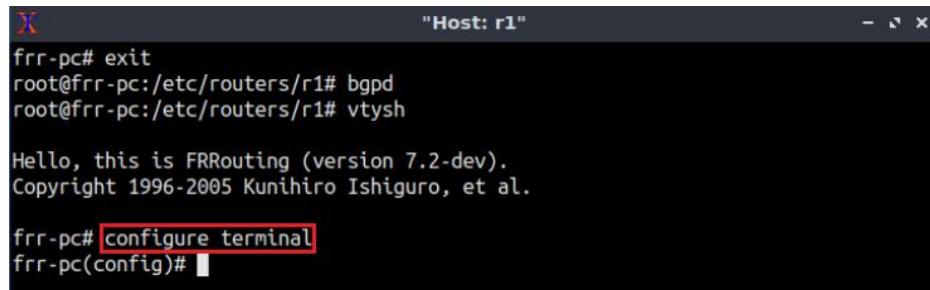
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc#
```

Figure 36. Starting vtysh on router r1.

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

```
configure terminal
```



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

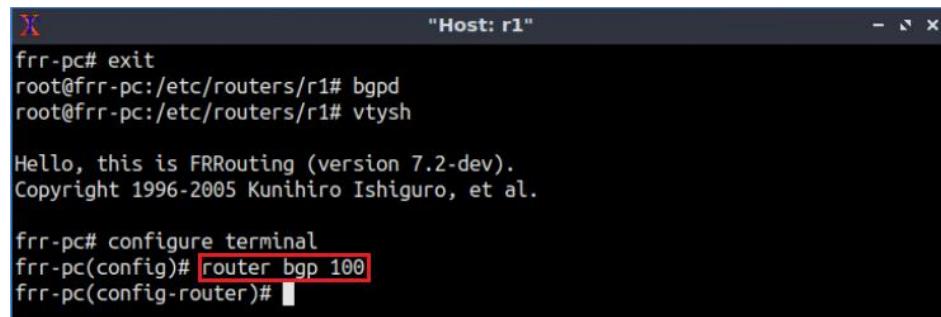
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

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

Figure 37. Enabling configuration mode on router r1.

Step 5. The ASN assigned for router r1 is 100. In order to configure BGP, type the following command:

```
router bgp 100
```



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

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

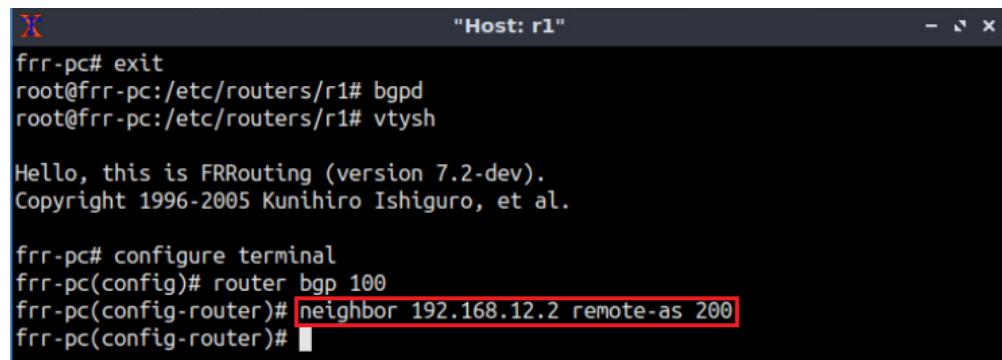
frr-pc# configure terminal
frr-pc(config)# router bgp 100
frr-pc(config-router)#

```

Figure 38. Configuring BGP on router r1.

Step 6. To configure a BGP neighbor to router r1 (AS 100), type the command shown below. This command specifies the neighbor IP address (192.168.12.2) and the ASN of the remote BGP peer (AS 200).

```
neighbor 192.168.12.2 remote-as 200
```



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

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

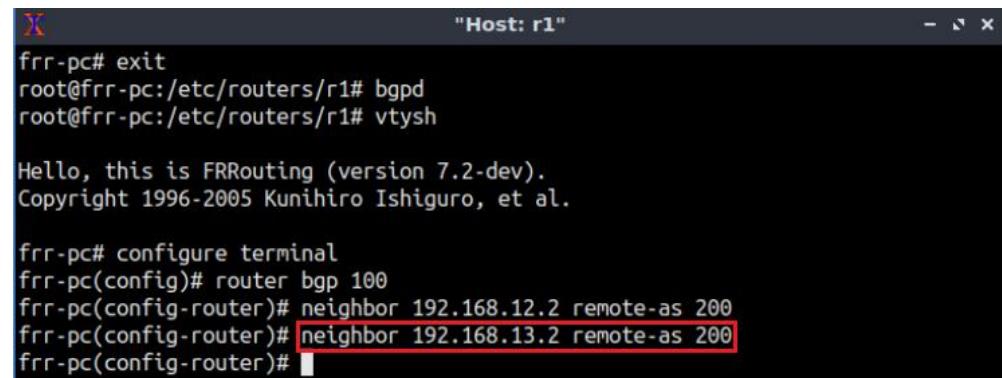
frr-pc# configure terminal
frr-pc(config)# router bgp 100
frr-pc(config-router)# neighbor 192.168.12.2 remote-as 200
frr-pc(config-router)#

```

Figure 39. Assigning BGP neighbor to router r1.

Step 7. Type the following command to add another BGP neighbor (192.168.13.2) to router r1.

```
neighbor 192.168.13.2 remote-as 200
```



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

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

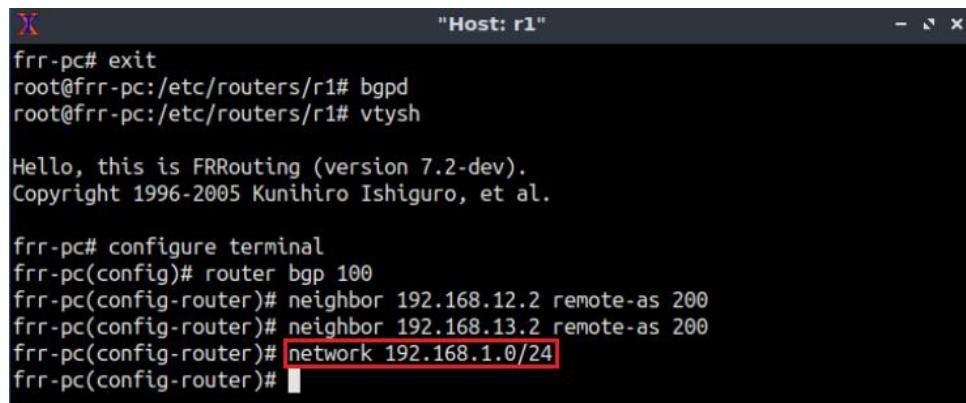
frr-pc# configure terminal
frr-pc(config)# router bgp 100
frr-pc(config-router)# neighbor 192.168.12.2 remote-as 200
frr-pc(config-router)# neighbor 192.168.13.2 remote-as 200
frr-pc(config-router)#

```

Figure 40. Assigning BGP neighbor to router r1.

Step 8. In this step, router r1 will advertise the LAN 192.168.1.0/24 to its BGP peers. To do so, issue the following command:

```
network 192.168.1.0/24
```



```
"Host: r1"
frr-pc# exit
root@frr-pc:/etc/routers/r1# bgpd
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 bgp 100
frr-pc(config-router)# neighbor 192.168.12.2 remote-as 200
frr-pc(config-router)# neighbor 192.168.13.2 remote-as 200
frr-pc(config-router)# network 192.168.1.0/24
frr-pc(config-router)#

```

Figure 41. Advertising local network on router r1.

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

```
end
```



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

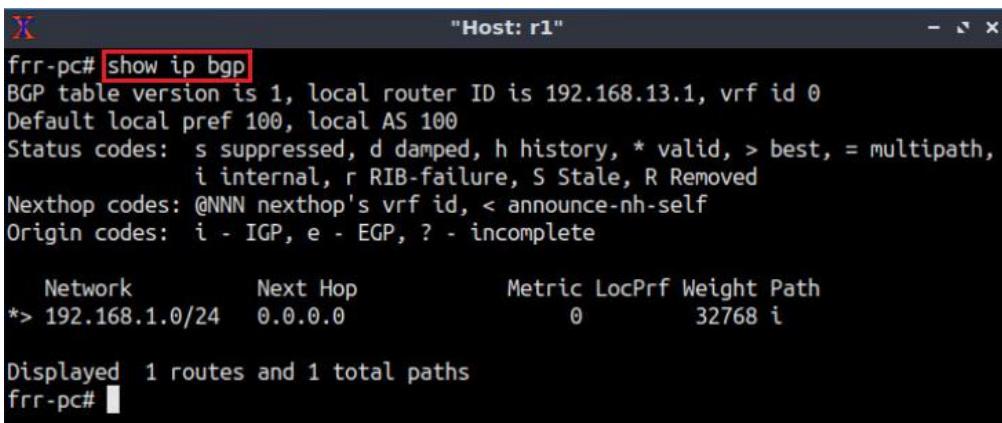
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# router bgp 100
frr-pc(config-router)# neighbor 192.168.12.2 remote-as 200
frr-pc(config-router)# neighbor 192.168.13.2 remote-as 200
frr-pc(config-router)# network 192.168.1.0/24
frr-pc(config-router)# end
frr-pc# 
```

Figure 42. Exiting from configuration mode.

Step 10. Type the following command to verify BGP networks. You will observe the LAN network of router r1.

```
show ip bgp
```



```
"Host: r1"
frr-pc# show ip bgp
BGP table version is 1, local router ID is 192.168.13.1, vrf id 0
Default local pref 100, local AS 100
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
              i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*-> 192.168.1.0/24    0.0.0.0                  0        32768 i

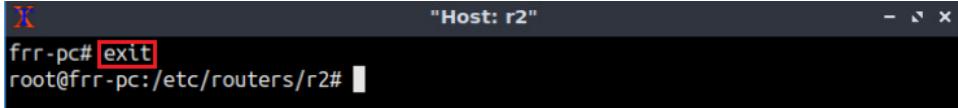
Displayed 1 routes and 1 total paths
frr-pc# 
```

Figure 43. Verifying BGP networks on router r1.

4.2 Configure EBGP and IBGP on router r2 and router r3

Step 1. To configure BGP routing protocol, you need to enable the BGP daemon first. In router r2 terminal, type the following command to exit the vtysh session:

```
exit
```

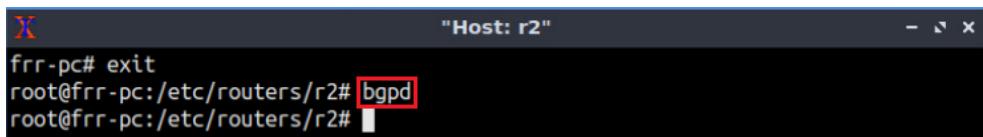


The terminal window has a title bar "Host: r2". The command "exit" is highlighted with a red box. The prompt "root@frr-pc:/etc/routers/r2#" is visible below the command line.

Figure 44. Exiting the vtysh session on router r2 terminal.

Step 2. Type the following command in router r2 terminal to enable and to start BGP routing protocol.

```
bgpd
```

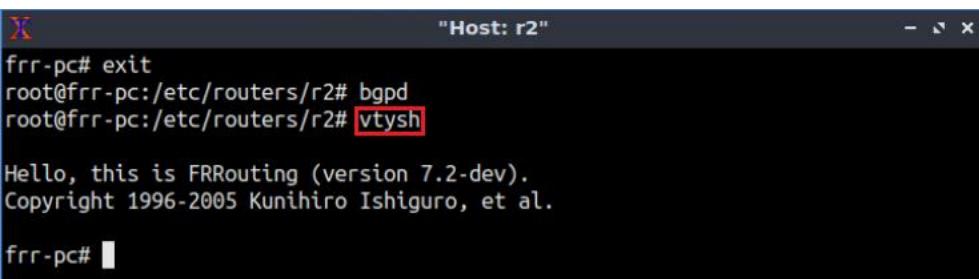


The terminal window has a title bar "Host: r2". The command "bgpd" is highlighted with a red box. The prompt "root@frr-pc:/etc/routers/r2#" is visible below the command line.

Figure 45. Starting BGP daemon.

Step 3. In order to enter to router r2 terminal, type the following command:

```
vtysh
```



The terminal window has a title bar "Host: r2". The command "vtysh" is highlighted with a red box. The prompt "root@frr-pc:/etc/routers/r2#" is visible below the command line. Below the prompt, there is a message: "Hello, this is FRRouting (version 7.2-dev). Copyright 1996-2005 Kunihiro Ishiguro, et al."

Figure 46. Starting vtysh on router r2.

Step 4. To enable router r2 into configuration mode, issue the following command:

```
configure terminal
```



```
"Host: r2"
frr-pc# exit
root@frr-pc:/etc/routers/r2# bgpd
root@frr-pc:/etc/routers/r2# vtysh

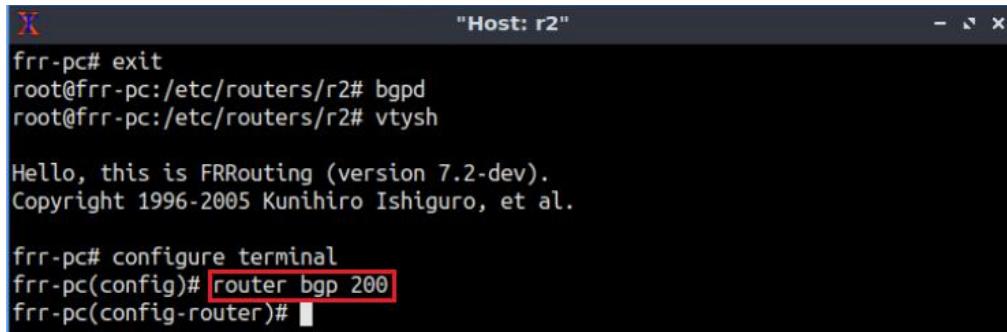
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

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

Figure 47. Enabling configuration mode on router r2.

Step 5. The ASN assigned for router r2 is 200. In order to configure BGP, type the following command:

```
router bgp 200
```



```
"Host: r2"
frr-pc# exit
root@frr-pc:/etc/routers/r2# bgpd
root@frr-pc:/etc/routers/r2# vtysh

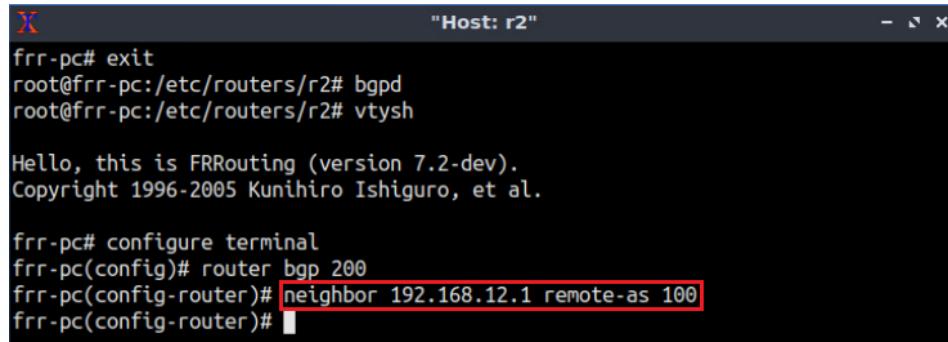
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# router bgp 200
frr-pc(config-router)#
```

Figure 48. Configuring BGP on router r2.

Step 6. To configure a BGP neighbor to router r2 (AS 200), type the command shown below. This command specifies the neighbor IP address (192.168.12.1) and the ASN of the remote BGP peer (AS 100).

```
neighbor 192.168.12.1 remote-as 100
```



```
"Host: r2"
frr-pc# exit
root@frr-pc:/etc/routers/r2# bgpd
root@frr-pc:/etc/routers/r2# vtysh

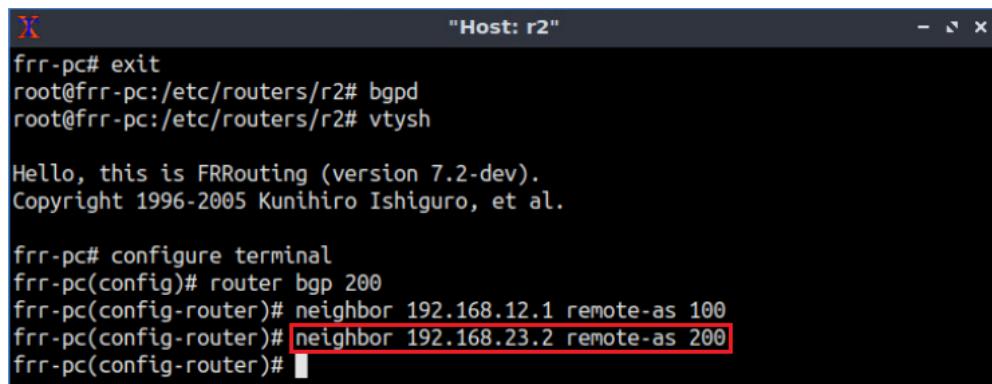
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.12.1 remote-as 100
frr-pc(config-router)#
```

Figure 49. Assigning BGP neighbor to router r2.

Step 7. Type the following command to add another BGP neighbor (192.168.23.2) to router r2. When BGP neighbor routers are within the same AS, they are referred to as IBGP neighbors.

```
neighbor 192.168.23.2 remote-as 200
```



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

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

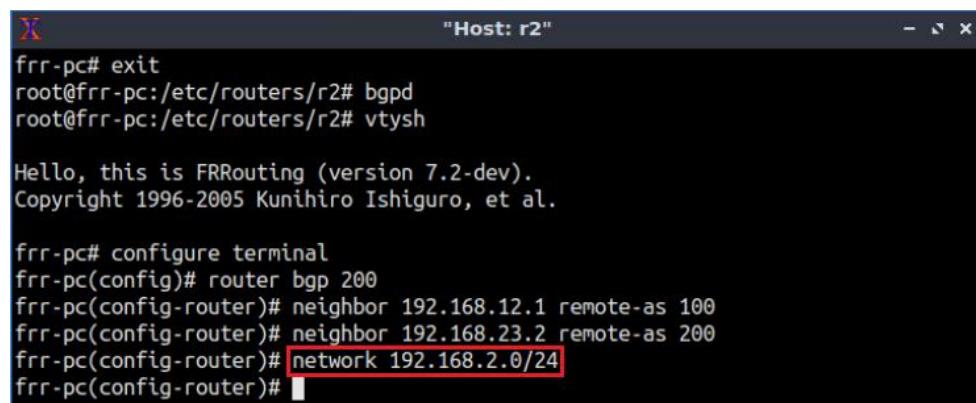
frr-pc# configure terminal
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.12.1 remote-as 100
frr-pc(config-router)# neighbor 192.168.23.2 remote-as 200
frr-pc(config-router)#

```

Figure 50. Assigning BGP neighbor to router r2.

Step 8. In this step, router r2 will advertise the LAN, 192.168.2.0/24 and the network will be learned by router r1 through EBGP. In order to advertise the LAN 192.168.2.0/24 connected to router r2, issue the following command:

```
network 192.168.2.0/24
```



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

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

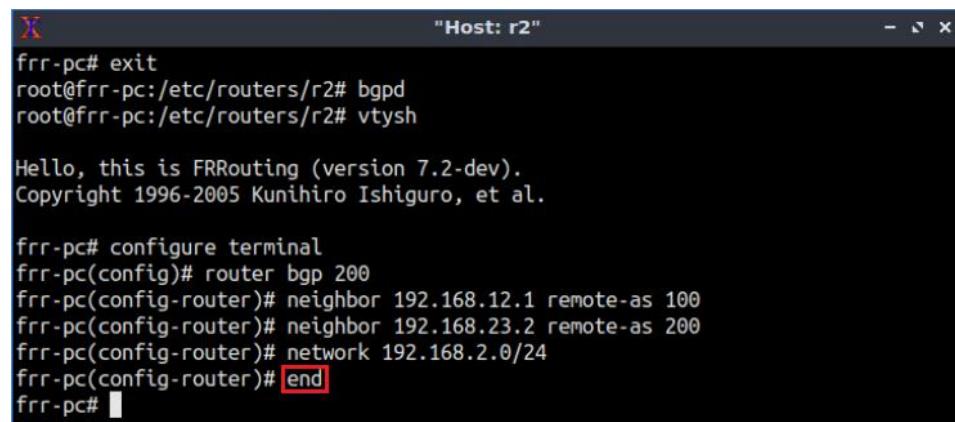
frr-pc# configure terminal
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.12.1 remote-as 100
frr-pc(config-router)# neighbor 192.168.23.2 remote-as 200
frr-pc(config-router)# network 192.168.2.0/24
frr-pc(config-router)#

```

Figure 51. Advertising local network on router r2.

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

```
end
```



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

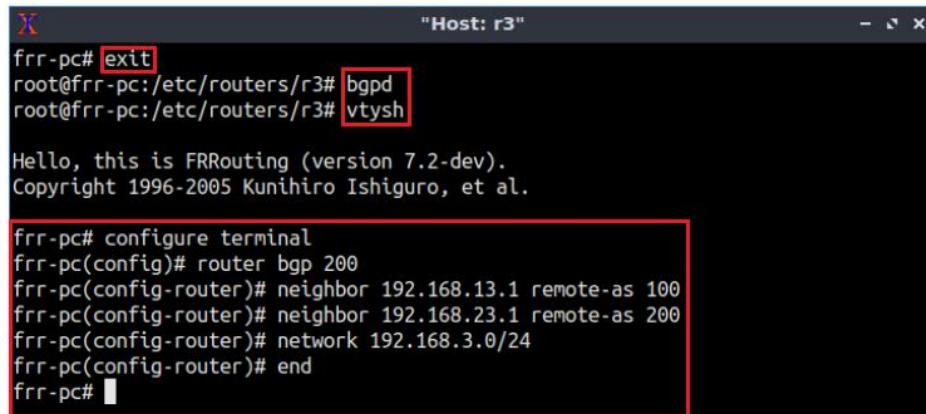
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.12.1 remote-as 100
frr-pc(config-router)# neighbor 192.168.23.2 remote-as 200
frr-pc(config-router)# network 192.168.2.0/24
frr-pc(config-router)# end
frr-pc#

```

Figure 52. Exiting from configuration mode.

Step 10. Follow from step 1 to step 9 but with different metrics in order to configure BGP on router r3. All the steps are summarized in the following figure.



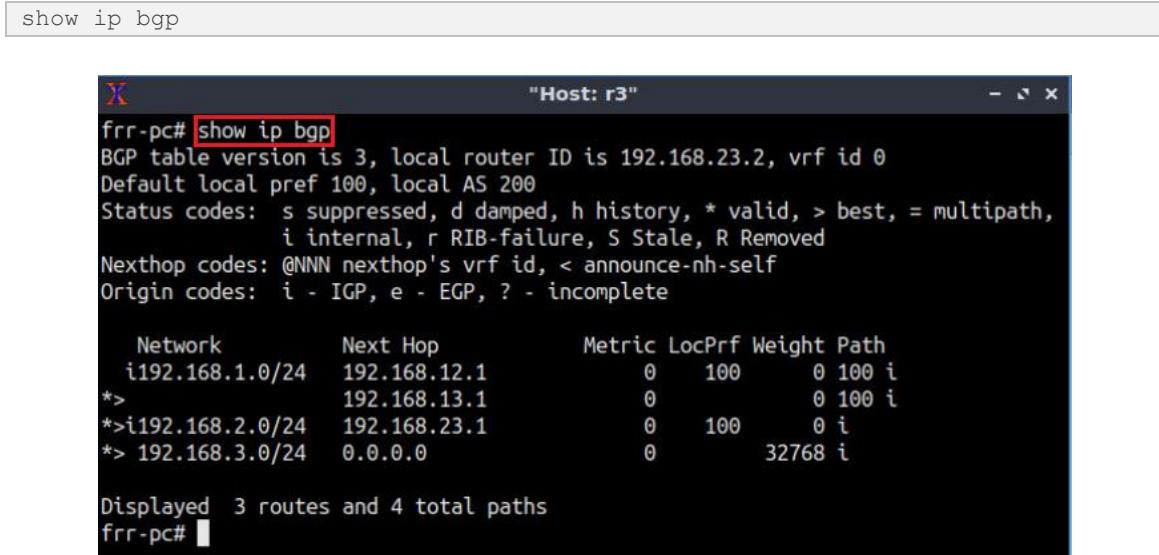
```
frr-pc# exit
root@frr-pc:/etc/routers/r3# bgpd
root@frr-pc:/etc/routers/r3# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.13.1 remote-as 100
frr-pc(config-router)# neighbor 192.168.23.1 remote-as 200
frr-pc(config-router)# network 192.168.3.0/24
frr-pc(config-router)# end
frr-pc#
```

Figure 53. Configuring BGP on router r3.

Step 11. Type the following command to verify the BGP table of router r3.



```
show ip bgp
```

```
frr-pc# show ip bgp
BGP table version is 3, local router ID is 192.168.23.2, vrf id 0
Default local pref 100, local AS 200
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexhop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
i192.168.1.0/24    192.168.12.1        0     100      0 100 i
*>                 192.168.13.1        0           0 100 i
*>i192.168.2.0/24  192.168.23.1        0     100      0 i
*> 192.168.3.0/24  0.0.0.0           0           0 32768 i

Displayed 3 routes and 4 total paths
frr-pc#
```

Figure 54. Verifying BGP networks on router r3.

Consider Figure 54. BGP table of router r3 contains three LANs advertised through BGP. There are two available paths to reach the network 192.168.1.0/24 (through 192.168.12.1 and 192.168.13.1). Router r3 communicates with the network 192.168.1.0/24 via 192.168.13.1.

4.3 Advertise networks on router r2 and router r3 through OSPF

At this point, the routing table of router r2 does not have a route to the network 192.168.13.0/30, i.e., the link between router r1 and router r3 as the network (192.168.13.0/30) exists in another AS. Similarly, the routing table of router r3 does not have a route to the network 192.168.12.0/30, i.e., the link between router r1 and router r2. In this section, you will configure OSPF to advertise the networks 192.168.12.0/30 and

192.168.13.0/30 within area 0. Consequently, routers r2 and r3 will be aware of both networks.

Step 1. Type the following command to verify the routing table of router r3.

```
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

B>* 192.168.1.0/24 [20/0] via 192.168.13.1, r3-eth2, 00:18:30
B  192.168.2.0/24 [200/0] via 192.168.23.1, r3-eth1, 00:18:10
O>* 192.168.2.0/24 [110/20] via 192.168.23.1, r3-eth1, 00:28:32
O  192.168.3.0/24 [110/10] is directly connected, r3-eth0, 00:28:42
C>* 192.168.3.0/24 is directly connected, r3-eth0, 00:31:04
C>* 192.168.13.0/30 is directly connected, r3-eth2, 00:31:04
O  192.168.23.0/30 [110/10] is directly connected, r3-eth1, 00:28:42
C>* 192.168.23.0/30 is directly connected, r3-eth1, 00:31:04
frr-pc#
```

Figure 55. Verifying BGP networks on router r3.

Consider Figure 55. The routing table of router r3 does not contain a route to 192.168.12.0/30, since this network was not advertised before.

Step 2. In router r2 terminal, type the following command to enable the configuration mode:

```
configure terminal
```

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

Figure 56. Enabling configuration mode on router r2.

Step 3. Type the following command to configure OSPF routing protocol.

```
router ospf
```

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

Figure 57. Configuring OSPF on router r2.

Step 4. The interface *r2-eth1* does not need to receive OSPF advertisements since BGP is running between routers r1 and r2. You will configure OSPF to set the interface *r2-eth1* as passive, i.e., the interface does not participate in OSPF and does not establish

adjacencies or send routing updates. However, the interface is announced as part of the routing network.

```
passive-interface r2-eth1
```

```
"Host: r2"
frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# passive-interface r2-eth1
frr-pc(config-router)#

```

Figure 58. Enabling passive-interface r2-eth1.

Step 5. In this step, type the following command to enable the interface *r2-eth1*, corresponding to the network 192.168.12.0/30, to advertise the network in the OSPF routing process within area 0.

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

```
"Host: r2"
frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# passive-interface r2-eth1
frr-pc(config-router)# network 192.168.12.0/30 area 0
frr-pc(config-router)#

```

Figure 59. Advertising the interface 192.168.12.0/30.

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

```
end
```

```
"Host: r2"
frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# passive-interface r2-eth1
frr-pc(config-router)# network 192.168.12.0/30 area 0
frr-pc(config-router)# end
frr-pc#
```

Figure 60. Exiting from configuration mode.

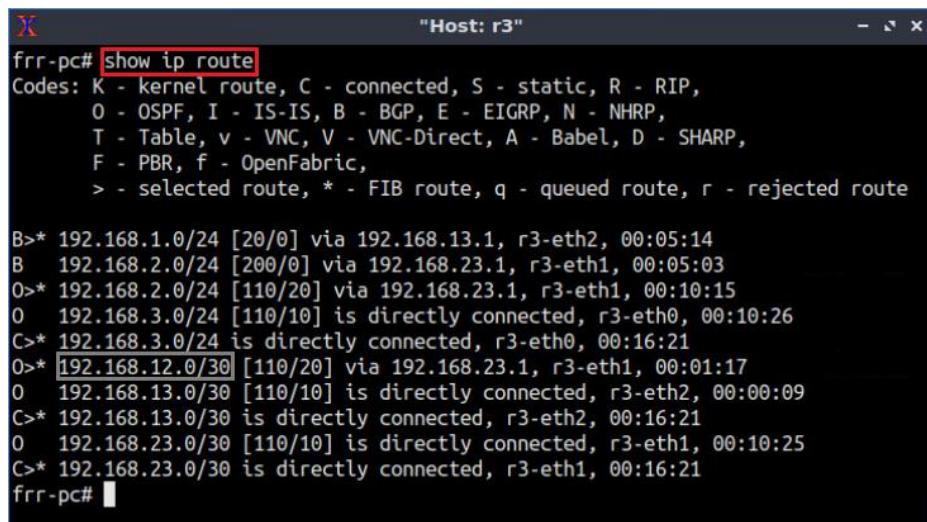
Step 7. In router r3 terminal, follow from step 2 to step 6 but with different metrics to set the interface *r3-eth2* as passive, as well as to advertise the network 192.168.13.0/30 in area 0.

```
"Host: r3"
frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# passive-interface r3-eth2
frr-pc(config-router)# network 192.168.13.0/30 area 0
frr-pc(config-router)# end
frr-pc#
```

Figure 61. Configuring OSPF on router r3.

Step 8. Type the following command in router r3 terminal to verify its routing table. Network 192.168.12.0/30 should be listed in the table as it was advertised by router r2 through OSPF.

```
show ip route
```



The terminal window shows the output of the 'show ip route' command on router r3. The output lists various network routes with their details. The route '192.168.12.0/30' is highlighted in red, indicating it was advertised from router r2 through OSPF.

```
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

B>* 192.168.1.0/24 [20/0] via 192.168.13.1, r3-eth2, 00:05:14
B    192.168.2.0/24 [200/0] via 192.168.23.1, r3-eth1, 00:05:03
O>* 192.168.2.0/24 [110/20] via 192.168.23.1, r3-eth1, 00:10:15
O    192.168.3.0/24 [110/10] is directly connected, r3-eth0, 00:10:26
C>* 192.168.3.0/24 is directly connected, r3-eth0, 00:16:21
O>* 192.168.12.0/30 [110/20] via 192.168.23.1, r3-eth1, 00:01:17
O    192.168.13.0/30 [110/10] is directly connected, r3-eth2, 00:00:09
C>* 192.168.13.0/30 is directly connected, r3-eth2, 00:16:21
O    192.168.23.0/30 [110/10] is directly connected, r3-eth1, 00:10:25
C>* 192.168.23.0/30 is directly connected, r3-eth1, 00:16:21
frr-pc#
```

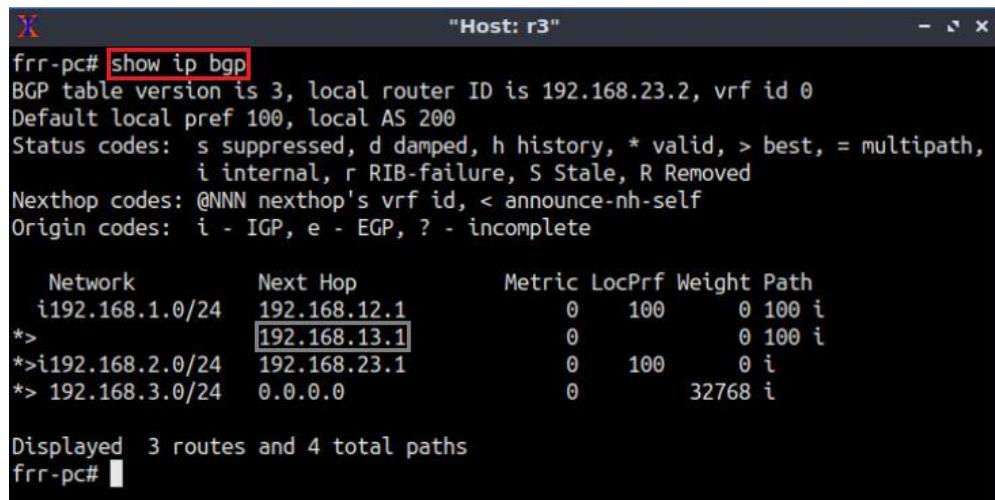
Figure 62. Verifying the routing table of router r3.

5 Set and verify BGP LOCAL_PREF on router r2 and router r3

In this section, you will adjust the LOCAL_PREF attribute on a router using route maps. A route map consists of a series of statements that allow or deny after checking if a route matches the policy. You will create a route map that sets LOCAL_PREF attribute. Eventually, you will inject this route map into BGP, so that primary and secondary links are created.

Step 1. Type the following command to verify the BGP table of router r3.

```
show ip bgp
```



```
"Host: r3"
frr-pc# show ip bgp
BGP table version is 3, local router ID is 192.168.23.2, vrf id 0
Default local pref 100, local AS 200
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

Network          Next Hop            Metric LocPrf Weight Path
i192.168.1.0/24 192.168.12.1      0    100      0 100 i
*->              192.168.13.1      0          0 100 i
*>i192.168.2.0/24 192.168.23.1      0    100      0 i
*> 192.168.3.0/24 0.0.0.0          0          32768 i

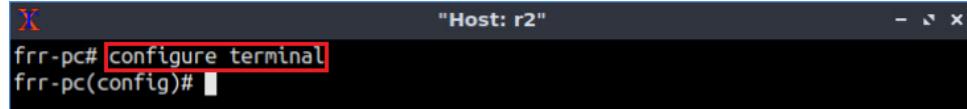
Displayed 3 routes and 4 total paths
frr-pc#"
```

Figure 63. Verifying BGP networks on router r3.

Consider Figure 63. Router r3 communicates with router r1 through the IP address 192.168.13.1 which is expected to act as secondary link in this lab. In the next step, you will configure local-preference attribute to make the link as secondary path.

Step 2. In router r2 terminal, type the following command to enable the configuration mode:

```
configure terminal
```

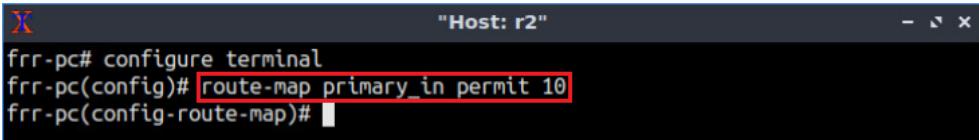


```
"Host: r2"
frr-pc# configure terminal
frr-pc(config)#"
```

Figure 64. Enabling configuration mode on router r2.

Step 3. Type the following command to create a route map named *primary_in* with permit clause. The *permit* clause is used to allow the set operation, i.e., to allow setting the LOCAL_PREF value. The route map must be assigned a sequence number. You will use default sequence number which is 10.

```
route-map primary_in permit 10
```



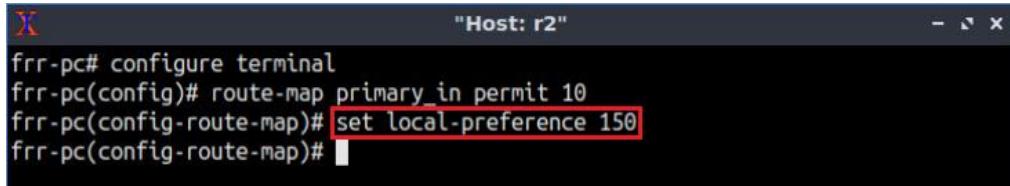
```
"Host: r2"
frr-pc# configure terminal
frr-pc(config)# route-map primary_in permit 10
frr-pc(config-route-map)"
```

Figure 65. Configuring primary route-map on router r2.

Consider Figure 65. After creating the route map *primary_in*, you will be in the configuration mode of this route map.

Step 4. Type the following command to set the LOCAL_PREF value on router r2. You will use 150 for the primary link.

```
set local-preference 150
```



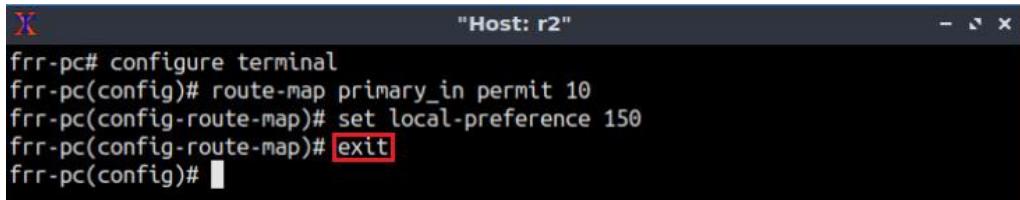
```
frr-pc# configure terminal
frr-pc(config)# route-map primary_in permit 10
frr-pc(config-route-map)# set local-preference 150
frr-pc(config-route-map)#

```

Figure 66. Set BGP LOCAL_PREF on router r2.

Step 5. Type the following command to exit from route-map configuration mode.

```
exit
```



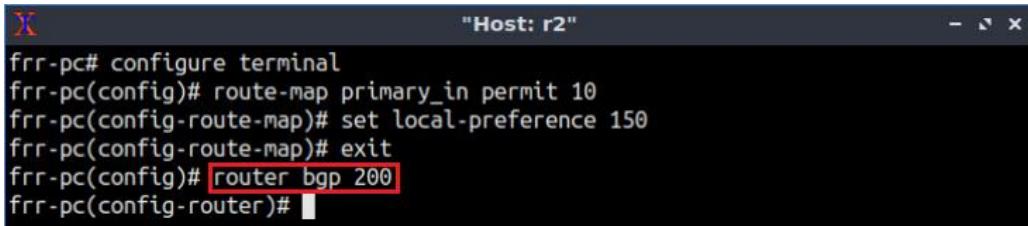
```
frr-pc# configure terminal
frr-pc(config)# route-map primary_in permit 10
frr-pc(config-route-map)# set local-preference 150
frr-pc(config-route-map)# exit
frr-pc(config)#

```

Figure 67. Exiting from route-map configuration mode.

Step 6. To enable router r2 into configuration mode, issue the following command:

```
router bgp 200
```



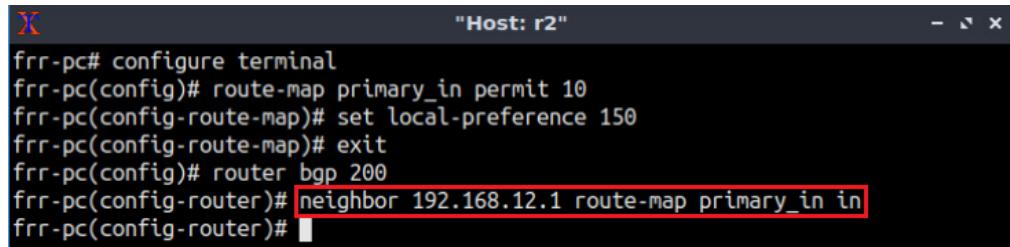
```
frr-pc# configure terminal
frr-pc(config)# route-map primary_in permit 10
frr-pc(config-route-map)# set local-preference 150
frr-pc(config-route-map)# exit
frr-pc(config)# router bgp 200
frr-pc(config-router)#

```

Figure 68. Configuring BGP on router r2.

Step 7. Type the following command to apply the route map *primary_in* on BGP neighbor router r1 (192.168.12.1). Notice that there is *[in]* at the end of the command which specifies that the policy will be applicable for incoming traffic.

```
neighbor 192.168.12.1 route-map primary_in in
```



```
frr-pc# configure terminal
frr-pc(config)# route-map primary_in permit 10
frr-pc(config-route-map)# set local-preference 150
frr-pc(config-route-map)# exit
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.12.1 route-map primary_in in
frr-pc(config-router)#

```

Figure 69. Assigning route-map to the BGP neighbor.

After applying the command depicted in Figure 69, router r2 will set a LOCAL_PREF value of 150 for the link between router r2 and router r1 (primary link).

Step 8. Type the following command to exit from configuration mode.

```
end
```

```
"Host: r2"
frr-pc# configure terminal
frr-pc(config)# route-map primary_in permit 10
frr-pc(config-route-map)# set local-preference 150
frr-pc(config-route-map)# exit
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.12.1 route-map primary_in in
frr-pc(config-router)# end
frr-pc#
```

Figure 70. Exiting from configuration mode.

Step 9. Follow from step 1 to step 7 but with different metrics in order to set the LOCAL_PREF on router r3. Name the route-map as *secondary_in* and set the LOCAL_PREF value to 125. All the steps are summarized in the following figure.

```
"Host: r3"
frr-pc# configure terminal
frr-pc(config)# route-map secondary_in permit 10
frr-pc(config-route-map)# set local-preference 125
frr-pc(config-route-map)# exit
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.13.1 route-map secondary_in in
frr-pc(config-router)# end
frr-pc#
```

Figure 71. Set LOCAL_PREF on router r3.

After setting the LOCAL_PREF values on routers r2 and r3, all the traffic from the Campus network to the ISP will use the primary link. However, if the primary link fails, the secondary link will be used.

Step 10. Type the following command to verify the BGP table of router r3.

```
show ip bgp
```

```

frr-pc# show ip bgp
BGP table version is 4, local router ID is 192.168.23.2, vrf id 0
Default local pref 100, local AS 200
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop           Metric LocPrf Weight Path
*>i192.168.1.0/24  192.168.12.1        0      150      0 100 i
*          192.168.13.1        0      125      0 100 i
*>i192.168.2.0/24  192.168.23.1        0      100      0 i
*> 192.168.3.0/24   0.0.0.0            0            32768 i

Displayed 3 routes and 4 total paths
frr-pc#

```

Figure 72. Verifying BGP networks on router r3.

Consider Figure 72. Router r3 will use the next hop 192.168.12.1 (primary link) to reach the LAN 192.168.1.0/24. Router r3 will not use the next hop 192.168.13.1 (secondary link), unless the primary link is unavailable.

6 Set and verify BGP MED on router r2 and router r3

At this point, routers r2 and r3 will use the primary link to route their traffic to router r1. However, router r1 is not forced to use the primary link to route its traffic to routers r2 and r3. In this section, you will set the MED attribute on routers r2 and r3 so that router r1 always uses the primary link. The MED indicates to external neighbors the preferred path into an AS. A lower metric value is preferred. You will create a route map which will be injected into BGP later.

6.1 Set BGP MED on router r2 and router r3

Step 1. Type the following command to verify the BGP table of router r1.

```
show ip bgp
```

```

frr-pc# show ip bgp
BGP table version is 5, local router ID is 192.168.13.1, vrf id 0
Default local pref 100, local AS 100
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*-> 192.168.1.0/24    0.0.0.0                  0        32768 i
*=> 192.168.2.0/24   192.168.13.2             0        0 200 i
*>   192.168.3.0/24   192.168.12.2             0        0 200 i
*=>   192.168.3.0/24  192.168.12.2             0        0 200 i
*>   192.168.3.0/24   192.168.13.2             0        0 200 i

Displayed 3 routes and 5 total paths
frr-pc#

```

Figure 73. Verifying BGP networks on router r1.

Consider Figure 73. Router r1 uses the secondary link (192.168.13.2) to reach the LAN 192.168.3.0/24.

Step 2. In router r2 terminal, type the following command to enable the configuration mode:

```
configure terminal
```

```

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

```

Figure 74. Enabling configuration mode on router r2.

Step 3. Type the following command to create a route-map named *primary_med_out* with permit clause. Set the sequence-number as 10.

```
route-map primary_med_out permit 10
```

```

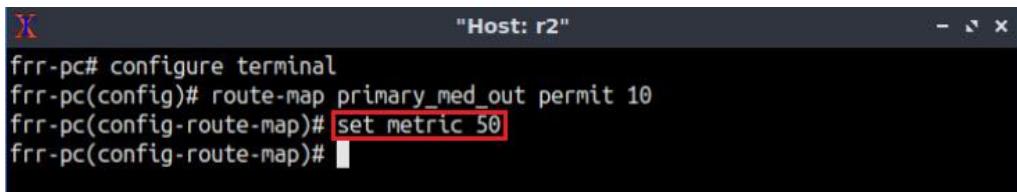
frr-pc# configure terminal
frr-pc(config)# route-map primary_med_out permit 10
frr-pc(config-route-map)#

```

Figure 75. Creating primary route-map on router r2.

Step 4. Type the following command to set the MED value on router r2. You will use 50 for the primary link.

```
set metric 50
```

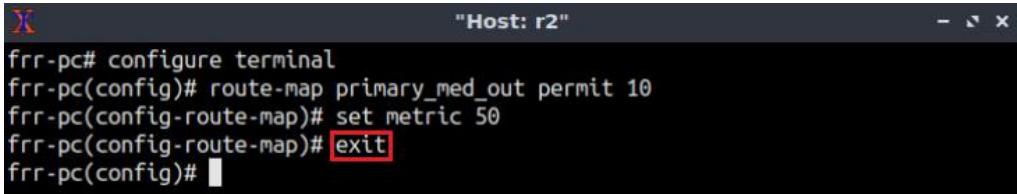


```
frr-pc# configure terminal
frr-pc(config)# route-map primary_med_out permit 10
frr-pc(config-route-map)# set metric 50
frr-pc(config-route-map)#
```

Figure 76. Set metric on router r2.

Step 5. Type the following command to exit from route map configuration mode.

```
exit
```

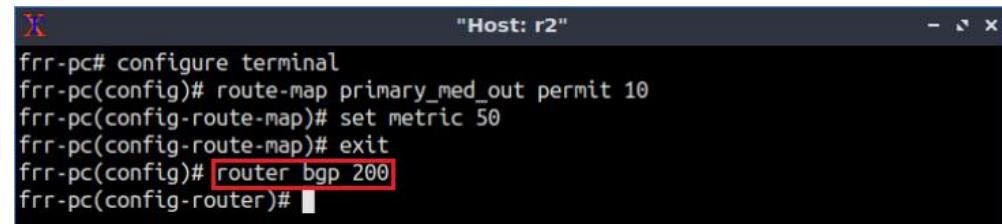


```
frr-pc# configure terminal
frr-pc(config)# route-map primary_med_out permit 10
frr-pc(config-route-map)# set metric 50
frr-pc(config-route-map)# exit
frr-pc(config)#
```

Figure 77. Exiting from route-map configuration mode.

Step 6. Type the following command to enter BGP configuration mode.

```
router bgp 200
```

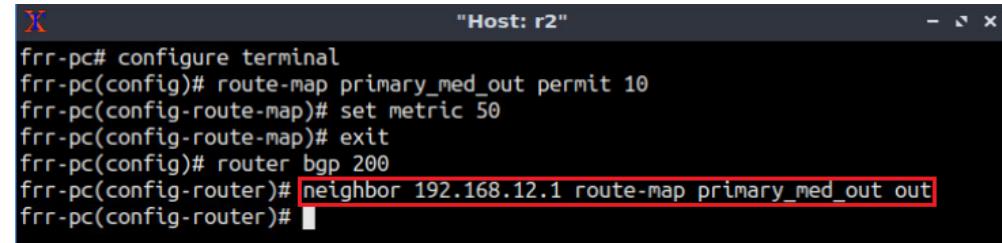


```
frr-pc# configure terminal
frr-pc(config)# route-map primary_med_out permit 10
frr-pc(config-route-map)# set metric 50
frr-pc(config-route-map)# exit
frr-pc(config)# router bgp 200
frr-pc(config-router)#
```

Figure 78. Configuring BGP on router r2.

Step 7. Type the following command to apply the route map *primary_med_out* on BGP neighbor router r1 (192.168.12.1). Notice that there is **out** at the end of the command which specifies that the policy will be applicable for outgoing traffic.

```
neighbor 192.168.12.1 route-map primary_med_out out
```



```
frr-pc# configure terminal
frr-pc(config)# route-map primary_med_out permit 10
frr-pc(config-route-map)# set metric 50
frr-pc(config-route-map)# exit
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.12.1 route-map primary_med_out out
frr-pc(config-router)#
```

Figure 79. Assign route-map to the BGP neighbor.

Step 8. Type the following command to exit from configuration mode.

```
end
```

```
frr-pc# configure terminal
frr-pc(config)# route-map primary_med_out permit 10
frr-pc(config-route-map)# set metric 50
frr-pc(config-route-map)# exit
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.12.1 route-map primary_med_out out
frr-pc(config-router)# end
frr-pc#
```

Figure 80. Exiting from configuration mode.

Step 9. Follow from step 1 to step 8 but with different metrics in order to set the MED value on router r3. Name the route map as *secondary_med_out* and set the MED value to 75. All the steps are summarized in the following figure.

```
frr-pc# configure terminal
frr-pc(config)# route-map secondary_med_out permit 10
frr-pc(config-route-map)# set metric 75
frr-pc(config-route-map)# exit
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.13.1 route-map secondary_med_out out
frr-pc(config-router)# end
frr-pc#
```

Figure 81. Set BGP MED on router r3.

After setting the MED values on routers r2 and r3, all the traffic from the ISP to the Campus network will use the primary link. However, if the primary link fails, the secondary link will be used.

6.2 Verify Configuration

Step 1. Type the following command to verify the BGP table of router r1.

```
show ip bgp
```

```
frr-pc# show ip bgp
BGP table version is 9, local router ID is 192.168.13.1, vrf id 0
Default local pref 100, local AS 100
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop           Metric LocPrf Weight Path
*-> 192.168.1.0/24    0.0.0.0              0        32768 i
*   192.168.2.0/24    192.168.13.2         75        0 200 i
*>   192.168.12.2      192.168.12.2        50        0 200 i
*-> 192.168.3.0/24    192.168.12.2        50        0 200 i
*   192.168.13.2      192.168.13.2        75        0 200 i

Displayed 3 routes and 5 total paths
frr-pc#
```

Figure 82. Verifying BGP networks on router r1.

Consider Figure 82. Router r1 will use the next hop 192.168.12.2 (primary link) to reach the LAN 192.168.3.0/24. Router r1 will not use the next hop 192.168.13.2 (secondary link), unless the primary link is unavailable.

Step 2. In router r3, type the following command to exit the vtysh session:

```
exit
```

Figure 83. Exiting the vtysh session.

You will test the connectivity between the routers and the hosts using `traceroute` command. This command is used to check connectivity where it shows all the path it is taking to reach to the destination.

Use `traceroute` command to test the connectivity between router r3 and host h1 (192.168.1.10). Specify the source IP address 192.168.3.1 using the option `-s`.

Step 3. Run a traceroute test from router r3 to host h1 using the source route IP address 192.168.3.1 as specified below. Router r3 will reach the host h1 via the IP address 192.168.12.1 and 192.168.23.1. Use the following command to test the connectivity.

```
traceroute -s 192.168.3.1 192.168.1.10
```

Figure 84. Traceroute from router r3 to host h1 using the source IP 192.168.3.1.

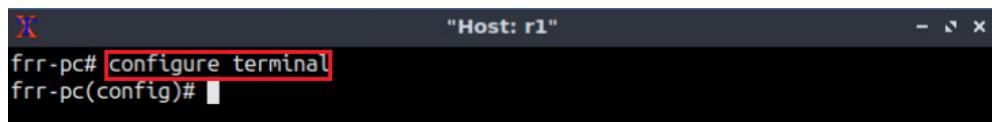
Consider Figure 84. The output of the `traceroute` command shows a successful connectivity test. Packets sent from router r3 (192.168.3.1) will be routed to router r2 (192.168.23.1), then router r1 (192.168.12.1), and finally to host h1 (192.168.1.10).

7 Verify the secondary link

In this section, you will shut down the link between router r1 and router r2 to verify that the secondary link will be used when the primary link is down.

Step 1. In router r1 terminal, type the following command to enable the configuration mode:

```
configure terminal
```

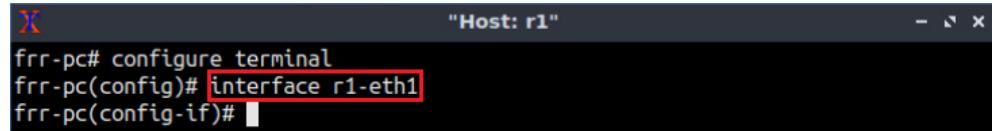


```
frr-pc# configure terminal  
frr-pc(config)#
```

Figure 85. Enabling configuration mode on router r1.

Step 2. In order to configure the interface *r1-eth1* of router r1, type the following command:

```
interface r1-eth1
```

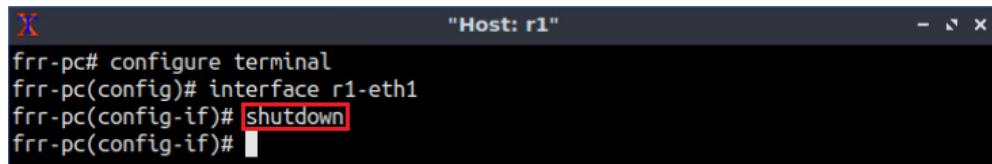


```
frr-pc# configure terminal  
frr-pc(config)# interface r1-eth1  
frr-pc(config-if)#
```

Figure 86. Configuring router r1 interface.

Step 3. Type the following command to shut down the link between router r1 and router r2:

```
shutdown
```

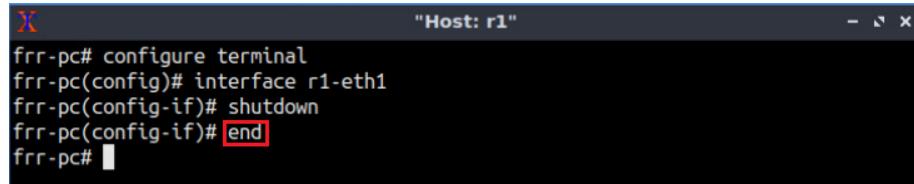


```
frr-pc# configure terminal  
frr-pc(config)# interface r1-eth1  
frr-pc(config-if)# shutdown  
frr-pc(config-if)#
```

Figure 87. Shutting down the link between router r1 and router r2.

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

```
end
```

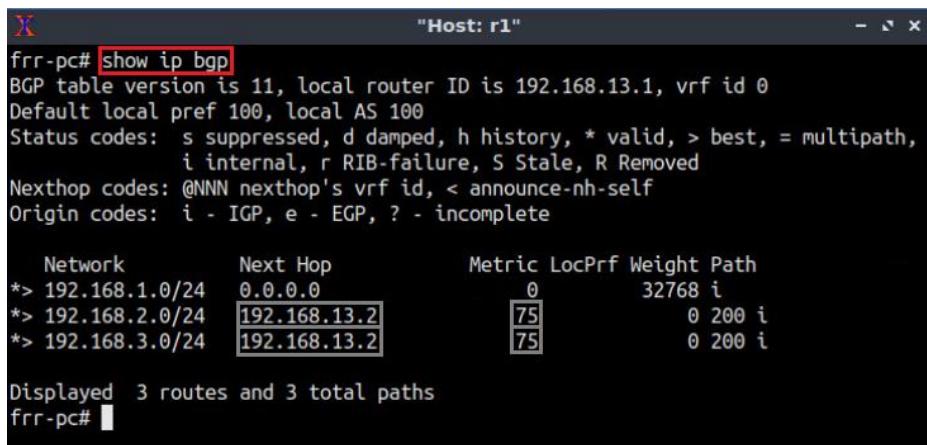


```
frr-pc# configure terminal  
frr-pc(config)# interface r1-eth1  
frr-pc(config-if)# shutdown  
frr-pc(config-if)# end  
frr-pc#
```

Figure 88. Exiting from configuration mode.

Step 5. Type the following command to verify the BGP table of router r1.

```
show ip bgp
```



```
"Host: r1"
frr-pc# show ip bgp
BGP table version is 11, local router ID is 192.168.13.1, vrf id 0
Default local pref 100, local AS 100
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop           Metric LocPrf Weight Path
*-> 192.168.1.0/24  0.0.0.0                  0        32768 i
*-> 192.168.2.0/24  192.168.13.2            75        0 200 i
*-> 192.168.3.0/24  192.168.13.2            75        0 200 i

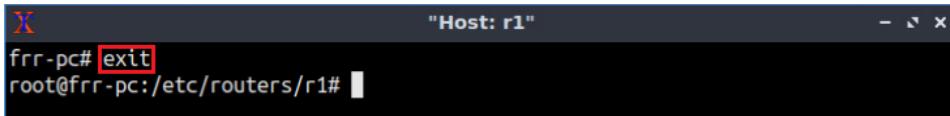
Displayed 3 routes and 3 total paths
frr-pc#"
```

Figure 89. Verifying BGP networks on router r1.

Consider Figure 89. Router r1 will use the secondary link (192.168.13.2) to reach the LANs 192.168.2.0/24 and 192.168.3.0/24.

Step 6. In router r1, type the following command to exit the vtysh session:

```
exit
```

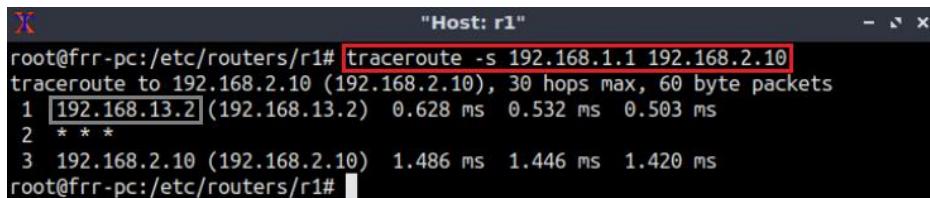


```
"Host: r1"
frr-pc# exit
root@frr-pc:/etc/routers/r1#"
```

Figure 90. Exiting from terminal on router r1.

Step 7. Perform a traceroute test from router r1 to host h2 using the source route IP address 192.168.1.1 as specified below.

```
traceroute -s 192.168.1.1 192.168.2.10
```



```
"Host: r1"
root@frr-pc:/etc/routers/r1# traceroute -s 192.168.1.1 192.168.2.10
traceroute to 192.168.2.10 (192.168.2.10), 30 hops max, 60 byte packets
 1 [192.168.13.2] (192.168.13.2)  0.628 ms  0.532 ms  0.503 ms
 2 * *
 3 192.168.2.10 (192.168.2.10)  1.486 ms  1.446 ms  1.420 ms
root@frr-pc:/etc/routers/r1#"
```

Figure 91. Traceroute test from router r1 to host h2 using source IP 192.168.1.1.

Consider Figure 91. This time router1 will use router r3 (192.168.13.2), i.e., secondary link, to reach host h3.

This concludes Lab 8. Stop the emulation and then exit out of MiniEdit.

References

1. A. Tanenbaum, D. Wetherall, "Computer networks", 5th Edition, Pearson, 2012.

2. Cisco, "What are OSPF areas and virtual links?", 2016. [Online]. Available: <https://www.cisco.com/c/en/us/support/docs/ip/open-shortest-path-first-ospf/13703-8.html>
3. J. Kurose, K. Ross, "Computer networking, a top-down approach," 7th Edition, Pearson, 2017.
4. Cisco, "BGP best path selection algorithm", 2016. [Online]. Available: <https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/13753-25.html#anc2>
5. Cisco, "IP routing: BGP configuration guide, Cisco IOS release 15M&T", 2013. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_bgp/configuration/15-mt/irg-15-mt-book.pdf
6. Cisco, "Understanding BGP MED attribute", 2014. [Online]. Available: <https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/112965-bgpmed-attr-00.html>



BORDER GATEWAY PROTOCOL

Lab 9: IBGP, Next Hop and Full Mesh Topology

Document Version: **2-20-2020**



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 Intradomain and Interdomain routing protocols.....	4
1.2 IBGP Next Hop attribute.....	4
1.3 BGP synchronization and full mesh IBGP	5
2 Lab topology.....	6
2.1 Lab settings.....	7
2.2 Open topology and load the configuration.....	8
2.3 Load zebra daemon and Verify IP addresses	11
3 Configure OSPF on routers r2, r3 and r4	16
4 Configure and verify BGP on routers r1, r2 and r4	21
4.1 Configure BGP on r1, r2 and r4	21
5 Troubleshoot BGP connectivity between routers r1 and r4.....	28
5.1 Examine and troubleshoot IBGP next hop reachability on router r4	28
5.2 Troubleshoot Connectivity problem between routers r1 and r4.	31
6 Configure and verify full mesh IBGP	32
6.1 Configure full mesh IBGP on routers r2, r3, and r4.....	32
6.2 Verify full mesh IBGP on routers r2, r3, and r4.....	33
References	34

Overview

This lab introduces and configures the Border Gateway Protocol (BGP) next hop attribute and explains how this attribute is transmitted in External BGP (EBGP) and Internal BGP (IBGP). Furthermore, the lab presents full mesh topology used in IBGP. In this lab, BGP will be configured as the Exterior Gateway Protocol (EGP) and Open Shortest Path First (OSPF) as the Interior Gateway Protocol (IGP).

Objectives

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

1. Configure OSPF and BGP.
2. Understand BGP next hop attribute.
3. Troubleshoot and resolve next hop attribute issues in IBGP.
4. Configure full mesh IBGP.
5. Verify the connectivity of the configured topology.

Lab settings

The information in Table 1 provides the credentials to access Client1 machine.

Table 1. Credentials to access Client1 machine.

Device	Account	Password
Client1	admin	password

Lab roadmap

This lab is organized as follows:

1. Section 1: Introduction.
2. Section 2: Lab topology.
3. Section 3: Configure OSPF on routers r2, r3 and r4.
4. Section 4: Configure and verify BGP on routers r1, r2 and r4.
5. Section 5: Troubleshoot BGP connectivity between routers r1 and r4.
6. Section 6: Configure and verify full mesh IBGP.

1 Introduction

1.1 Intradomain and Interdomain routing protocols

The Internet consists of many independent administrative domains, referred to as Autonomous Systems (ASes). ASes are operated by different organizations, which can run their own internal routing protocols. A routing protocol that runs within an AS is referred to as intradomain routing protocol. One of the most widely used intradomain protocols is OSPF. Since an AS may be large and nontrivial to manage, OSPF allows an AS to be divided into numbered areas¹. An area is a logical collection of networks, routers, and links. All routers in the same area have detailed information of the topology within their area².

A routing protocol that runs between ASes is referred to as interdomain routing protocol. ASes may use different intradomain routing protocols; however, they must use the same interdomain routing protocol, i.e., BGP. BGP allows the enforcement of different routing policies on the traffic from one AS to another. For example, a security policy can prevent the dissemination of routing information from one AS to another¹.

BGP is referred to as External BGP (EBGP) when it is running between different ASes, whereas it is referred to as Internal BGP (IBGP) when it is running within an AS¹. IBGP is usually used to distribute the routes learned using EBGP among the routers within the same AS¹.

Consider Figure 1. The intradomain routing protocol within AS 100 is OSPF, and the interdomain routing protocol between AS 100 and AS 200 is BGP (EBGP). Routers within the same AS advertise their EBGP learned routes among each other through IBGP.

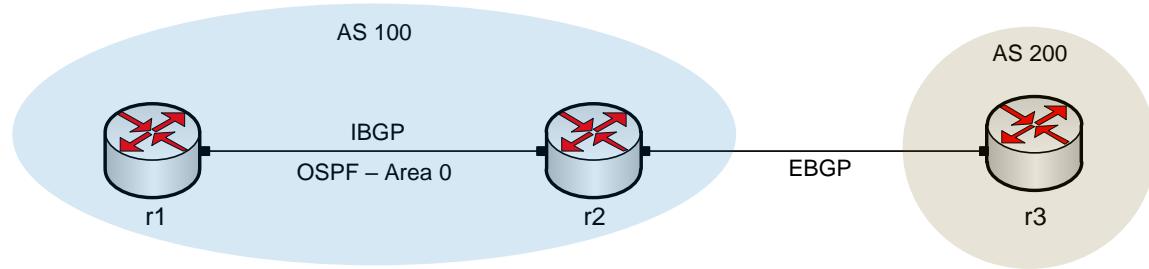


Figure 1. Routers that exchange information within the same AS use OSPF and IBGP, while routers that exchange information between different ASes use EBGP.

1.2 IBGP Next Hop attribute

In BGP, when a router advertises a route across a BGP session, i.e., between two routers running BGP, it includes the next hop attribute in the advertisement. This attribute determines the next hop IP address to use in order to reach a destination. For EBGP, the next hop attribute is updated to be the IP address of the EBGP neighbor. However, for IBGP, the routers don't update this attribute, and the next hop that EBGP advertises should be carried into IBGP³.

Consider Figure 2. In an EBGP route advertisement, the next hop attribute will be the IP address of the EBGP neighbor that advertised the route (router r1). The next hop attribute advertised by an EBGP peer (router r1) will be forwarded to all IBGP peers (routers r2 and r3) without being updated.

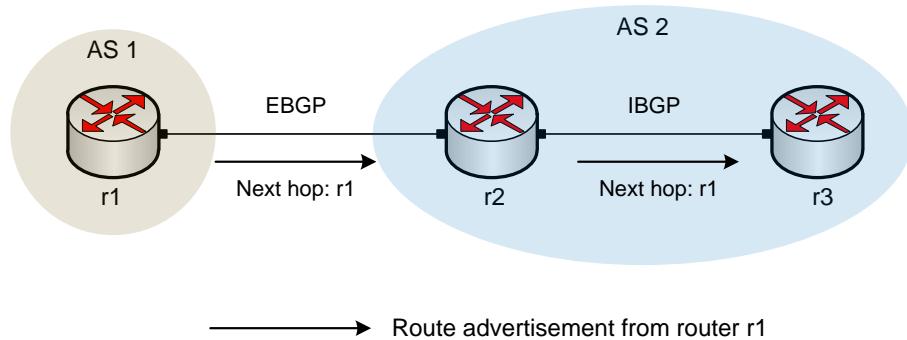


Figure 2. Router r2 advertises an EBGP learned route to its IBGP neighbor (router r3) without updating the next hop attribute.

Routers can be configured to update the next hop attribute before they advertise EBGP learned routes it to their IBGP peers. In Figure 3, router r2 updates the next hop attribute to itself (next-hop-self) before it advertises it to its IBGP peers, i.e., router r3.

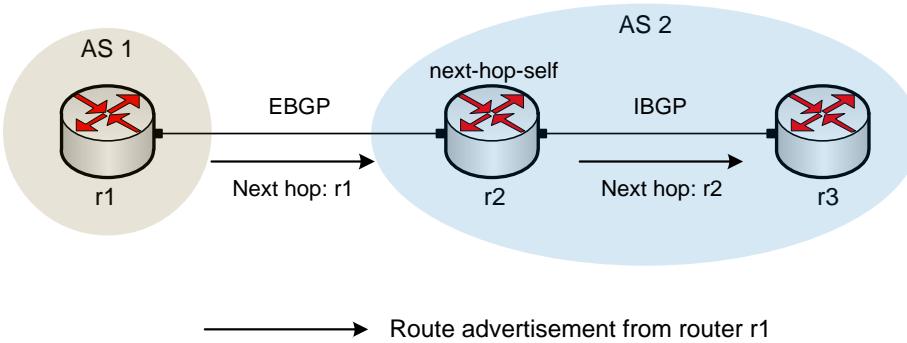


Figure 3. Router r2 updates the next hop attribute of the EBGP learned route before it advertises it to router r3.

1.3 BGP synchronization and full mesh IBGP

The BGP synchronization rule states that a router will not include in its routing table nor advertise routes learned by IBGP unless that route is directly connected or learned from an IGP³. By default, BGP has the synchronization rule disabled³.

Consider Figure 4. Router r2 advertises the route information that are learned from router r1 to router r3. Since BGP synchronization is disabled, router r3 will include the route information received from its IBGP peer (router r2) in its routing table. Thus, router r3 will be able to reach the advertised networks from router r1.

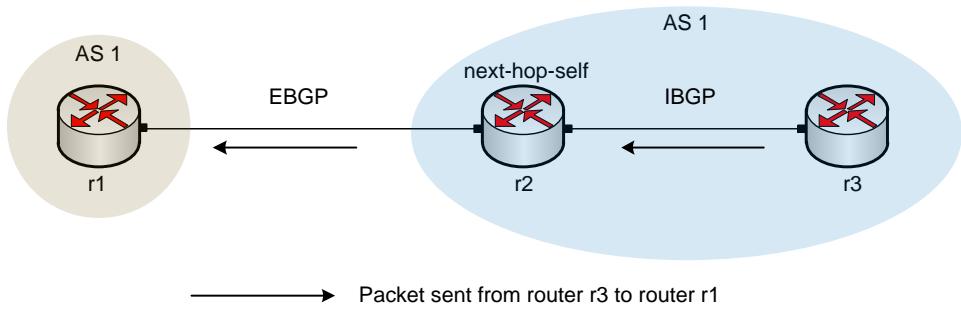


Figure 4. With the default BGP no synchronization rule, router r3 will include the route information learned via IBGP in its routing table.

A topology is called full mesh (fully meshed topology) when there is an IBGP peering relationship between any two routers in the AS. An Example of a full mesh topology is shown in Figure 5.

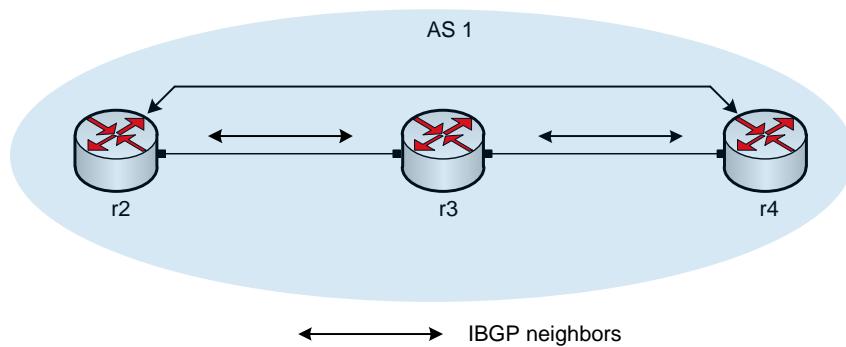


Figure 5. Full mesh IBGP topology.

2 Lab topology

Consider Figure 6. The lab topology consists of two ASes, each identified by an Autonomous System Number (ASN). The Internet Service Provider (ISP), i.e., router r1, provides Internet service to the Campus network (routers r2, r3 and r4). The ASN assigned to the ISP and the Campus network are 100 and 200, respectively. The ISP communicates with the Campus via EBGP routing protocol, and the routers within the Campus network communicate using IBGP and OSPF. Additionally, all the routers have a loopback interface for testing purposes.

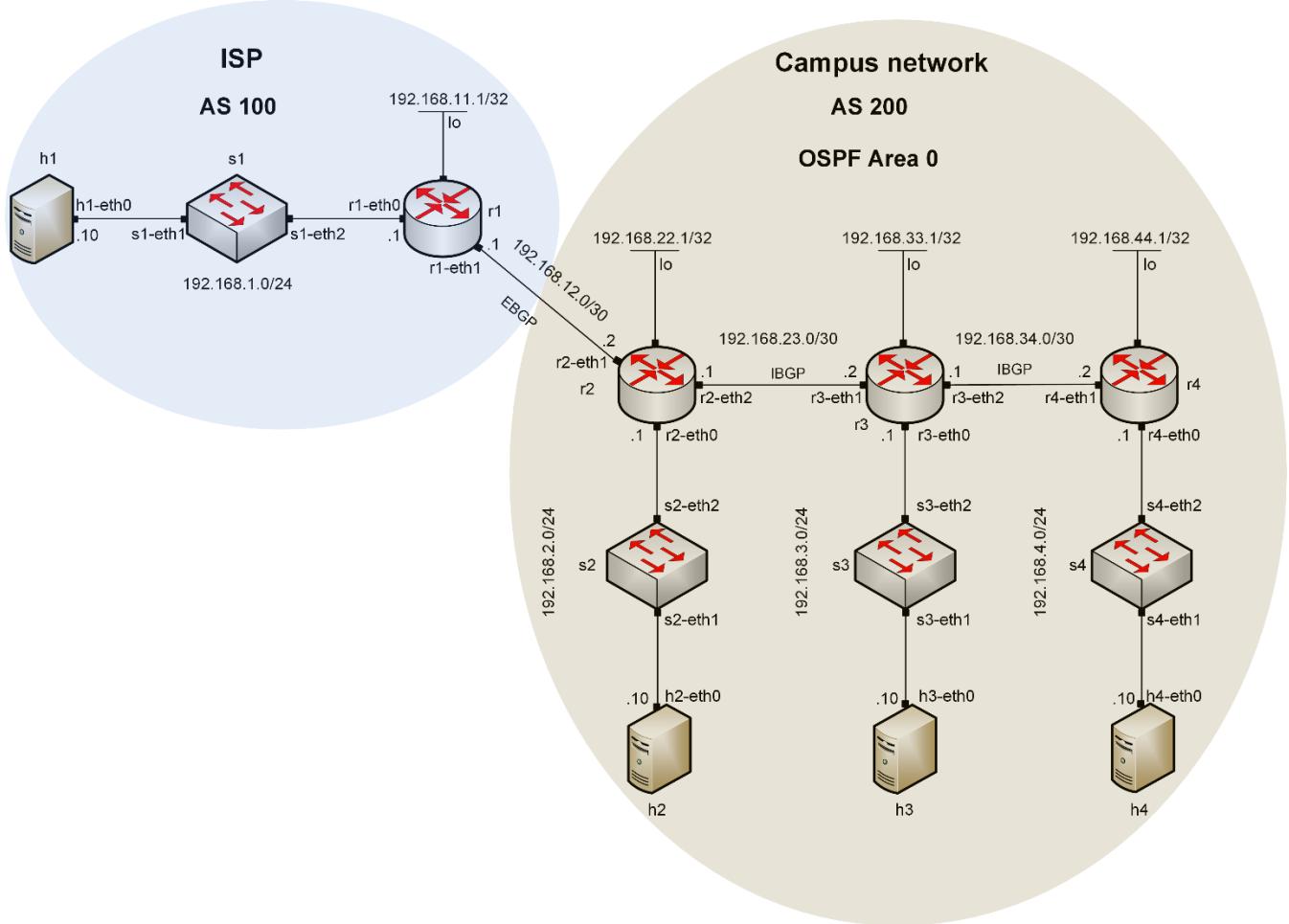


Figure 6. Lab topology.

2.1 Lab settings

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

Table 2. Topology information.

Device	Interface	IPV4 Address	Subnet	Default gateway
r1 (ISP)	r1-eth0	192.168.1.1	/24	N/A
	r1-eth1	192.168.12.1	/30	N/A
	lo	192.168.11.1	/32	N/A
r2 (Campus network)	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
	lo	192.168.22.1	/32	N/A
r3 (Campus network)	r3-eth0	192.168.3.1	/24	N/A
	r3-eth1	192.168.23.2	/30	N/A
	r3-eth2	192.168.34.1	/30	N/A
	lo	192.168.33.1	/32	N/A
r4 (Campus network)	r4-eth0	192.168.4.1	/24	N/A
	r4-eth1	192.168.34.2	/30	N/A
	lo	192.168.44.1	/32	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
h3	h3-eth0	192.168.3.10	/24	192.168.3.1
h4	h4-eth0	192.168.4.10	/24	192.168.4.1

2.2 Open topology and load the configuration

Step 1. Start by launching Miniedit by clicking on Desktop's shortcut. When prompted for a password, type `password`.



Figure 7. MiniEdit shortcut.

Step 2. On Miniedit's menu bar, click on *File* then *open* to load the lab's topology. Locate the *Lab9.mn* topology file in the default directory, */home/frr/BGP_Labs/lab9* and click on *Open*.

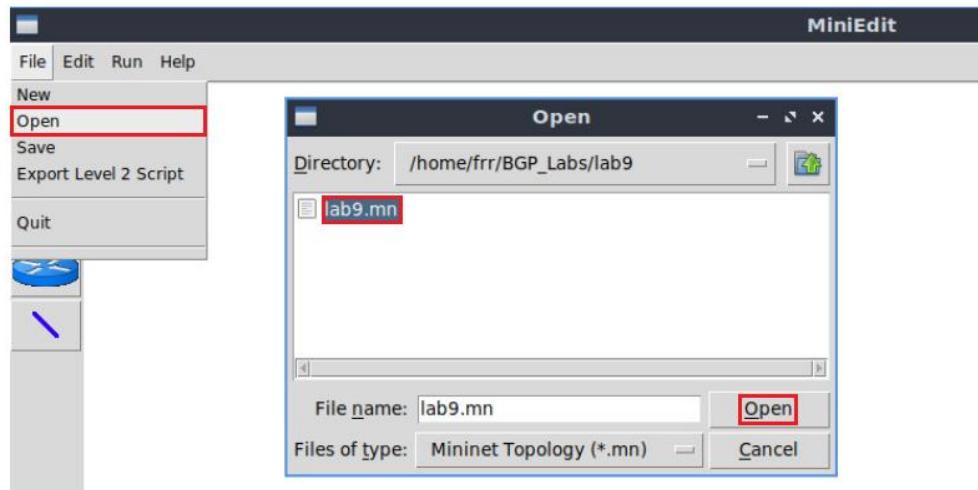


Figure 8. MiniEdit's Open dialog.

At this point the topology is loaded with all the required network components. You will execute a script that will load the configuration of the routers.

Step 3. Open the Linux terminal.

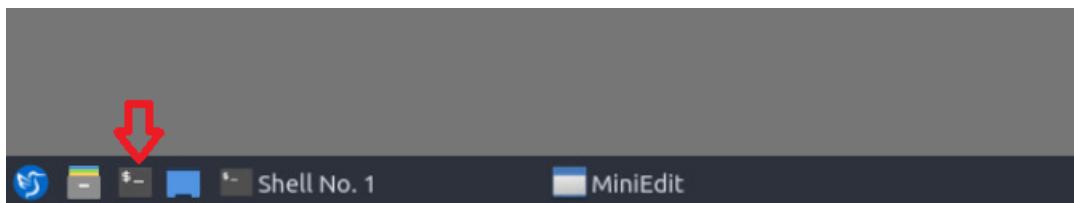


Figure 9. Opening Linux terminal

Step 4. Click on the Linux's terminal and navigate into *BGP_Labs/lab9* 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 BGP_Labs/lab9
```

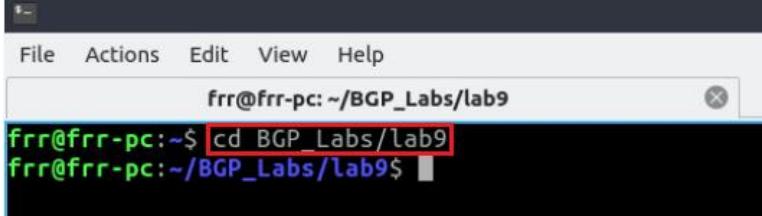


Figure 10. Entering the *BGP_Labs/lab9* directory.

Step 5. To execute the shell script, type the following command. The argument of the program corresponds to the configuration zip file that will be loaded in all the routers in the topology.

```
./config_loader.sh lab9_conf.zip
```

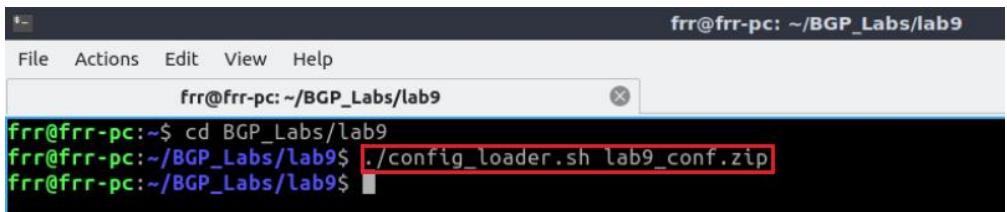


Figure 11. Executing the shell script to load the configuration.

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

```
exit
```

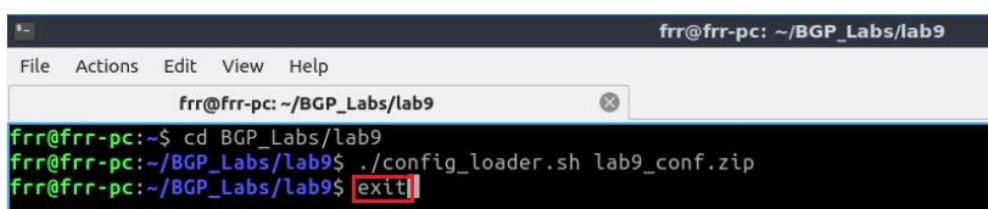


Figure 12. Exiting from the terminal.

Step 7. At this point hosts h1, h2, h3 and h4 interfaces are configured. To proceed with the emulation, click on the *Run* button located in lower left-hand side.



Figure 13. Starting the emulation.

Step 8. Click on Mininet's terminal, i.e., the one launched when MiniEdit was started.

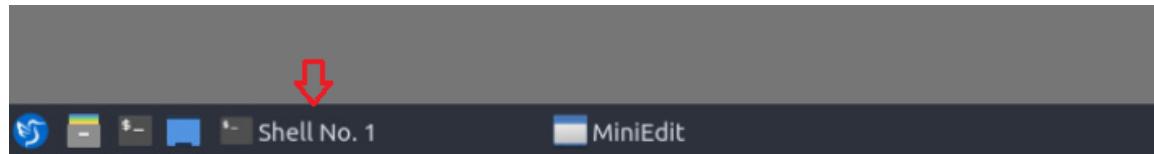


Figure 14. Opening Mininet's terminal.

Step 9. Issue the following command to display the interface names and connections.

A screenshot of the "Shell No. 1" terminal window. The window has a menu bar with File, Actions, Edit, View, and Help. Below the menu is a sub-menu titled "Shell No. 1" with a close button. The main area of the window shows the command-line interface of Mininet. A red box highlights the command "links". The output of the command is a list of network connections: 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), and r3-eth2<->r4-eth1 (OK OK). The prompt "mininet>" is visible at the bottom.

Figure 15. Displaying network interfaces.

In Figure 15, the link displayed within the gray box indicates that interface *eth0* of host *h1* connects to interface *eth1* of switch *s1* (i.e., *h1-eth0<->s1-eth1*).

2.3 Load zebra daemon and Verify IP addresses

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

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.

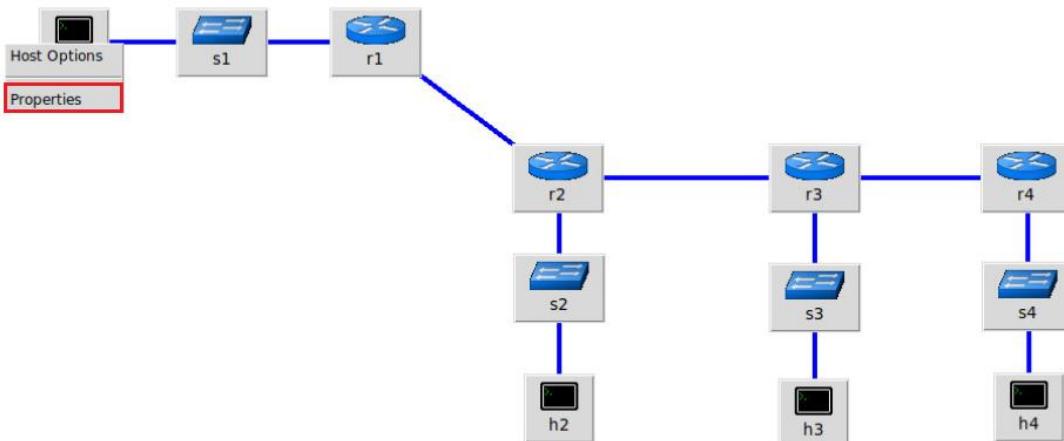


Figure 16. Opening terminal on host h1.

Step 2. On host h1 terminal, type the command shown below to verify that the IP address was assigned successfully. You will verify that host h1 has an interface, *h1-eth0* 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 17. Output of `ifconfig` command.

Step 3. On 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 18. Output of `route` command.

Step 4. In order to verify hosts h2, h3 and h4, proceed similarly by repeating from step 1 to step 3 on host h2, h3 and h4 terminals. Similar results should be observed.

Step 5. You will validate that the router interfaces are configured correctly according to Table 2. In order to verify router r1, hold right-click on router r1 and select Terminal.

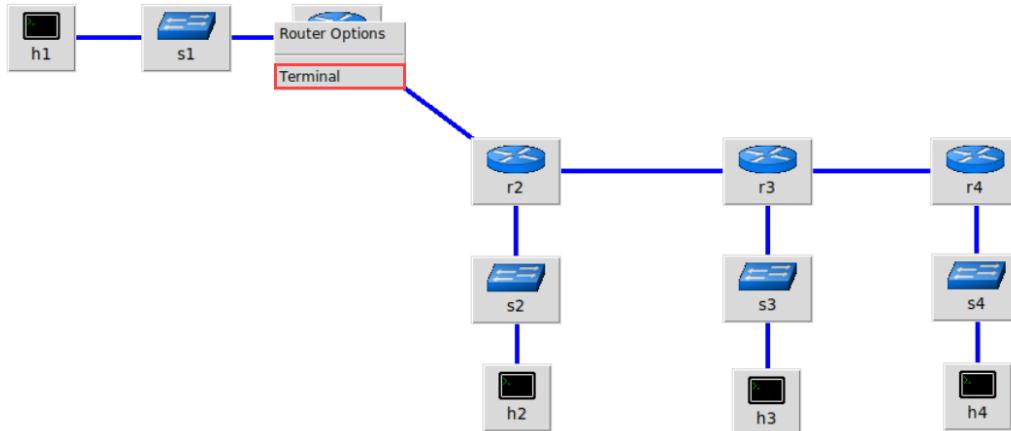
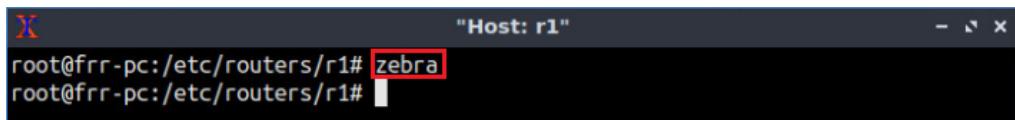


Figure 19. Opening terminal on router r1.

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

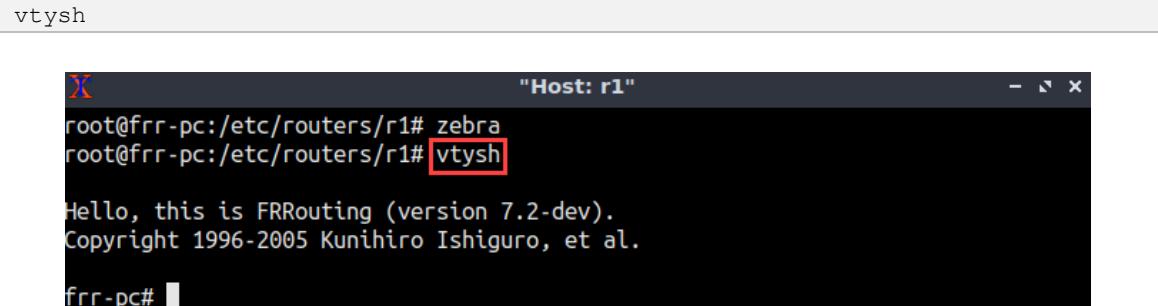
```
zebra
```



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

Figure 20. Starting zebra daemon.

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



```
vtysh
```



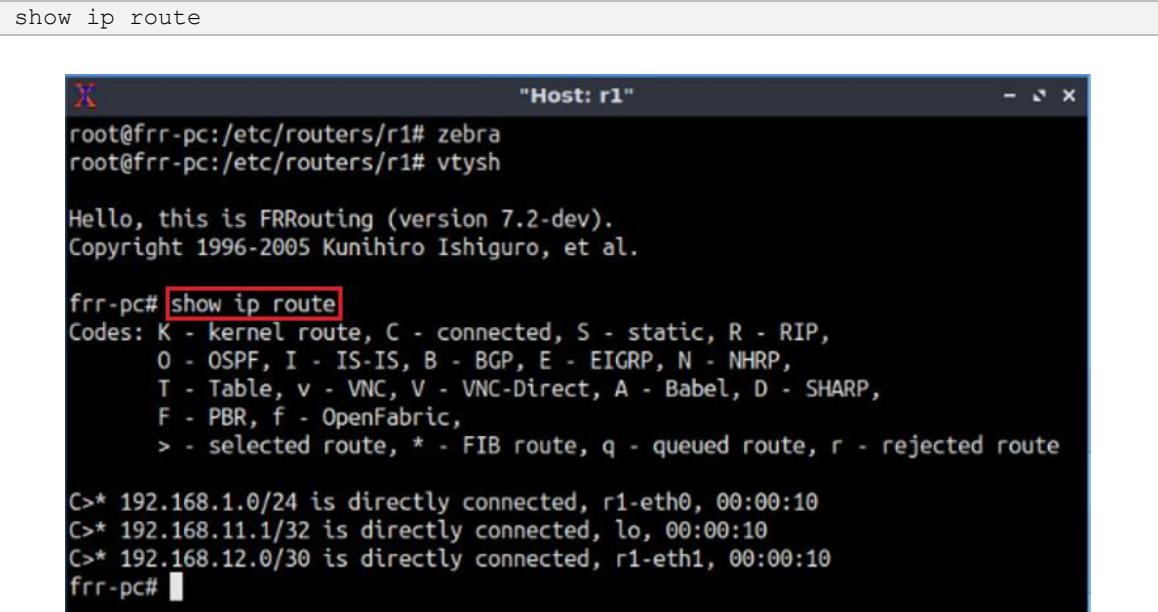
```
"Host: r1"
root@frr-pc:/etc/routers/r1# zebra
root@frr-pc:/etc/routers/r1# vtysh
```

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

```
frr-pc#
```

Figure 21. 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 networks attached to routers r2 (192.168.2.0/24), r3 (192.168.3.0/24) and r4 (192.168.4.0/24) 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.

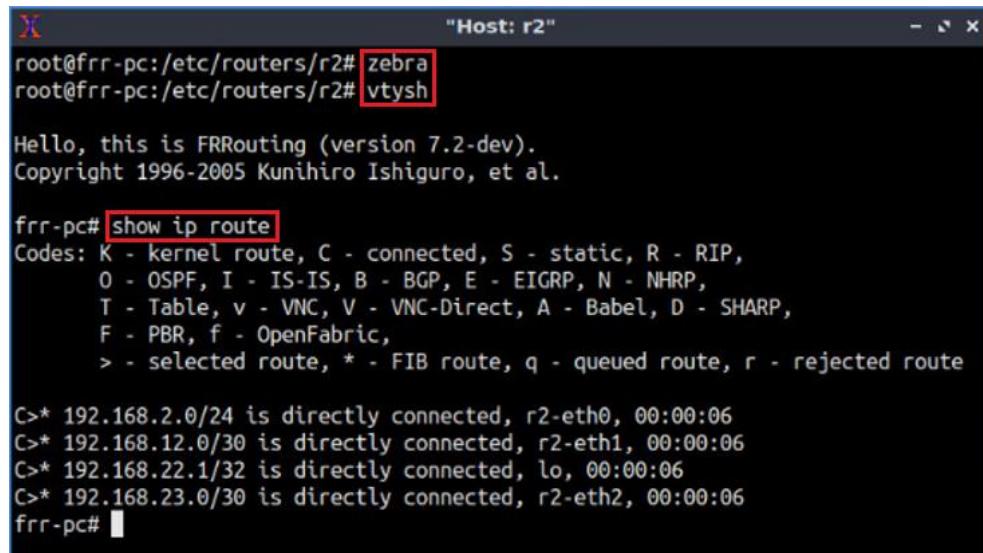
```
frr-pc# show ip route
```

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, 00:00:10
C>* 192.168.11.1/32 is directly connected, lo, 00:00:10
C>* 192.168.12.0/30 is directly connected, r1-eth1, 00:00:10
frr-pc#
```

Figure 22. Displaying the routing table of router r1.

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.



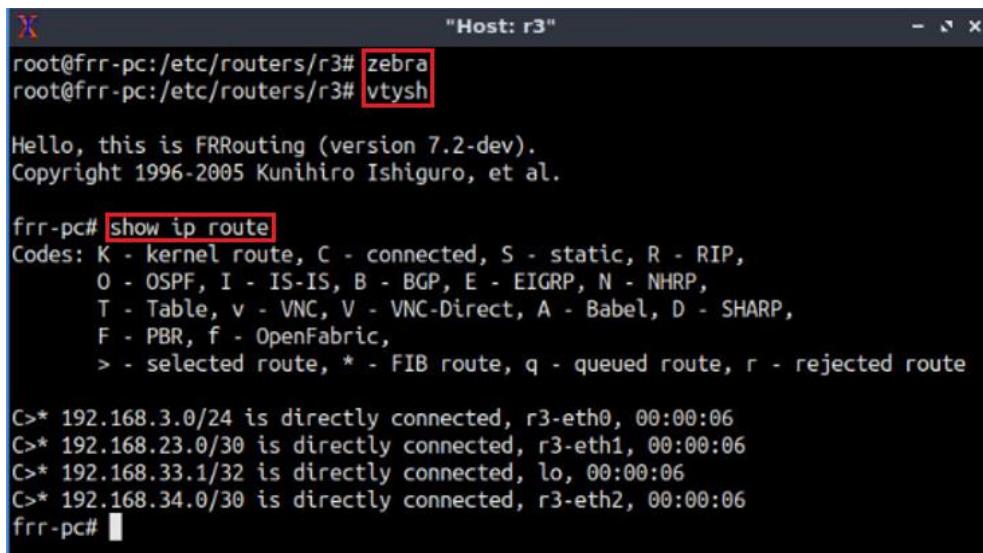
The terminal window shows the configuration of router r2. It starts with 'zebra' and 'vtysh' commands. Then it displays the routing table with the command 'show ip route'. The output shows the following routes:

```
Codes: K - kernel route, C - connected, S - static, R - RIP,
      0 - 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, 00:00:06
C>* 192.168.12.0/30 is directly connected, r2-eth1, 00:00:06
C>* 192.168.22.1/32 is directly connected, lo, 00:00:06
C>* 192.168.23.0/30 is directly connected, r2-eth2, 00:00:06
```

Figure 23. Displaying the routing table of router r2.

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



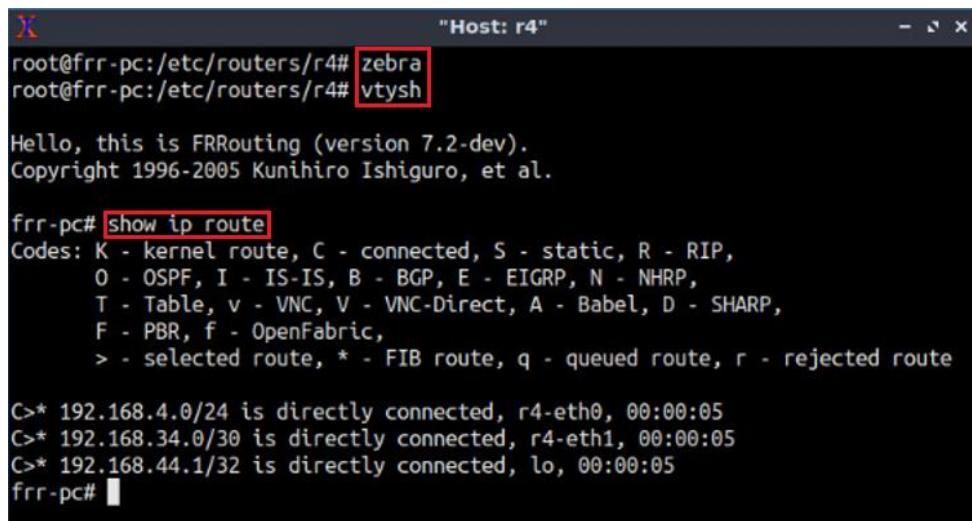
The terminal window shows the configuration of router r3. It starts with 'zebra' and 'vtysh' commands. Then it displays the routing table with the command 'show ip route'. The output shows the following routes:

```
Codes: K - kernel route, C - connected, S - static, R - RIP,
      0 - 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>* 192.168.3.0/24 is directly connected, r3-eth0, 00:00:06
C>* 192.168.23.0/30 is directly connected, r3-eth1, 00:00:06
C>* 192.168.33.1/32 is directly connected, lo, 00:00:06
C>* 192.168.34.0/30 is directly connected, r3-eth2, 00:00:06
```

Figure 24. Displaying the routing table of router r3.

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



```
"Host: r4"
root@frr-pc:/etc/routers/r4# zebra
root@frr-pc:/etc/routers/r4# 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>* 192.168.4.0/24 is directly connected, r4-eth0, 00:00:05
C>* 192.168.34.0/30 is directly connected, r4-eth1, 00:00:05
C>* 192.168.44.1/32 is directly connected, lo, 00:00:05
frr-pc#
```

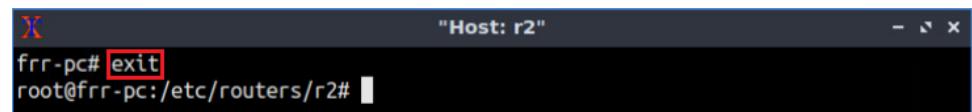
Figure 25. Displaying the routing table of router r4.

3 Configure OSPF on routers r2, r3 and r4

In this section, you will configure OSPF routing protocol on routers r2, r3 and r4. First, you will enable the OSPF daemon on the routers. Second, you will establish a single area OSPF, which is classified as area 0 or backbone area. Finally, all the directly connected networks (except 192.168.12.0/30) will be advertised in area 0 between routers r2, r3 and r4. Network 192.168.12.0/30 will be used to configure EBGP between routers r1 and r2 in the following section.

Step 1. To configure OSPF routing protocol, you need to enable the OSPF daemon first. In router r2, type the following command to exit the vtysh session.

```
exit
```

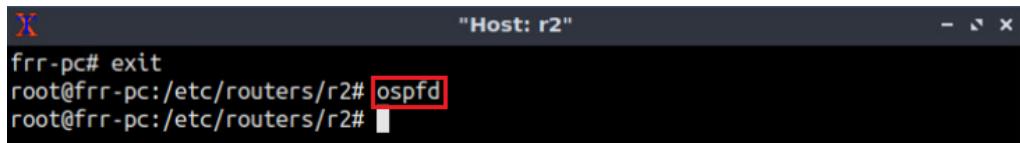


```
"Host: r2"
frr-pc# exit
root@frr-pc:/etc/routers/r2#
```

Figure 26. Exiting the vtysh session.

Step 2. Type the following command on router r2 terminal to enable OSPF daemon.

```
ospfd
```



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

Figure 27. Starting OSPF daemon.

Step 3. In order to enter to router r2 terminal, issue the following command:

```
vtysh
```



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

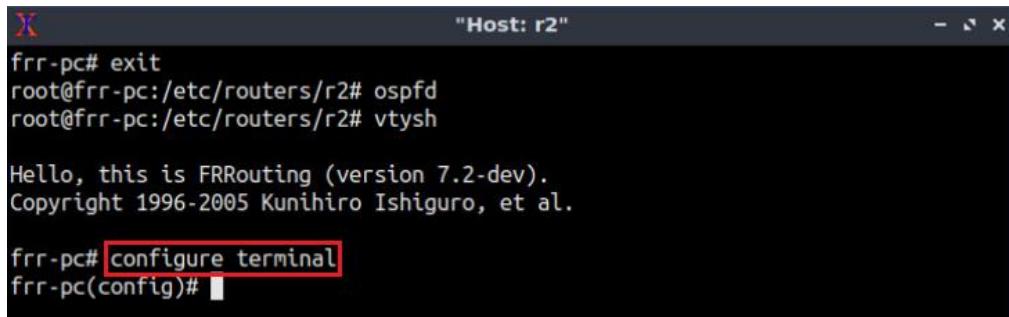
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc#
```

Figure 28. Starting vtysh on router r2.

Step 4. To enable router r2 configuration mode, issue the following command:

```
configure terminal
```



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

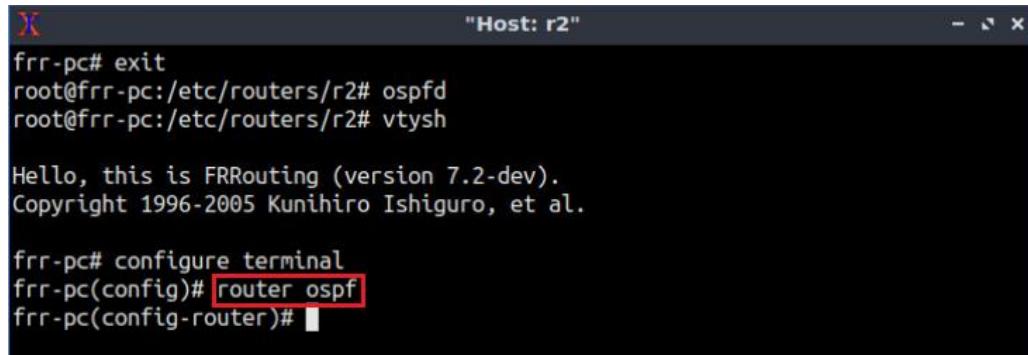
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

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

Figure 29. Enabling configuration mode on router r2.

Step 5. In order to configure OSPF routing protocol, type the command shown below. This command will enable OSPF configuration mode where you can advertise the networks directly connected to the router r2.

```
router ospf
```



```
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)#
```

Figure 30. Configuring OSPF on router r2.

Step 6. In this step, type the following command to enable the interface *r2-eth2*, corresponding to the network 192.168.23.0/30, to participate in the routing process. This network is associated with area 0.

```
network 192.168.23.0/30 area 0
```

```
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)# network 192.168.23.0/30 area 0
frr-pc(config-router)#
```

Figure 31. Enabling the interface corresponding to the network 192.168.23.0/30 to participate in the OSPF routing process.

Step 7. Type the following command to enable the loopback interface 192.168.22.1/24 to participate in the routing process.

```
network 192.168.22.1/32 area 0
```

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

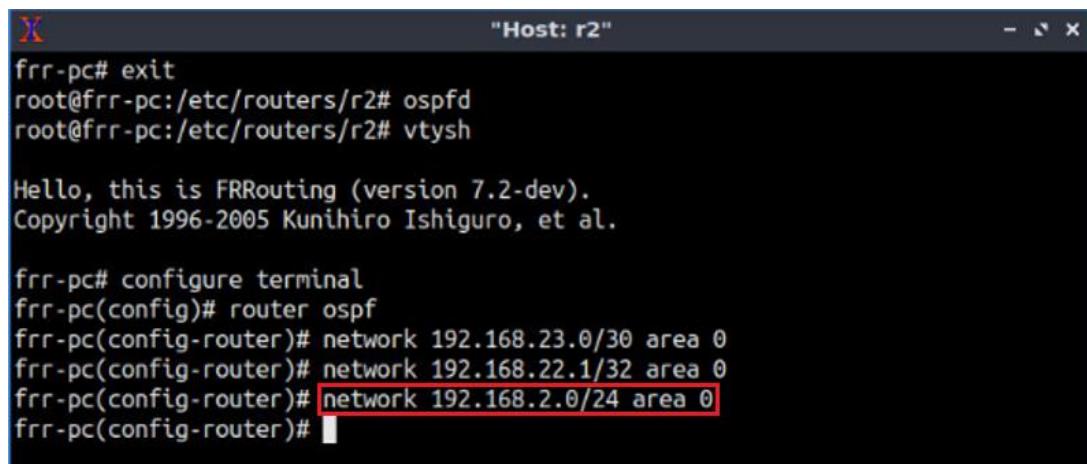
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# network 192.168.23.0/30 area 0
frr-pc(config-router)# network 192.168.22.1/32 area 0
frr-pc(config-router)#
```

Figure 32. Enabling the interface corresponding to 192.168.22.1/32 to participate in the OSPF routing process.

Step 8. Similarly, type the following command in router r2 terminal to enable the interface *r2-eth0* to participate in the OSPF routing process.

```
network 192.168.2.0/24 area 0
```



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

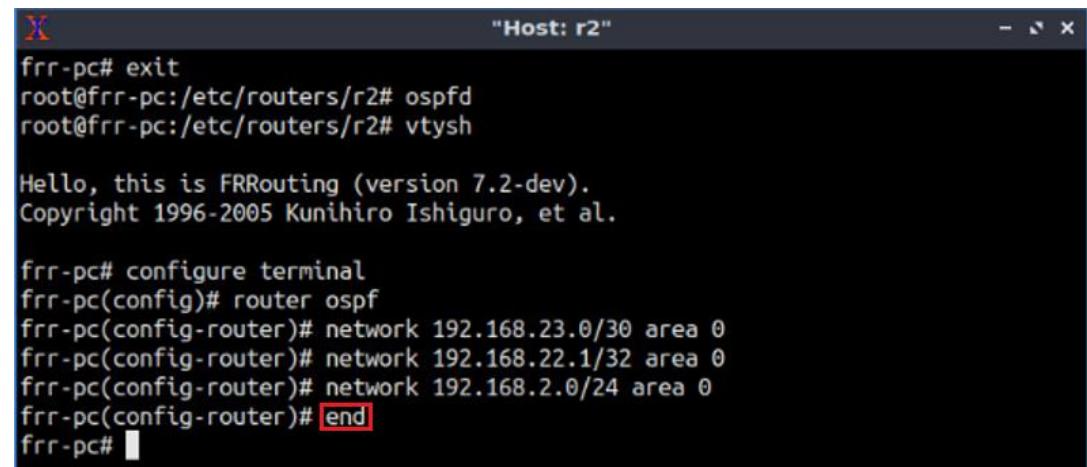
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# network 192.168.23.0/30 area 0
frr-pc(config-router)# network 192.168.22.1/32 area 0
frr-pc(config-router)# network 192.168.2.0/24 area 0
frr-pc(config-router)#
```

Figure 33. Enabling the interface corresponding to the network 192.168.2.0/24 to participate in the OSPF routing process.

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

```
end
```



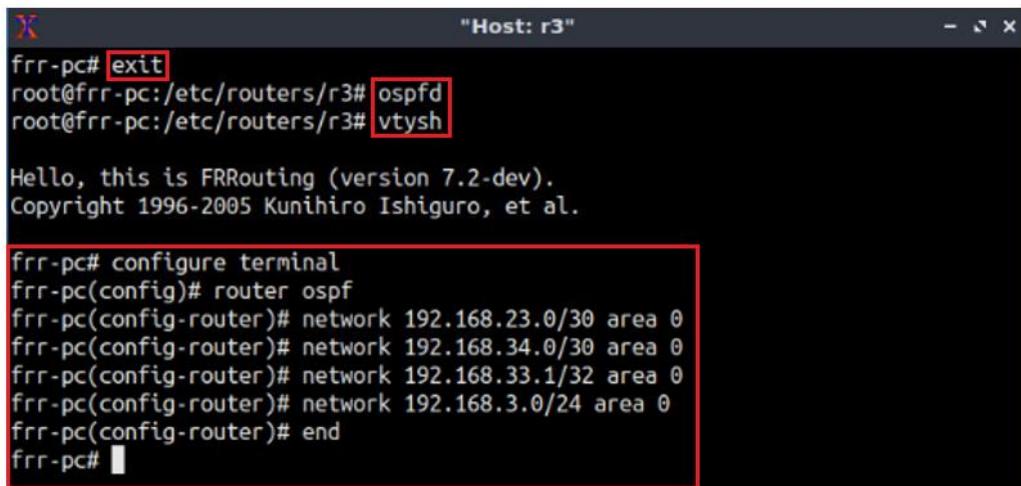
```
frr-pc# exit
root@frr-pc:/etc/routers/r2# ospfd
root@frr-pc:/etc/routers/r2# vtysh

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)# network 192.168.23.0/30 area 0
frr-pc(config-router)# network 192.168.22.1/32 area 0
frr-pc(config-router)# network 192.168.2.0/24 area 0
frr-pc(config-router)# end
frr-pc#
```

Figure 34. Exiting from configuration mode.

Step 10. 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.

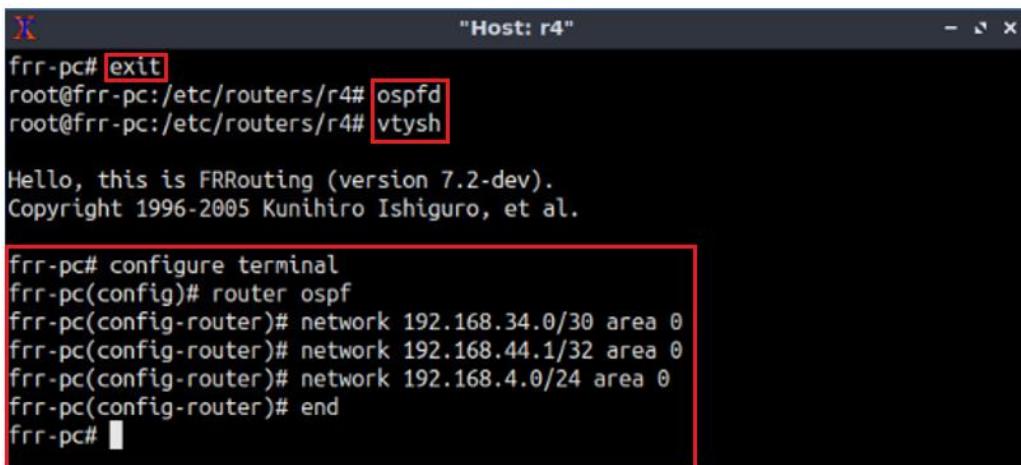


The terminal window shows the configuration of OSPF on router r3. The commands entered are:

```
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)# network 192.168.23.0/30 area 0  
frr-pc(config-router)# network 192.168.34.0/30 area 0  
frr-pc(config-router)# network 192.168.33.1/32 area 0  
frr-pc(config-router)# network 192.168.3.0/24 area 0  
frr-pc(config-router)# end  
frr-pc#
```

Figure 35. Configuring OSPF on router r3.

Step 11. Router r4 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 r4 terminal issue the commands depicted below.



The terminal window shows the configuration of OSPF on router r4. The commands entered are:

```
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)# network 192.168.34.0/30 area 0  
frr-pc(config-router)# network 192.168.44.1/32 area 0  
frr-pc(config-router)# network 192.168.4.0/24 area 0  
frr-pc(config-router)# end  
frr-pc#
```

Figure 36. Configuring OSPF on router r4.

Step 12. Type the following command to verify the routing table of router r4.

```
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 [110/30] via 192.168.34.1, r4-eth1, 00:01:18
O>* 192.168.3.0/24 [110/20] via 192.168.34.1, r4-eth1, 00:01:18
O  192.168.4.0/24 [110/10] is directly connected, r4-eth0, 00:01:03
C>* 192.168.4.0/24 is directly connected, r4-eth0, 00:18:25
O>* 192.168.22.1/32 [110/20] via 192.168.34.1, r4-eth1, 00:01:18
O>* 192.168.23.0/30 [110/20] via 192.168.34.1, r4-eth1, 00:01:18
O>* 192.168.33.1/32 [110/10] via 192.168.34.1, r4-eth1, 00:01:18
O  192.168.34.0/30 [110/10] is directly connected, r4-eth1, 00:01:28
C>* 192.168.34.0/30 is directly connected, r4-eth1, 00:18:25
O  192.168.44.1/32 [110/0] is directly connected, lo, 00:01:16
C>* 192.168.44.1/32 is directly connected, lo, 00:18:25
frr-pc# 

```

Figure 37. Displaying the routing table of router r4.

Consider Figure 37. Router r4 reaches the network 192.168.2.0/24 via the IP address 192.168.34.1. Networks 192.168.4.0/24 and 192.168.34.0/30 have two available paths from router r4. The administrative distance (AD) of the paths advertised through OSPF is 110. The AD is a value used by routers to select the best path when there are multiple available routes to the same destination. A smaller AD is always preferable to the routers. The characters **>*** indicates that the following path is used to reach a specific network. Router r3 prefers directly connected networks over OSPF since the former has a lower AD than the latter.

Step 13. In router r4 terminal, perform a connectivity test by running the command shown below. To stop the test, press **Ctrl+c**. The result will show a successful connectivity test between router r4 and host h2.

```

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=0.536 ms
64 bytes from 192.168.2.10: icmp_seq=2 ttl=62 time=0.095 ms
64 bytes from 192.168.2.10: icmp_seq=3 ttl=62 time=0.098 ms
^C
--- 192.168.2.10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 36ms
rtt min/avg/max/mdev = 0.095/0.243/0.536/0.207 ms
frr-pc# 

```

Figure 38. Connectivity test using **ping** command.

4 Configure and verify BGP on routers r1, r2 and r4

4.1 Configure BGP on r1, r2 and r4

In this section, you will configure BGP on all routers. Routers r2 and r4 communicate with router r1 through EBGP, while router r2 communicates with router r4 through IBGP. You will assign BGP neighbors to allow the routers to exchange BGP routes. Furthermore, routers r1, r2, and r3 will advertise their LANs via BGP so that the LANs are learned by peer routers.

Step 1. To configure BGP routing protocol, you need to enable the BGP daemon first. In router r1 terminal, type the following command to exit the vtysh session:

```
exit
```

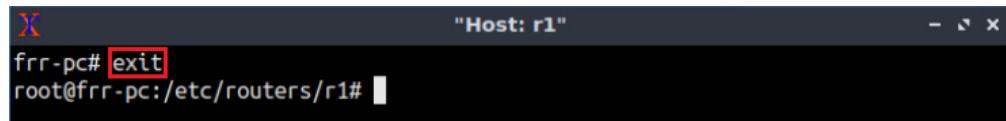


Figure 39. Exiting the vtysh session.

Step 2. Type the following command on r1 terminal to enable and start BGP routing protocol.

```
bgpd
```

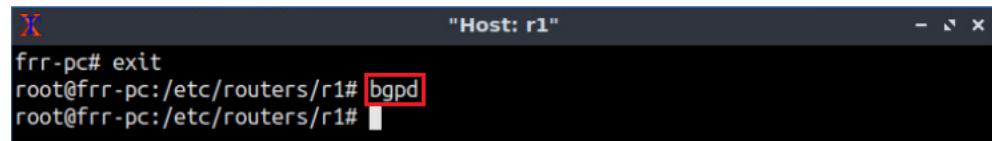


Figure 40. Starting BGP daemon.

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

```
vtysh
```

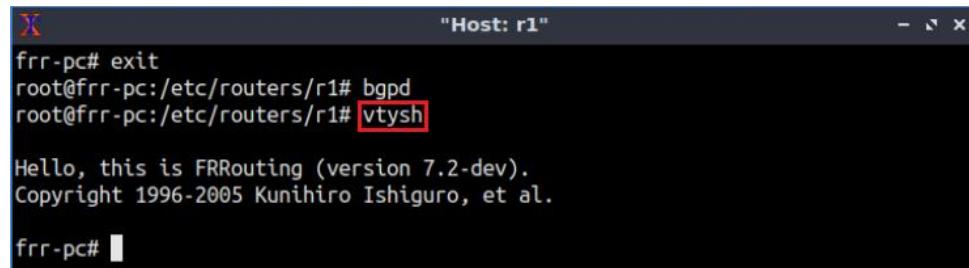


Figure 41. Starting vtysh in router r1.

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

```
configure terminal
```



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

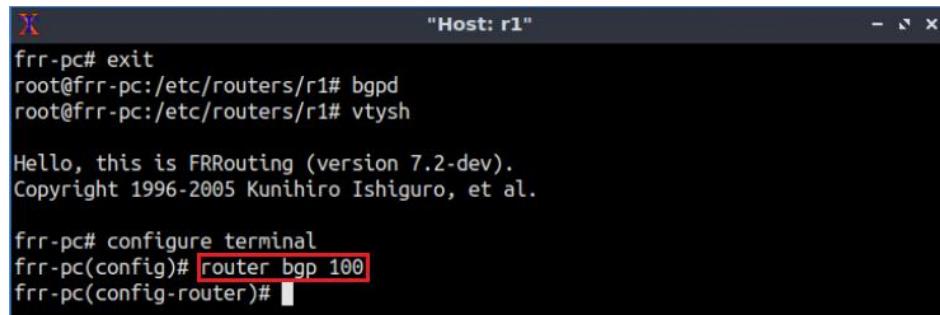
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

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

Figure 42. Enabling configuration mode in router r1.

Step 5. The ASN assigned for router r1 is 100. In order to configure BGP, type the following command:

```
router bgp 100
```



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

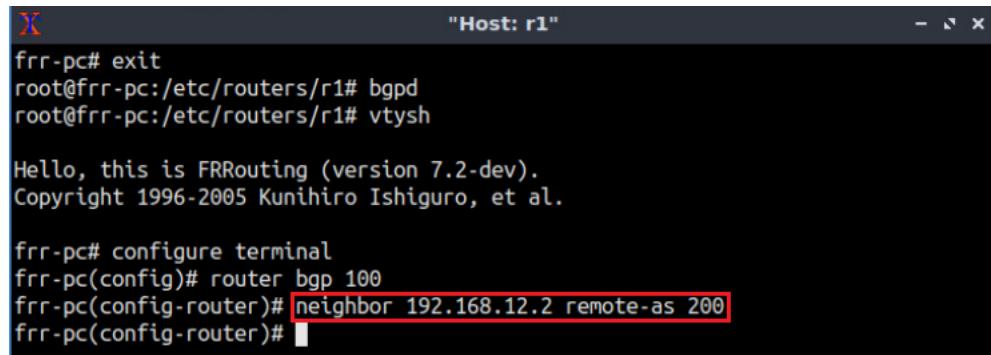
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# router bgp 100
frr-pc(config-router)# 
```

Figure 43. Configuring BGP on router r1.

Step 6. To configure a BGP neighbor to router r1 (AS 100), type the command shown below. This command specifies the neighbor IP address (192.168.12.2) and the ASN of the remote BGP peer (AS 200).

```
neighbor 192.168.12.2 remote-as 200
```



```
"Host: r1"
frr-pc# exit
root@frr-pc:/etc/routers/r1# bgpd
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 bgp 100
frr-pc(config-router)# neighbor 192.168.12.2 remote-as 200
frr-pc(config-router)# 
```

Figure 44. Assigning BGP neighbor to router r1.

Step 7. In this step, router r1 will advertise the LAN 192.168.1.0/24 to its BGP peers. To do so, issue the following command:

```
network 192.168.1.0/24
```

```
frr-pc# exit
root@frr-pc:/etc/routers/r1# bgpd
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 bgp 100
frr-pc(config-router)# neighbor 192.168.12.2 remote-as 200
frr-pc(config-router)# network 192.168.1.0/24
frr-pc(config-router)#

```

Figure 45. Advertising local network in router r1.

Step 8. Type the following command to exit from configuration mode.

end

```
frr-pc# exit
root@frr-pc:/etc/routers/r1# bgpd
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 bgp 100
frr-pc(config-router)# neighbor 192.168.12.2 remote-as 200
frr-pc(config-router)# network 192.168.1.0/24
frr-pc(config-router)# end
frr-pc#

```

Figure 46. Exiting from configuration mode.

Step 9. Type the following command to verify BGP networks. You will observe the LAN network of router r1.

show ip bgp

```
frr-pc# show ip bgp
BGP table version is 1, local router ID is 192.168.12.1, vrf id 0
Default local pref 100, local AS 100
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

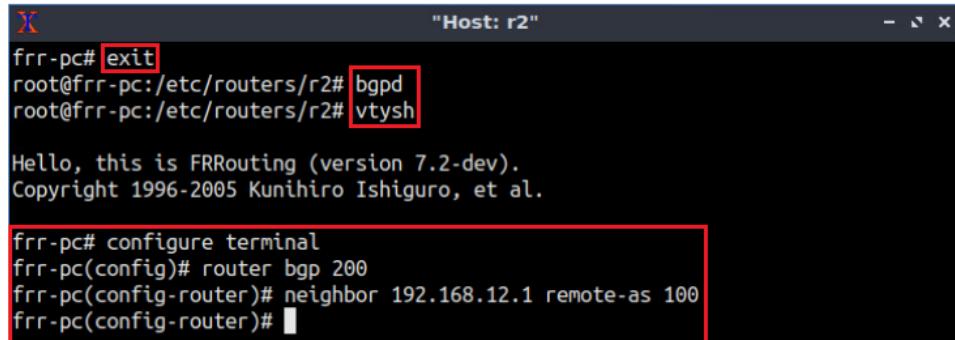
      Network          Next Hop           Metric LocPrf Weight Path
* 192.168.1.0/24    0.0.0.0                  0        32768 i

Displayed 1 routes and 1 total paths
frr-pc#

```

Figure 47. Verifying BGP networks in router r1.

Step 10. Follow from step 1 to step 6 but with different metrics in order to configure BGP on router r2. All the steps are summarized in the following figure.

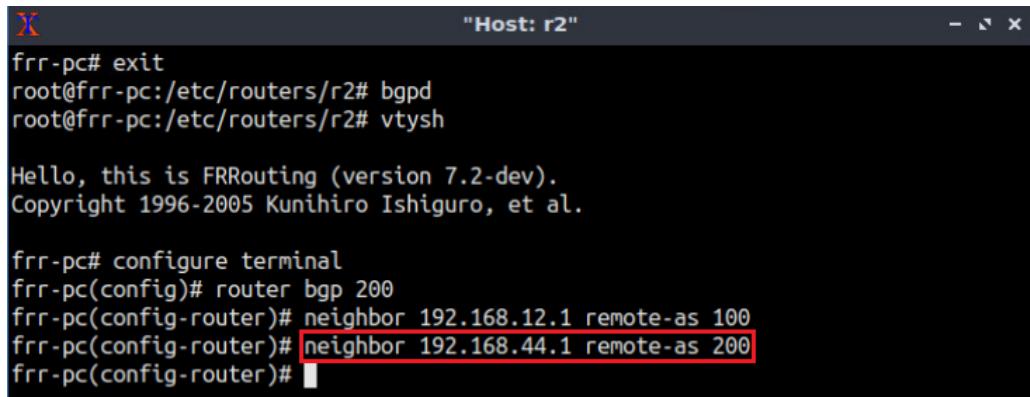


A terminal window titled "Host: r2". The command "frr-pc# exit" is highlighted with a red box. The command "root@frr-pc:/etc/routers/r2# bgpd" is also highlighted with a red box. The command "root@frr-pc:/etc/routers/r2# vtysh" is partially visible at the bottom. Below the prompt, the FRRouting version and copyright information are displayed. A red box highlights the configuration commands:
frr-pc# configure terminal
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.12.1 remote-as 100
frr-pc(config-router)#

Figure 48. Configuring BGP on router r2.

Step 11. Type the following command to assign the loopback address of router r4 (192.168.44.1) as BGP neighbor to router r2. When BGP neighbor routers are within the same AS, they are referred to as IBGP neighbors.

```
neighbor 192.168.44.1 remote-as 200
```

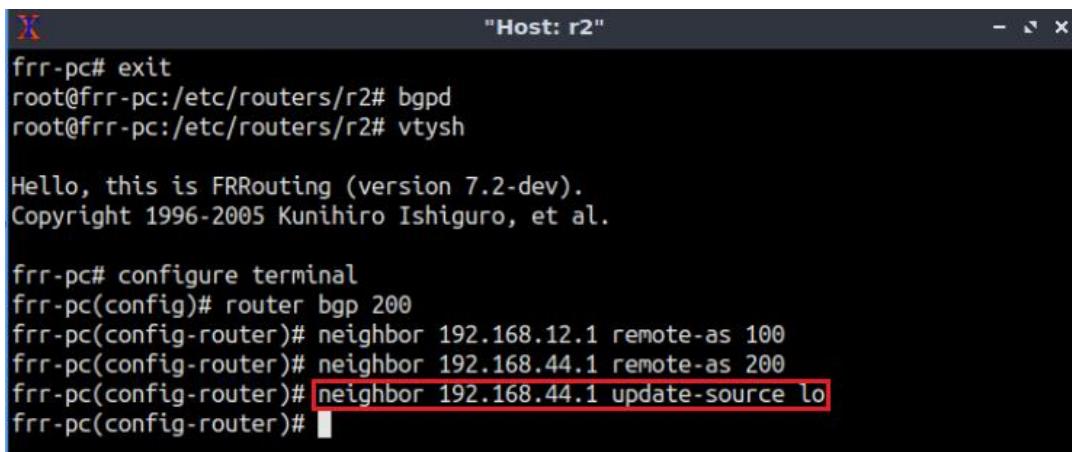


A terminal window titled "Host: r2". The command "frr-pc# exit" is highlighted with a red box. The command "root@frr-pc:/etc/routers/r2# bgpd" is also highlighted with a red box. The command "root@frr-pc:/etc/routers/r2# vtysh" is partially visible at the bottom. Below the prompt, the FRRouting version and copyright information are displayed. A red box highlights the command:
frr-pc# configure terminal
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.12.1 remote-as 100
frr-pc(config-router)# neighbor 192.168.44.1 remote-as 200
frr-pc(config-router)#

Figure 49. Assigning BGP neighbor to router r2.

Step 12. In BGP, the source IP address of BGP packets sent by the router must be the same as the neighbor IP address set on the neighboring router. As you are assigning the loopback as neighbor address, you must use loopback address as the source of BGP packets sent to the neighbor. In router r2, type the following command to assign the loopback address *lo* as the source IP address.

```
neighbor 192.168.44.1 update-source lo
```



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

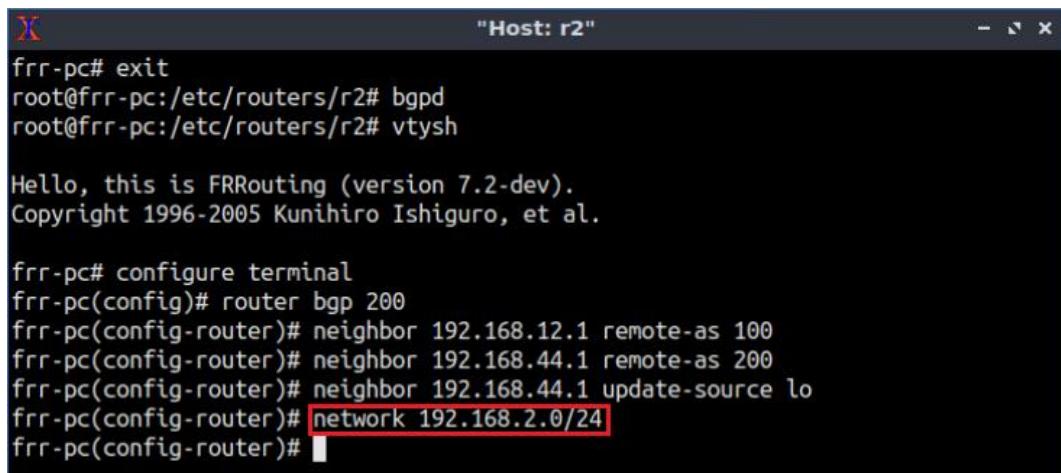
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.12.1 remote-as 100
frr-pc(config-router)# neighbor 192.168.44.1 remote-as 200
frr-pc(config-router)# neighbor 192.168.44.1 update-source lo
frr-pc(config-router)#
```

Figure 50. Assigning loopback address as source IP.

Step 13. In this step, router r2 will advertise the LAN 192.168.2.0/24 to its BGP peers. To do so, issue the following command:

```
network 192.168.2.0/24
```



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

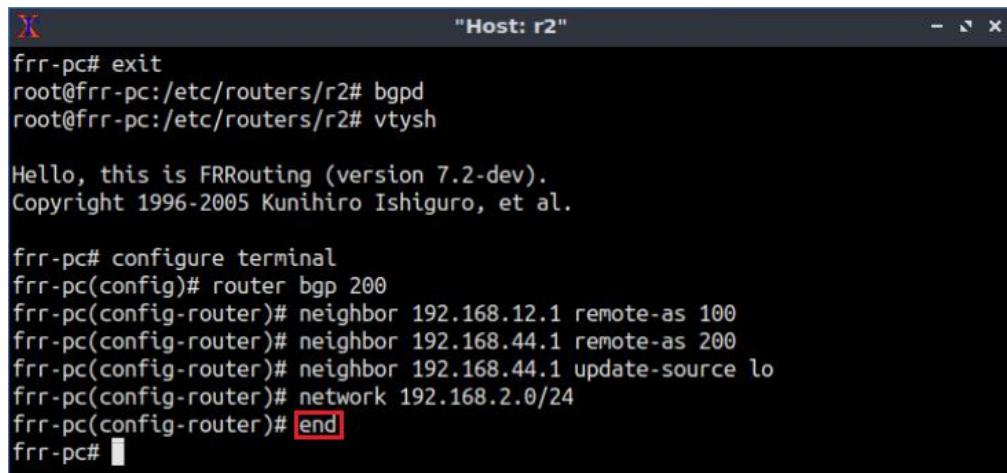
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.12.1 remote-as 100
frr-pc(config-router)# neighbor 192.168.44.1 remote-as 200
frr-pc(config-router)# neighbor 192.168.44.1 update-source lo
frr-pc(config-router)# network 192.168.2.0/24
frr-pc(config-router)#
```

Figure 51. Advertising local network on router r2.

Step 14. Type the following command to exit from the configuration mode.

```
end
```



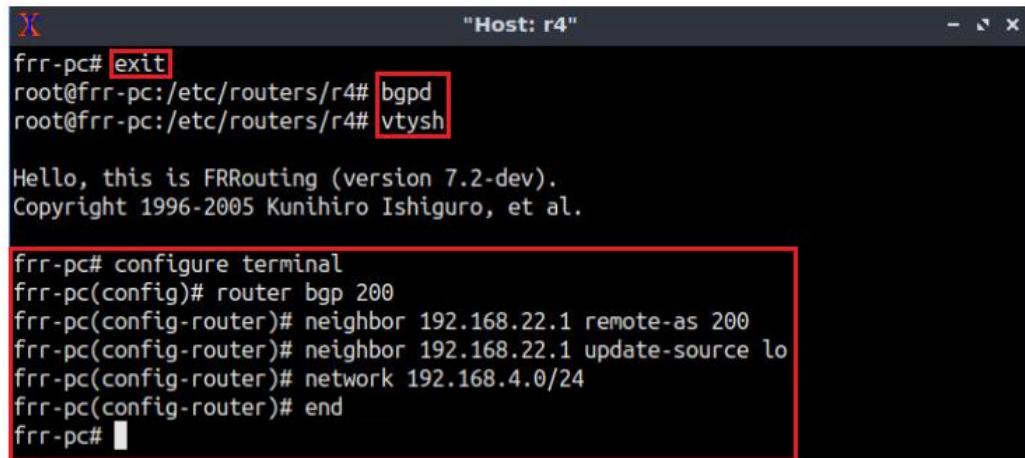
```
frr-pc# exit
root@frr-pc:/etc/routers/r2# bgpd
root@frr-pc:/etc/routers/r2# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.12.1 remote-as 100
frr-pc(config-router)# neighbor 192.168.44.1 remote-as 200
frr-pc(config-router)# neighbor 192.168.44.1 update-source lo
frr-pc(config-router)# network 192.168.2.0/24
frr-pc(config-router)# end
frr-pc#
```

Figure 52. Exiting from configuration mode.

Step 15. Follow from step 10 to step 14 but with different metrics in order to configure BGP on router r4. All the steps are summarized in the following figure.



```
frr-pc# exit
root@frr-pc:/etc/routers/r4# bgpd
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 bgp 200
frr-pc(config-router)# neighbor 192.168.22.1 remote-as 200
frr-pc(config-router)# neighbor 192.168.22.1 update-source lo
frr-pc(config-router)# network 192.168.4.0/24
frr-pc(config-router)# end
frr-pc#
```

Figure 53. Configuring BGP in router r4.

Step 16. In router r2 terminal, type the following command to verify BGP networks. You will observe the LAN networks (192.168.1.0/24, 192.168.2.0/24 and 192.168.4.0/24) participating in the BGP routing process.

```
show ip bgp
```

```
frr-pc# show ip bgp
BGP table version is 3, local router ID is 192.168.22.1, vrf id 0
Default local pref 100, local AS 200
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop           Metric LocPrf Weight Path
*-> 192.168.1.0/24    192.168.12.1        0            0 100 i
*-> 192.168.2.0/24    0.0.0.0            0            32768 i
*>i192.168.4.0/24    192.168.44.1        0         100            0 i

Displayed 3 routes and 3 total paths
frr-pc#
```

Figure 54. Verifying BGP networks on router r2.

Step 17. In router r2 terminal, perform a connectivity test by running the command shown below. To stop the test, press **Ctrl+c**. The result will show a successful connectivity test between router r2 and host h1.

```
ping 192.168.1.10
```

```
frr-pc# ping 192.168.1.10
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_seq=1 ttl=63 time=0.722 ms
64 bytes from 192.168.1.10: icmp_seq=2 ttl=63 time=0.086 ms
64 bytes from 192.168.1.10: icmp_seq=3 ttl=63 time=0.085 ms
^C
--- 192.168.1.10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 47ms
rtt min/avg/max/mdev = 0.085/0.297/0.722/0.300 ms
frr-pc#
```

Figure 55. Connectivity test using **ping** command.

5 Troubleshoot BGP connectivity between routers r1 and r4

In this section, you will verify the connectivity between routers r4 and r1. At this point, the network 192.168.1.0/24 is not listed in the routing table of router r4, however, it's listed in the BGP table. Thus, you will troubleshoot router r4 by changing the next hop of the network 192.168.1.0/24 so that, router r4 receives the route to reach such network.

5.1 Examine and troubleshoot IBGP next hop reachability on router r4

Step 1. Type the following command to verify the routing table of router r4. The routing table does not contain any route to the network 192.168.1.0/24.

```
show ip route
```

```
frrr-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

B    192.168.2.0/24 [200/0] via 192.168.22.1 (recursive), 00:05:38
      via 192.168.34.1, r4-eth1, 00:05:38
0>* 192.168.2.0/24 [110/30] via 192.168.34.1, r4-eth1, 00:27:03
0>* 192.168.3.0/24 [110/20] via 192.168.34.1, r4-eth1, 00:27:03
0    192.168.4.0/24 [110/10] is directly connected, r4-eth0, 00:26:53
C>* 192.168.4.0/24 is directly connected, r4-eth0, 00:32:27
0>* 192.168.22.1/32 [110/20] via 192.168.34.1, r4-eth1, 00:27:03
0>* 192.168.23.0/30 [110/20] via 192.168.34.1, r4-eth1, 00:27:03
0>* 192.168.33.1/32 [110/10] via 192.168.34.1, r4-eth1, 00:27:03
0    192.168.34.0/30 [110/10] is directly connected, r4-eth1, 00:27:13
C>* 192.168.34.0/30 is directly connected, r4-eth1, 00:32:27
0    192.168.44.1/32 [110/0] is directly connected, lo, 00:27:06
C>* 192.168.44.1/32 is directly connected, lo, 00:32:27
frrr-pc#
```

Figure 56. Displaying the routing table of router r4.

Step 2. Type the following command to verify the BGP table of router r4.

```
show ip bgp
```

```
frrr-pc# show ip bgp
BGP table version is 2, local router ID is 192.168.44.1, vrf id 0
Default local pref 100, local AS 200
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
      i192.168.1.0/24  192.168.12.1        0     100      0 100 i
*->i192.168.2.0/24  192.168.22.1        0     100      0 i
*> 192.168.4.0/24   0.0.0.0           0             32768 i

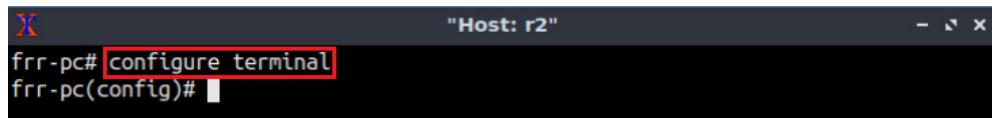
Displayed 3 routes and 3 total paths
frrr-pc#
```

Figure 57. Displaying BGP table of router r4.

Consider Figure 57. The BGP table contains the network 192.168.1.0/24, however, its status code lacks the symbol \star indicating that it is not being offered to the IP routing table. Moreover, the next hop of this network is 192.168.12.1 which is not available in the routing table of router r4. By replacing the next hop IP address by an IP address listed in router r4, will allow router r4 to communicate with router r1.

Step 3. In router r2 terminal, you will configure BGP so that the neighbor address of router r4 uses the loopback address of router r2 (192.168.22.1) as the next hop. To enable router r2 into configuration mode, issue the following command:

```
configure terminal
```

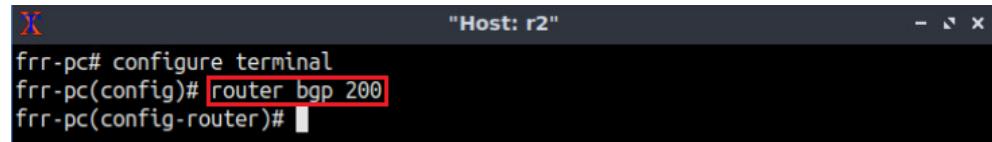


```
"Host: r2"
frr-pc# configure terminal
frr-pc(config)#
```

Figure 58. Enabling configuration mode in router r2.

Step 4. In order to configure BGP, type the following command:

```
router bgp 200
```



```
"Host: r2"
frr-pc# configure terminal
frr-pc(config)# router bgp 200
frr-pc(config-router)#
```

Figure 59. Configuring BGP in router r2.

Step 5. Type the following command to change the next hop address on router r2. Router r2 will use its own loopback address as next hop.

```
neighbor 192.168.44.1 next-hop-self
```



```
"Host: r2"
frr-pc# configure terminal
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.44.1 next-hop-self
frr-pc(config-router)#
```

Figure 60. Changing BGP next hop in router r2.

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



```
"Host: r2"
frr-pc# configure terminal
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.44.1 next-hop-self
frr-pc(config-router)# end
frr-pc#
```

Figure 61. Exiting from configuration mode.

Step 7. Type the following command to verify the BGP table of router r4. You can notice that the next hop of the network 192.168.1.0/24 has been changed to 192.168.22.1.

```
show ip bgp
```

```
frr-pc# show ip bgp
BGP table version is 3, local router ID is 192.168.44.1, vrf id 0
Default local pref 100, local AS 200
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop           Metric LocPrf Weight Path
*->i192.168.1.0/24  192.168.22.1        0      100      0 100 i
*->i192.168.2.0/24  192.168.22.1        0      100      0 i
*> 192.168.4.0/24   0.0.0.0            0          32768 i

Displayed 3 routes and 3 total paths
frr-pc#
```

Figure 62. Displaying BGP table of router r4.

Step 8. Type the following command to verify the routing table of router r4. Router r4 has a route to the network 192.168.1.0/24.

```
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

B> [192.168.1.0/24] [200/0] via [192.168.22.1] (recursive), 00:03:04
  *                               via 192.168.34.1, r4-eth1, 00:03:04
B  192.168.2.0/24 [200/0] via 192.168.22.1 (recursive), 00:13:32
  via 192.168.34.1, r4-eth1, 00:13:32
O>* 192.168.2.0/24 [110/30] via 192.168.34.1, r4-eth1, 00:34:57
O>* 192.168.3.0/24 [110/20] via 192.168.34.1, r4-eth1, 00:34:57
O  192.168.4.0/24 [110/10] is directly connected, r4-eth0, 00:34:47
C>* 192.168.4.0/24 is directly connected, r4-eth0, 00:40:21
O>* 192.168.22.1/32 [110/20] via 192.168.34.1, r4-eth1, 00:34:57
O>* 192.168.23.0/30 [110/20] via 192.168.34.1, r4-eth1, 00:34:57
O>* 192.168.33.1/32 [110/10] via 192.168.34.1, r4-eth1, 00:34:57
O  192.168.34.0/30 [110/10] is directly connected, r4-eth1, 00:35:07
C>* 192.168.34.0/30 is directly connected, r4-eth1, 00:40:21
O  192.168.44.1/32 [110/0] is directly connected, lo, 00:35:00
C>* 192.168.44.1/32 is directly connected, lo, 00:40:21
frr-pc#
```

Figure 63. Displaying the routing table of router r4.

5.2 Troubleshoot Connectivity problem between routers r1 and r4.

At this point, router r4 cannot reach to router r1 even after having the route to reach the network. Router r4 will reach to router r1 via routers r3 and r2. During the configuration, only routers r1, r2, and r4 were configured through BGP. Router r3 only has OSPF routes to routers r2 and r4. Router r3 doesn't have a route to the network 192.168.1.0/24, and thus, when a packet is received from router r4 having the destination 192.168.1.0/24, router r3 will drop it because it does not have a route to this network. In this section, you

will configure BGP on router r3 so that routers r1 and r4 can exchange routes with each other.

Step 1. On host h4 terminal, perform a connectivity test by running the command shown below. To stop the test, press **Ctrl+c**. The results show that host h4 cannot reach host h1.

```
ping 192.168.1.10
```

```
X "Host: h4"
root@frrr-pc:~# ping 192.168.1.10
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.
From 192.168.34.1 icmp_seq=1 Destination Net Unreachable
From 192.168.34.1 icmp_seq=2 Destination Net Unreachable
From 192.168.34.1 icmp_seq=3 Destination Net Unreachable
^C
--- 192.168.1.10 ping statistics ---
3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 49ms

root@frrr-pc:~#
```

Figure 64. Connectivity test using **ping** command.

6 Configure and verify full mesh IBGP

In this section, you will configure and verify fully meshed IBGP among the routers r2, r3, and r4.

6.1 Configure full mesh IBGP on routers r2, r3, and r4.

Step 1. In router r3 terminal, configure IBGP to peer with routers r2 and r4. All the steps are summarized in the following figure.

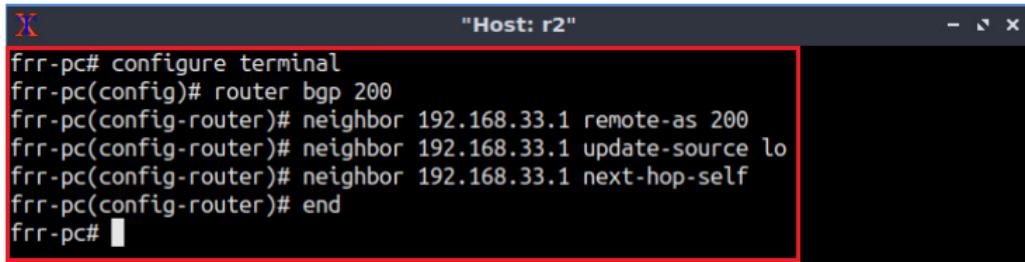
```
X "Host: r3"
frrr-pc# exit
root@frrr-pc:/etc/routers/r3# bgpd
root@frrr-pc:/etc/routers/r3# vtysh

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

frrr-pc# configure terminal
frrr-pc(config)# router bgp 200
frrr-pc(config-router)# neighbor 192.168.22.1 remote-as 200
frrr-pc(config-router)# neighbor 192.168.22.1 update-source lo
frrr-pc(config-router)# neighbor 192.168.44.1 remote-as 200
frrr-pc(config-router)# neighbor 192.168.44.1 update-source lo
frrr-pc(config-router)# network 192.168.3.0/24
frrr-pc(config-router)# end
frrr-pc#
```

Figure 65. Configuring IBGP on router r3.

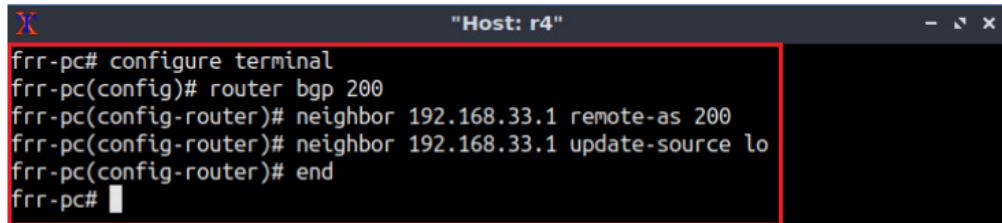
Step 2. In router r2 terminal, configure IBGP so that the neighbor address of router r3 uses the loopback address of router r2 (192.168.22.1) as the next hop so that the next hop address is known to router r3. All the steps are summarized in the following figure.



```
frr-pc# configure terminal
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.33.1 remote-as 200
frr-pc(config-router)# neighbor 192.168.33.1 update-source lo
frr-pc(config-router)# neighbor 192.168.33.1 next-hop-self
frr-pc(config-router)# end
frr-pc#
```

Figure 66. Configuring BGP in router r2.

Step 3. In router r4 terminal, configure BGP to peer with router r3. There is no need to change the next hop when configuring BGP, since r4 is not participating in any EBGP session. All the steps are summarized in the following figure.



```
frr-pc# configure terminal
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.33.1 remote-as 200
frr-pc(config-router)# neighbor 192.168.33.1 update-source lo
frr-pc(config-router)# end
frr-pc#
```

Figure 67. Configuring IBGP in router r4.

6.2 Verify full mesh IBGP on routers r2, r3, and r4.

Step 1. Type the following command to verify the routing table of router r3. You will notice that the routing table contains a route to the network of router r1 (192.168.1.0/24).

```
show ip route
```

```

frr-pc# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       0 - 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

B> [192.168.1.0/24] [200/0] via 192.168.22.1 (recursive), 00:05:13
   *                      via 192.168.23.1, r3-eth1, 00:05:13
B  192.168.2.0/24 [200/0] via 192.168.22.1 (recursive), 00:05:23
   *                      via 192.168.23.1, r3-eth1, 00:05:23
0>* 192.168.2.0/24 [110/20] via 192.168.23.1, r3-eth1, 00:51:03
0  192.168.3.0/24 [110/10] is directly connected, r3-eth0, 00:50:38
C>* 192.168.3.0/24 is directly connected, r3-eth0, 00:55:10
0>* 192.168.4.0/24 [110/20] via 192.168.34.2, r3-eth2, 00:49:22
0>* 192.168.22.1/32 [110/10] via 192.168.23.1, r3-eth1, 00:51:03
0  192.168.23.0/30 [110/10] is directly connected, r3-eth1, 00:51:13
C>* 192.168.23.0/30 is directly connected, r3-eth1, 00:55:10
0  192.168.33.1/32 [110/0] is directly connected, lo, 00:50:54
C>* 192.168.33.1/32 is directly connected, lo, 00:55:10
0  192.168.34.0/30 [110/10] is directly connected, r3-eth2, 00:51:12
C>* 192.168.34.0/30 is directly connected, r3-eth2, 00:55:10
0>* 192.168.44.1/32 [110/10] via 192.168.34.2, r3-eth2, 00:49:35
frr-pc#

```

Figure 68. Displaying the routing table of router r3.

Step 2. In host h4 terminal, perform a connectivity between host h4 and host h1 by issuing the command shown below. To stop the test, press **Ctrl+c**. The result will show a successful connectivity test.

ping 192.168.1.10

```

root@frr-pc:~# ping 192.168.1.10
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_seq=1 ttl=60 time=0.594 ms
64 bytes from 192.168.1.10: icmp_seq=2 ttl=60 time=0.125 ms
64 bytes from 192.168.1.10: icmp_seq=3 ttl=60 time=0.114 ms
^C
--- 192.168.1.10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 53ms
rtt min/avg/max/mdev = 0.114/0.277/0.594/0.224 ms
root@frr-pc:~#

```

Figure 69. Connectivity test using **ping** command.

This concludes Lab 9. Stop the emulation and then exit out of MiniEdit.

References

1. A. Tanenbaum, D. Wetherall, "Computer networks", 5th Edition, Pearson, 2012.
2. Cisco, "What Are OSPF Areas and Virtual Links?", 2016. [Online]. Available: <https://www.cisco.com/c/en/us/support/docs/ip/open-shortest-path-first-ospf/13703-8.html>

3. Cisco, “BGP case studies”, 2008. [Online]. Available:
<https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/26634-bgp-toc.html#bgpnexthop>



BORDER GATEWAY PROTOCOL

Lab 10: BGP Route Reflection

Document Version: **2-27-2020**



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 BGP overview	3
1.2 BGP route reflectors.....	4
2 Lab topology.....	5
2.1 Lab settings.....	6
2.2 Open topology and load configuration	7
2.3 Load zebra daemon and Verify IP addresses	10
3 Configure and verify IBGP	14
3.1 Configure BGP on routers r2, r3 and r4	14
4 Troubleshoot BGP connectivity between routers r2 and r4.....	19
5 Configure and verify EBGP	22
5.1 Configure EBGP on routers r1 and r2.....	22
5.2 Verify the connectivity of the configured topology	26
References	27

Overview

This lab introduces Border Gateway Protocol (BGP) route reflection that offers an alternative to the full mesh Internal BGP (IBGP) topology. In this lab, External BGP (EBGP) will be configured and verified among two Autonomous Systems (ASes). Furthermore, IBGP will be configured within an AS, where a route reflector will distribute IBGP routes to all routers within the AS.

Objectives

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

1. Configure IBGP and EBGP.
2. Understand BGP next hop attribute.
3. Configure BGP route reflectors.
4. Verify the connectivity of the configured topology.

Lab settings

The information in Table 1 provides the credentials to access Client1 machine.

Table 1. Credentials to access Client1 machine.

Device	Account	Password
Client1	admin	password

Lab roadmap

This lab is organized as follows:

1. Section 1: Introduction.
2. Section 2: Lab topology.
3. Section 3: Configure and verify IBGP on routers r2, r3 and r4.
4. Section 4: Troubleshoot BGP connectivity between routers r2 and r4
5. Section 5: Configure and verify EBGP on routers r1 and r2.
6. Section 6: Configure and verify full mesh IBGP.

1 Introduction

1.1 BGP overview

BGP is an exterior gateway protocol designed to exchange routing and reachability information among ASes on the Internet. BGP is relevant to network administrators of large organizations which connect to one or more Internet Service Providers (ISPs), as well as to ISPs who connect to other network providers. In terms of BGP, an AS is referred to as a routing domain, where all networked systems operate common routing protocols and are under the control of a single administration¹.

BGP is a form of distance vector protocol. It requires each router to maintain a table, which stores the distance and the output interface (i.e., vector) to remote networks. BGP makes routing decisions based on paths, network policies, or rule set configured by a network administrator and is involved in making core routing decisions¹.

Two routers that establish a BGP connection are referred to as BGP peers or neighbors. BGP sessions run over Transmission Control Protocol (TCP). If a BGP session is established between two neighbors in different ASes, the session is referred to as an EBGP session. If the session is established between two neighbors in the same AS, the session is referred to as IBGP¹. Figure 1 shows a network running BGP protocol. Routers that exchange information within the same AS use IBGP, while routers that exchange information between different ASes use EBGP.



Figure 1. Routers that exchange information within the same AS use IBGP, while routers that exchange information between different ASes use EBGP.

1.2 BGP route reflectors

IBGP routers do not re-advertise routes learned via IBGP to their peers. This behavior is used to prevent routing information cycles, i.e., the circulation of routing information between IBGP peers in a continuous repeated manner². Consider Figure 2, Router r1 sends a route advertisement to its IBGP neighbor (router r2). Router r2 will not advertise the learned route to its IBGP neighbors (router r3) since it was learned from an IBGP peer (router r1).

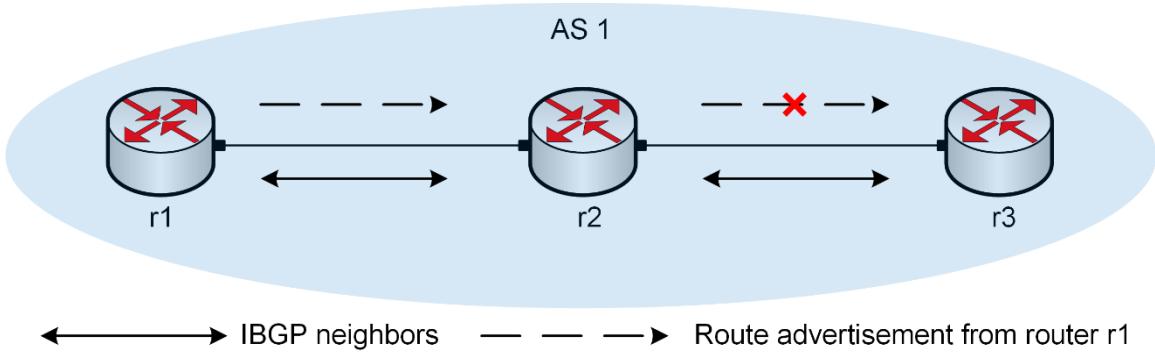


Figure 2. In BGP, routers do not advertise routes that are learned via IBGP to their IBGP neighbors.

A topology is called full mesh (fully meshed topology) when there is an IBGP peering relationship between any two routers in the AS. All BGP routers within a single AS must be fully meshed so that any external routing information must be re-distributed to all other routers within that AS. However, when the network topology of the AS is large, it becomes challenging to configure a full mesh IBGP².

A BGP route reflector is an IBGP router that repeats routes learned from IBGP peers to some of its other IBGP peers². Consider Figure 3, instead of having a full mesh IBGP topology, route reflection will be configured. The route reflector (router r2) advertises the route learned from its IBGP neighbors to all its clients. Thus, router r3 receives the advertised routes from router r1, and the way around, without establishing IBGP neighbor relationship.

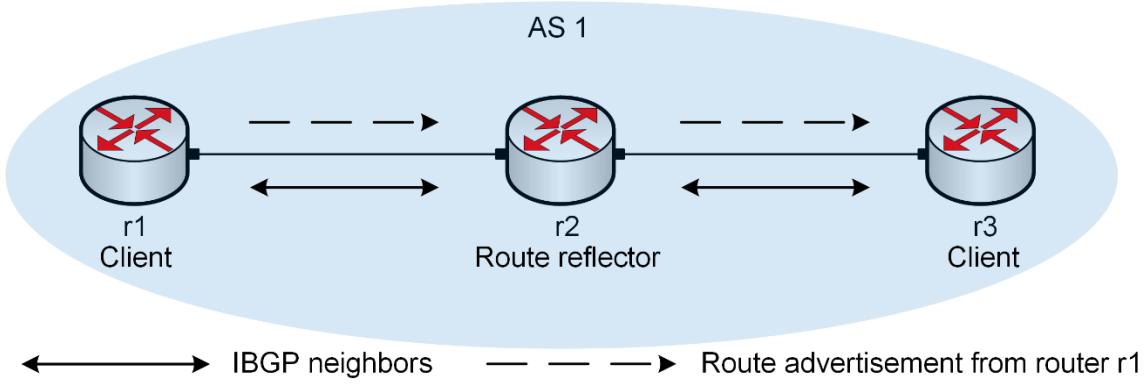


Figure 3. Route reflectors advertise IBGP routes to their IBGP peers.

2 Lab topology

Consider Figure 4. The lab topology consists of two ASes, each identified by an Autonomous System Number (ASN). The ISP, i.e., router r1, provides Internet service to the Campus network (routers r2, r3 and r4). The ASN assigned to the ISP and the Campus network are 100 and 200, respectively. The ISP communicates with the Campus via EBGP routing protocol, and the routers within the Campus network communicate using IBGP, where router r3 is a route reflector, and routers r1 and r2 are the clients.

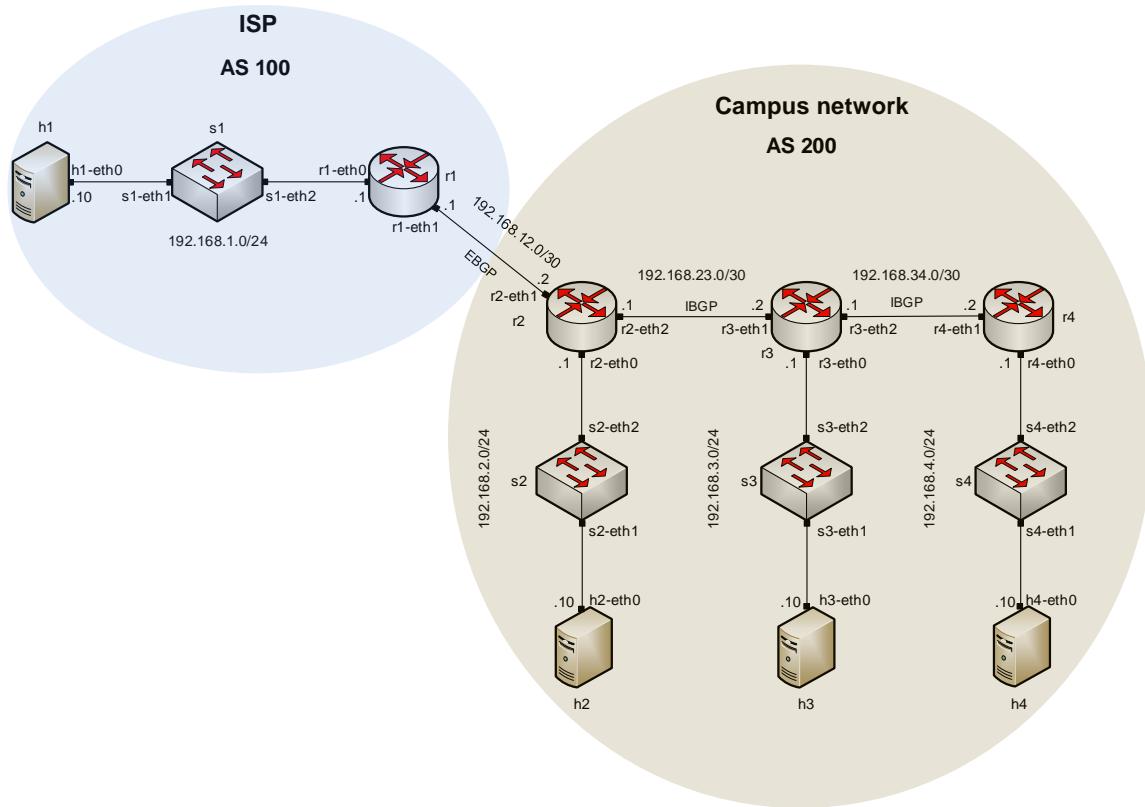


Figure 4. Lab topology.

2.1 Lab settings

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

Table 2. Topology information.

Device	Interface	IPV4 Address	Subnet	Default gateway
r1 (ISP)	r1-eth0	192.168.1.1	/24	N/A
	r1-eth1	192.168.12.1	/30	N/A
r2 (Campus network)	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
r3 (Campus network)	r3-eth0	192.168.3.1	/24	N/A
	r3-eth1	192.168.23.2	/30	N/A
	r3-eth2	192.168.34.1	/30	N/A
r4 (Campus network)	r4-eth0	192.168.4.1	/24	N/A
	r4-eth1	192.168.34.2	/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
h3	h3-eth0	192.168.3.10	/24	192.168.3.1
h4	h4-eth0	192.168.4.10	/24	192.168.4.1

2.2 Open topology and load configuration

Step 1. Start by launching Miniedit by clicking on Desktop's shortcut. When prompted for a password, type `password`.



Figure 5. MiniEdit shortcut.

Step 2. On Miniedit's menu bar, click on *File* then *open* to load the lab's topology. Locate the *Lab10.mn* topology file in the default directory, */home/frr/BGP_Labs/lab10* and click on *Open*.

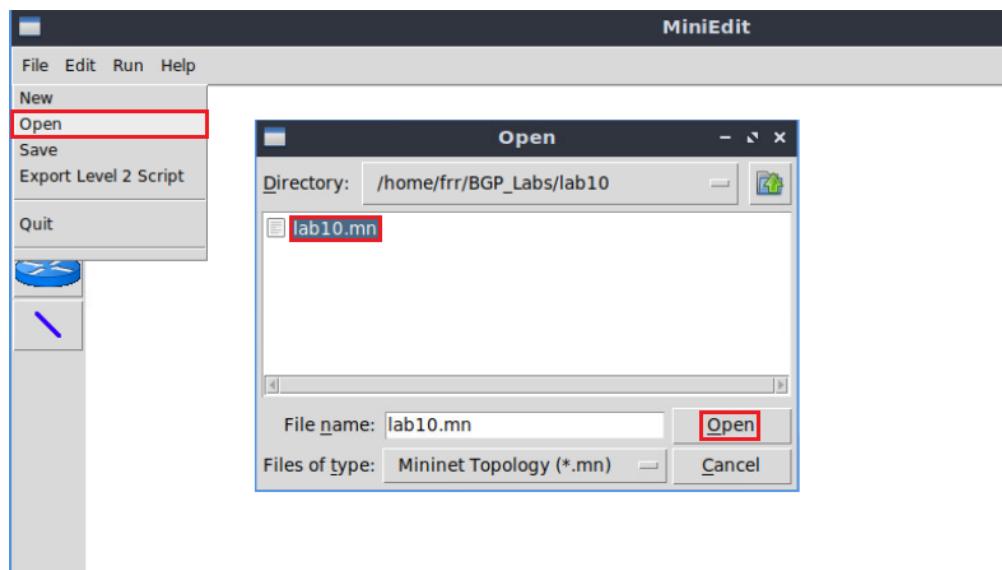


Figure 6. MiniEdit's open dialog.

At this point the topology is loaded with all the required network components. You will execute a script that will load the configuration of the routers.

Step 3. Open the Linux terminal.

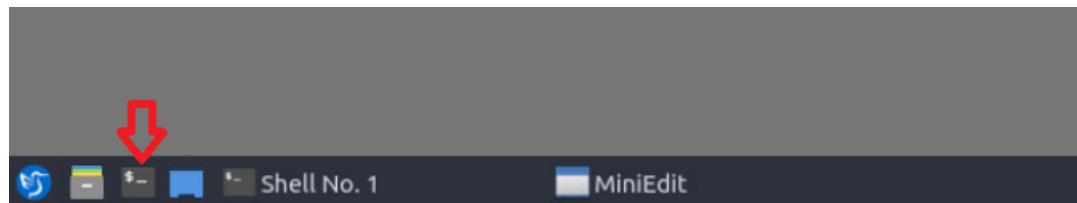


Figure 7. Opening Linux terminal

Step 4. Click on the Linux's terminal and navigate into *BGP_Labs/lab10* 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 BGP_Labs/lab10
```

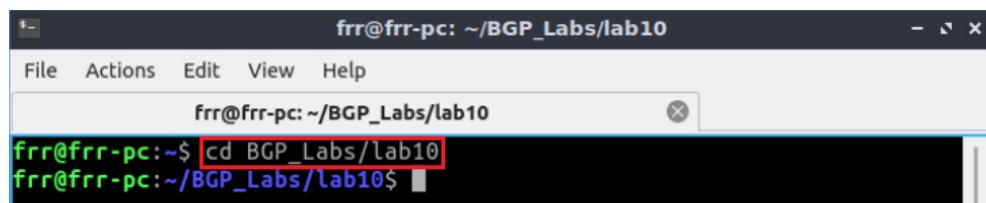


Figure 8. Entering the *BGP_Labs/lab10* directory.

Step 5. To execute the shell script, type the following command. The argument of the program corresponds to the configuration zip file that will be loaded in all the routers in the topology.

```
./config_loader.sh lab10_conf.zip
```

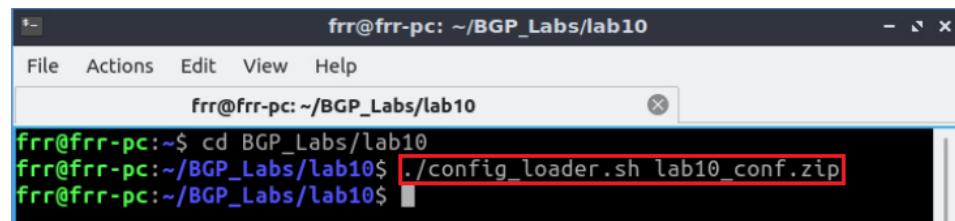
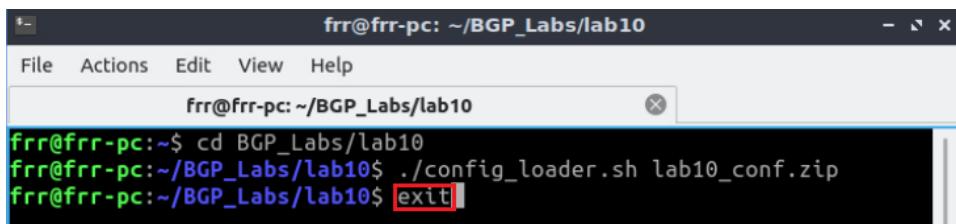


Figure 9. Executing the shell script to load the configuration.

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

```
exit
```



```
frr@frr-pc: ~/BGP_Labs/lab10
File Actions Edit View Help
frr@frr-pc: ~/BGP_Labs/lab10
frr@frr-pc:~/BGP_Labs/lab10$ ./config_loader.sh lab10_conf.zip
frr@frr-pc:~/BGP_Labs/lab10$ exit
```

Figure 10. Exiting from the terminal.

Step 7. At this point hosts h1, h2, h3 and h4 interfaces are configured. To proceed with the emulation, click on the *Run* button located in lower left-hand side.



Figure 11. Starting the emulation.

Step 8. Click on Mininet's terminal, i.e., the one launched when MiniEdit was started.

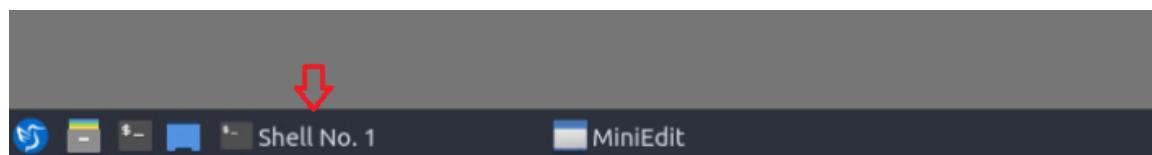
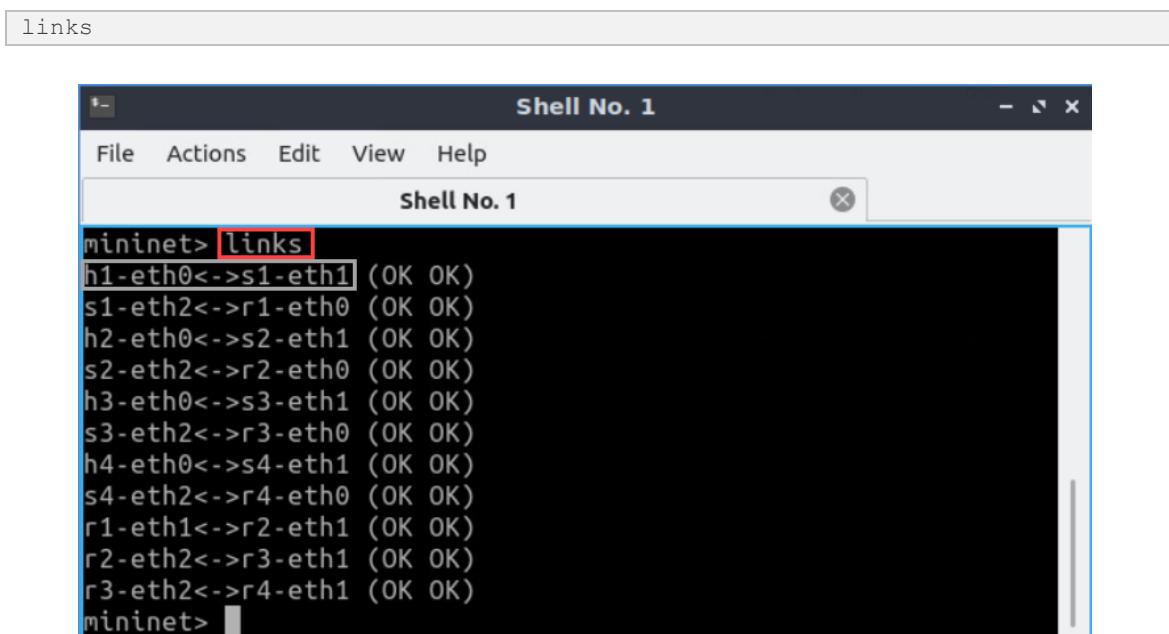


Figure 12. Opening Mininet's terminal.

Step 9. Issue the following command to display the interface names and connections.



```
links
Shell No. 1
File Actions Edit View Help
Shell No. 1
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)
r3-eth2<->r4-eth1 (OK OK)
mininet>
```

Figure 13. Displaying network interfaces.

In Figure 13, the link displayed within the gray box indicates that interface *eth0* of host h1 connects to interface *eth1* of switch s1 (i.e., $h1\text{-}eth0 <-> s1\text{-}eth1$).

2.3 Load zebra daemon and Verify IP addresses

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

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.

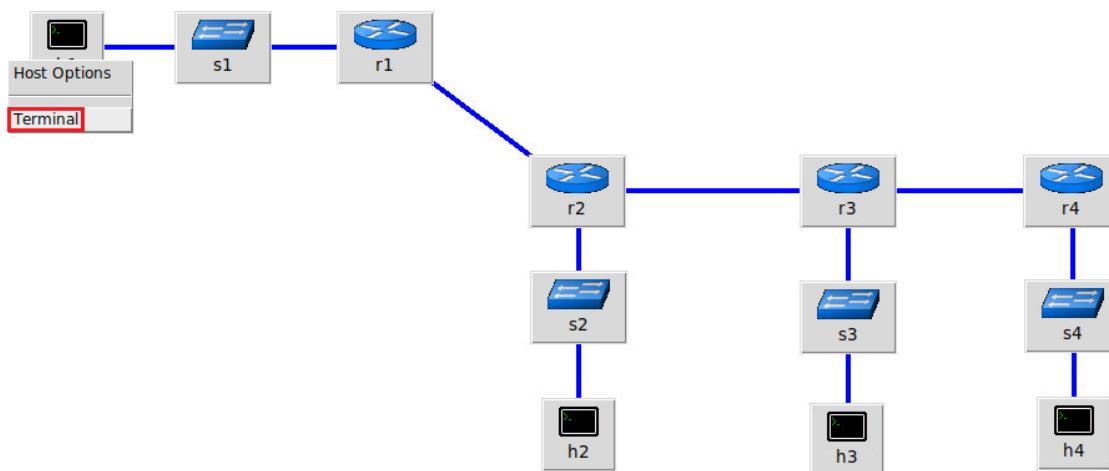


Figure 14. Opening terminal on host h1.

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 an interface, *h1-eth0* 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 15. 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:~# 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. Output of `route` command.

Step 4. In order to verify hosts h2, h3 and h4, proceed similarly by repeating from step 1 to step 3 on host h2, h3 and h4 terminals. Similar results should be observed.

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

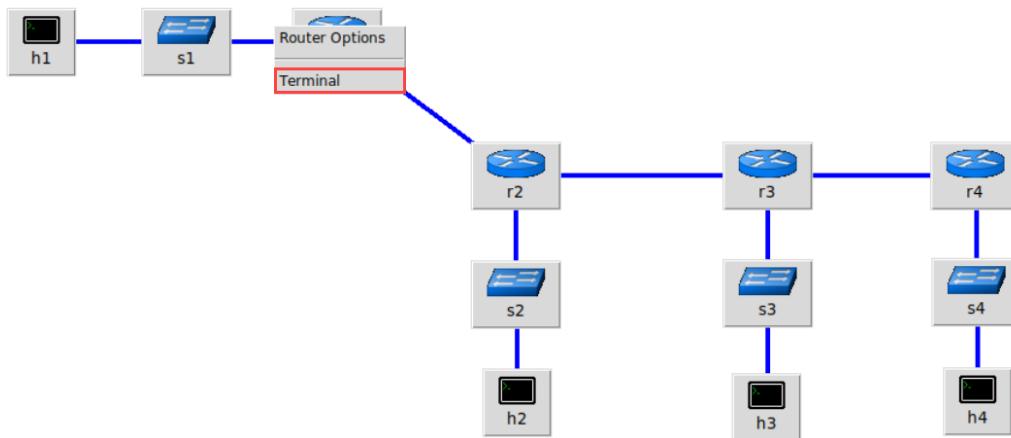


Figure 17. Opening terminal on router r1.

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

```
zebra
```

```

X "Host: r1"
root@frrr-pc:/etc/routers/r1# zebra
root@frrr-pc:/etc/routers/r1#
  
```

Figure 18. Starting zebra daemon.

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

```
vtysh
```

```

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

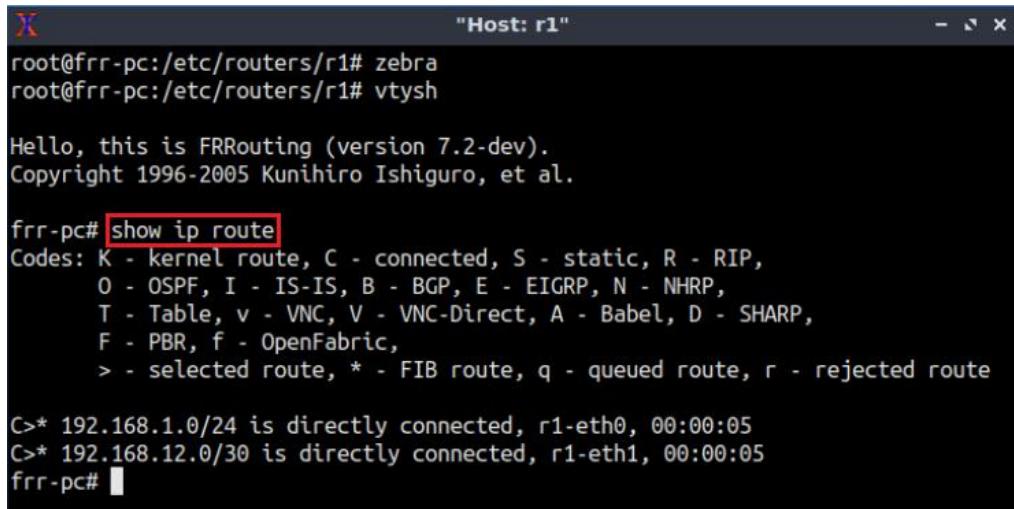
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frrr-pc# 
  
```

Figure 19. 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 networks attached to routers r2 (192.168.2.0/24), r3 (192.168.3.0/24) and r4 (192.168.4.0/24) 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

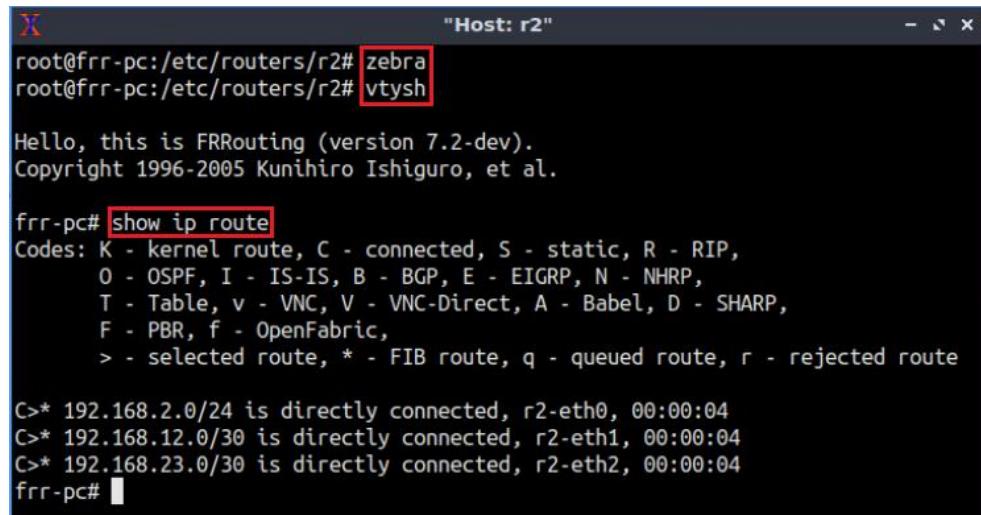
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

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, 00:00:05
C>* 192.168.12.0/30 is directly connected, r1-eth1, 00:00:05
frr-pc# "
```

Figure 20. Displaying the routing table of router r1.

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, 00:00:04
C>* 192.168.12.0/30 is directly connected, r2-eth1, 00:00:04
C>* 192.168.23.0/30 is directly connected, r2-eth2, 00:00:04
frr-pc# "
```

Figure 21. Displaying the routing table of router r2.

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

```

root@frr-pc:/etc/routers/r3# zebra
root@frr-pc:/etc/routers/r3# 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>* 192.168.3.0/24 is directly connected, r3-eth0, 00:00:06
C>* 192.168.23.0/30 is directly connected, r3-eth1, 00:00:06
C>* 192.168.34.0/30 is directly connected, r3-eth2, 00:00:06
frr-pc#
  
```

Figure 22. Displaying the routing table of router r3.

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

```

root@frr-pc:/etc/routers/r4# zebra
root@frr-pc:/etc/routers/r4# 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>* 192.168.4.0/24 is directly connected, r4-eth0, 00:00:06
C>* 192.168.34.0/30 is directly connected, r4-eth1, 00:00:06
frr-pc#
  
```

Figure 23. Displaying the routing table of router r4.

3 Configure and verify IBGP

3.1 Configure BGP on routers r2, r3 and r4

In this section, you will configure IBGP on routers r2, r3 and r4. IBGP peering relationship will occur between routers r2 and r3, as well as routers r3 and r4. Additionally, routers r2, r3, and r4 will advertise the networks 192.168.2.0/24, 192.168.3.0/24, and 192.168.4.0/24, respectively.

Step 1. To configure BGP routing protocol, you need to enable the BGP daemon first. In router r2 terminal, type the following command to exit the vtysh session:

```
exit
```

The terminal window shows the command 'exit' being typed at the prompt 'frr-pc#'. The window title is 'Host: r2'.

Figure 24. Exiting the vtysh session.

Step 2. Type the following command on r2 terminal to enable and start BGP routing protocol.

```
bgpd
```

The terminal window shows the command 'bgpd' being typed at the prompt 'root@frr-pc:/etc/routers/r2#'. The window title is 'Host: r2'.

Figure 25. Starting BGP daemon.

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

```
vtysh
```

The terminal window shows the command 'vtysh' being typed at the prompt 'root@frr-pc:/etc/routers/r2#'. The window title is 'Host: r2'. Below the prompt, the FRRouting version information is displayed: 'Hello, this is FRRouting (version 7.2-dev). Copyright 1996-2005 Kunihiro Ishiguro, et al.' The prompt then changes to 'frr-pc#'. The command 'bgpd' was run in the previous step.

Figure 26. Starting vtysh in router r2.

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

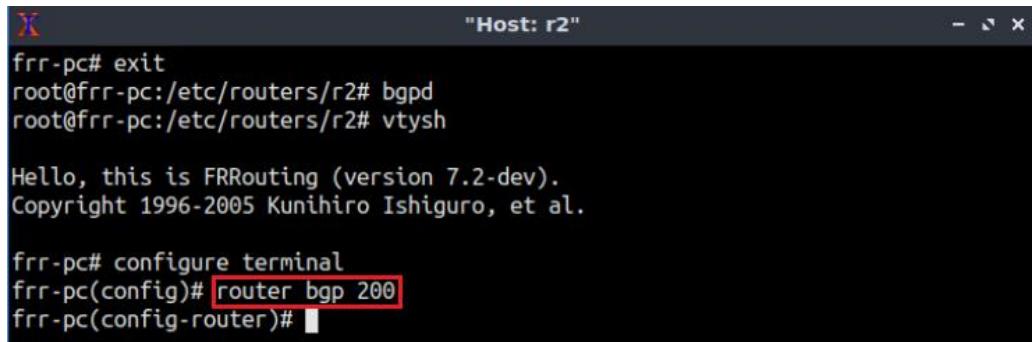
```
configure terminal
```

The terminal window shows the command 'configure terminal' being typed at the prompt 'frr-pc#'. The window title is 'Host: r2'. The command was preceded by 'vtysh' and 'bgpd' in the previous steps. The prompt then changes to 'frr-pc(config)#'. The command 'bgpd' was run in the previous step.

Figure 27. Enabling configuration mode in router r2.

Step 5. The ASN assigned for router r2 is 200. In order to configure BGP, type the following command:

```
router bgp 200
```



The terminal window shows the following session:

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

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

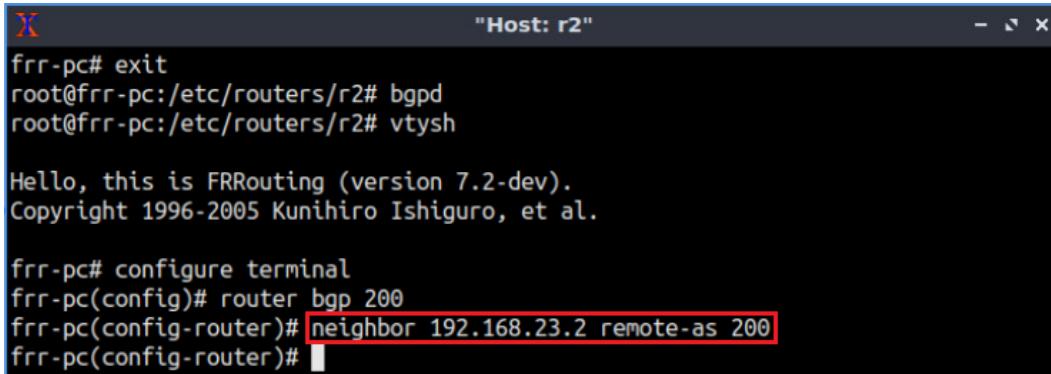
frr-pc# configure terminal
frr-pc(config)# router bgp 200
frr-pc(config-router)#

```

Figure 28. Configuring BGP on router r2.

Step 6. To configure a BGP neighbor to router r2 (AS 200), type the command shown below. This command specifies the neighbor IP address (192.168.23.2) and the ASN of the remote BGP peer (AS 200).

```
neighbor 192.168.23.2 remote-as 200
```



The terminal window shows the following session:

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

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

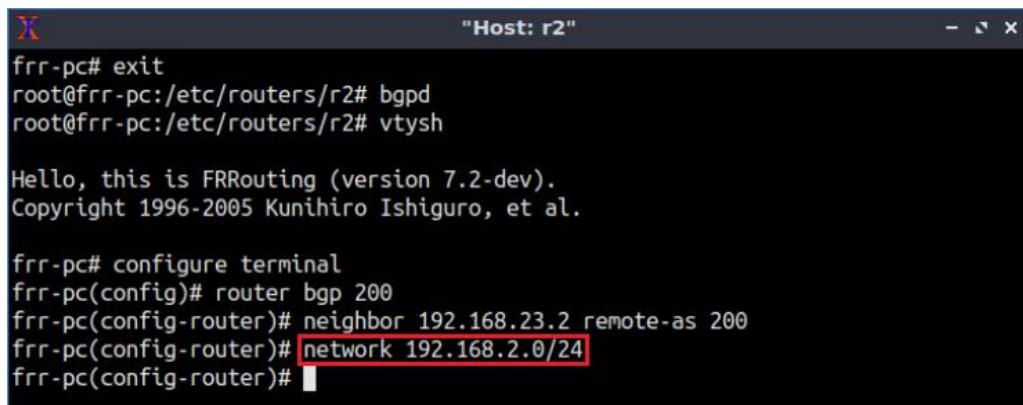
frr-pc# configure terminal
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.23.2 remote-as 200
frr-pc(config-router)#

```

Figure 29. Assigning BGP neighbor to router r2.

Step 7. In this step, router r2 will advertise the Local Area Network (LAN) 192.168.2.0/24 to its BGP peers. To do so, issue the following command:

```
network 192.168.2.0/24
```



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

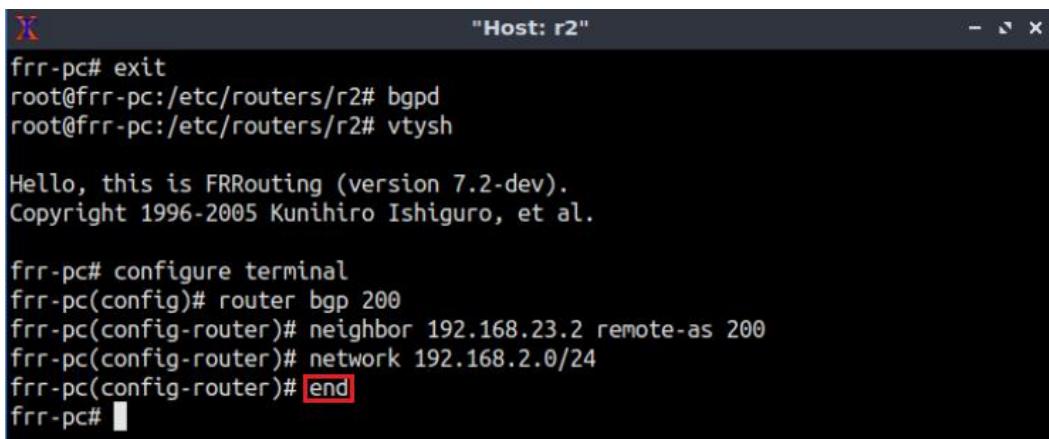
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.23.2 remote-as 200
frr-pc(config-router)# network 192.168.2.0/24
frr-pc(config-router)#
```

Figure 30. Advertising local network in router r2.

Step 8. Type the following command to exit from configuration mode.

```
end
```



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

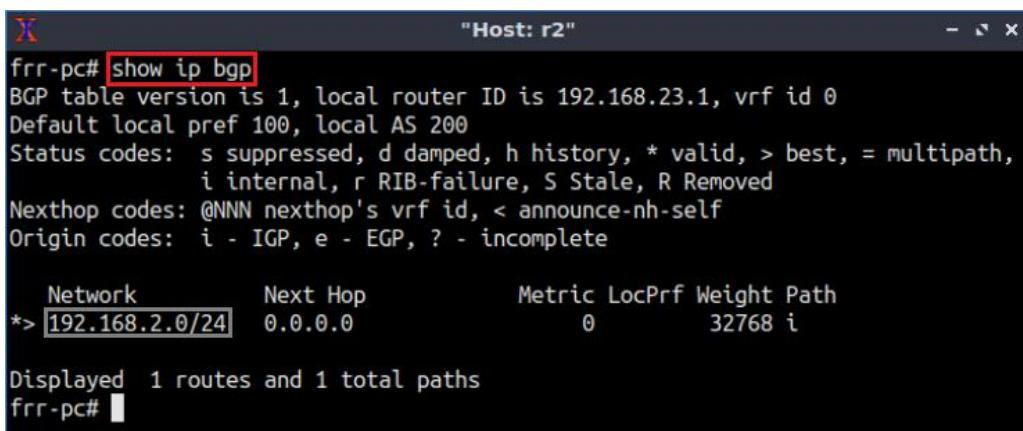
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.23.2 remote-as 200
frr-pc(config-router)# network 192.168.2.0/24
frr-pc(config-router)# end
frr-pc#
```

Figure 31. Exiting from configuration mode.

Step 9. Type the following command to verify BGP networks. Verify the LAN network of router r2.

```
show ip bgp
```



```
frr-pc# show ip bgp
BGP table version is 1, local router ID is 192.168.23.1, vrf id 0
Default local pref 100, local AS 200
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
              i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop           Metric LocPrf Weight Path
*-> 192.168.2.0/24    0.0.0.0                  0        32768 i

Displayed 1 routes and 1 total paths
frr-pc#
```

Figure 32. Verifying BGP networks in router r2.

Step 10. Follow from step 1 to step 8 but with different metrics in order to configure BGP on router r3. All the steps are summarized in the following figure.

The terminal window shows the following session:

```
frr-pc# exit  
root@frr-pc:/etc/routers/r3# bgpd  
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 bgp 200  
frr-pc(config-router)# neighbor 192.168.23.1 remote-as 200  
frr-pc(config-router)# neighbor 192.168.34.2 remote-as 200  
frr-pc(config-router)# network 192.168.3.0/24  
frr-pc(config-router)# end  
frr-pc#
```

Figure 33. Configuring BGP on router r3.

Step 11. Follow from step 1 to step 8 but with different metrics in order to configure BGP on router r4. All the steps are summarized in the following figure.

The terminal window shows the following session:

```
frr-pc# exit  
root@frr-pc:/etc/routers/r4# bgpd  
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 bgp 200  
frr-pc(config-router)# neighbor 192.168.34.1 remote-as 200  
frr-pc(config-router)# network 192.168.4.0/24  
frr-pc(config-router)# end  
frr-pc#
```

Figure 34. Configuring BGP in router r4.

Step 12. In router r2 terminal, type the following command to verify BGP networks. The network of router r3 (192.168.3.0/24) has been added to the BGP table.

```
show ip bgp
```

```
frr-pc# show ip bgp
BGP table version is 2, local router ID is 192.168.23.1, vrf id 0
Default local pref 100, local AS 200
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Next-hop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop           Metric LocPrf Weight Path
*-> 192.168.2.0/24    0.0.0.0                  0        32768 i
*>i192.168.3.0/24    192.168.23.2            0       100      0 i

Displayed 2 routes and 2 total paths
frr-pc#
```

Figure 35. Verifying BGP networks on router r2.

4 Troubleshoot BGP connectivity between routers r2 and r4

In this section, you will verify the connectivity between routers r2 and r4.

Step 1. Type the following command to verify the routing table of router r4. The routing table does not contain any route to the network 192.168.2.0/24.

```
show ip route
```

```
frr-pc# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       0 - 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

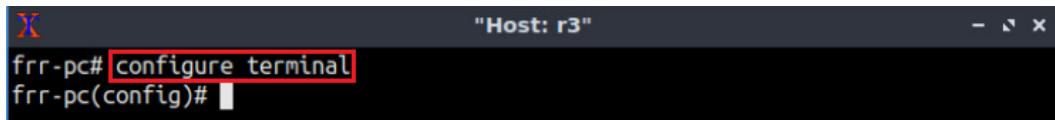
B>* 192.168.3.0/24 [200/0] via 192.168.34.1, r4-eth1, 00:06:03
C>* 192.168.4.0/24 is directly connected, r4-eth0, 00:39:03
C>* 192.168.34.0/30 is directly connected, r4-eth1, 00:39:03
frr-pc#
```

Figure 36. Displaying the routing table of router r4.

Consider Figure 36. The routing table of router r4 does not contain any route to the network 192.168.2.0/24. This result is expected since router r4 does not have a BGP peering relationship with router r2. Furthermore, router r3 will not advertise the network 192.168.2.0/24 to its IBGP neighbor router r4 since this route is learned via IBGP (router r2). To solve this issue, router r3 will be configured as a route reflector so that it advertises IBGP learned routes to IBGP neighbors.

Step 2. To enable router r3 into configuration mode, issue the following command:

```
configure terminal
```



```
"Host: r3"
frr-pc# configure terminal
frr-pc(config)#
```

Figure 37. Enabling configuration mode in router r3.

Step 3. In order to configure BGP, type the following command:

```
router bgp 200
```

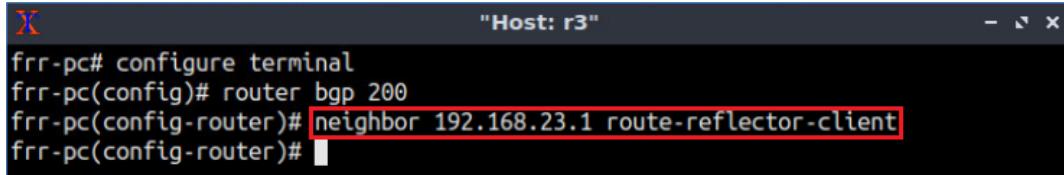


```
"Host: r3"
frr-pc# configure terminal
frr-pc(config)# router bgp 200
frr-pc(config-router)#
```

Figure 38. Configuring BGP on router r3.

Step 4. In order to configure router r3 so that it advertises IBGP learned routes to its IBGP neighbor router r2 (192.168.23.1), type the following command:

```
neighbor 192.168.23.1 route-reflector-client
```

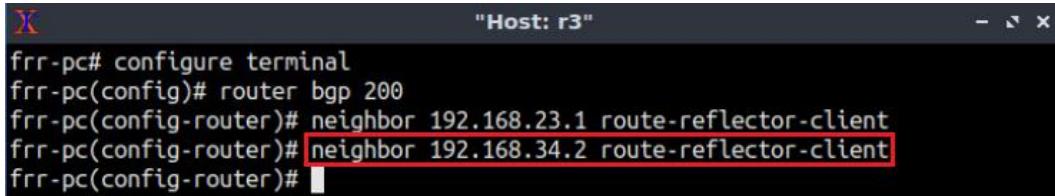


```
"Host: r3"
frr-pc# configure terminal
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.23.1 route-reflector-client
frr-pc(config-router)#
```

Figure 39. Configuring a client peer to the route reflector router r3.

Step 5. In order to configure router r3 so that it advertises IBGP learned routes to its IBGP neighbor router r4 (192.168.34.2), type the following command:

```
neighbor 192.168.34.2 route-reflector-client
```



```
"Host: r3"
frr-pc# configure terminal
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.23.1 route-reflector-client
frr-pc(config-router)# neighbor 192.168.34.2 route-reflector-client
frr-pc(config-router)#
```

Figure 40. Configuring client peer to the route reflector router r3.

At this point, router r2 receives route advertisements from router r4, and router r4 receives route advertisements from router r2, all via router r3. However, router r4 does not have a route to the next hop IP address 192.168.23.1 (router r2), and router r2 does not have a route to the next hop IP address 192.168.34.2 (router r4). Thus, hindering the routing process between routers r2 and r4. To solve this issue, router r3 will be configured to advertise the networks 192.168.23.0/30 and 192.168.34.0/30.

Step 6. In router r3 terminal, type the following command to advertise the network 192.168.23.0/30:

```
network 192.168.23.0/30
```

```
frr-pc# configure terminal
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.23.1 route-reflector-client
frr-pc(config-router)# neighbor 192.168.34.2 route-reflector-client
frr-pc(config-router)# network 192.168.23.0/30
frr-pc(config-router)#

```

Figure 41. Displaying the routing table of router r3.

Step 7. In router r3 terminal, type the following command to advertise the network 192.168.34.0/30:

```
network 192.168.34.0/30
```

```
frr-pc# configure terminal
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.23.1 route-reflector-client
frr-pc(config-router)# neighbor 192.168.34.2 route-reflector-client
frr-pc(config-router)# network 192.168.23.0/30
frr-pc(config-router)# network 192.168.34.0/30
frr-pc(config-router)#

```

Figure 42. Displaying the routing table of router r3.

Step 8. Type the following command to exit from configuration mode.

```
end
```

```
frr-pc# configure terminal
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.23.1 route-reflector-client
frr-pc(config-router)# neighbor 192.168.34.2 route-reflector-client
frr-pc(config-router)# network 192.168.23.0/30
frr-pc(config-router)# network 192.168.34.0/30
frr-pc(config-router)# end
frr-pc#

```

Figure 43. Exiting from configuration mode.

Step 9. Type the following command to verify the routing table of router r4.

```
show ip route
```

```
frrr-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

B> [192.168.2.0/24] [200/0] via 192.168.23.1 (recursive), 00:03:03
   *                               via 192.168.34.1, r4-eth1, 00:03:03
B>* 192.168.3.0/24 [200/0] via 192.168.34.1, r4-eth1, 00:04:31
C>* 192.168.4.0/24 is directly connected, r4-eth0, 04:18:13
B>* 192.168.23.0/30 [200/0] via 192.168.34.1, r4-eth1, 00:03:03
B  192.168.34.0/30 [200/0] via 192.168.34.1 inactive, 00:01:58
C>* 192.168.34.0/30 is directly connected, r4-eth1, 04:18:13
frrr-pc#
```

Figure 44. Displaying the routing table of router r4.

Consider Figure 44. The routing table of router r4 has a route to the network 192.168.2.0/24. This route was advertised by the route reflector router r3.

Step 10. In host h4 terminal, perform a connectivity test between host h4 and host h2 by issuing the command shown below. To stop the test, press **Ctrl+c**. The result will show a successful connectivity test.

```
ping 192.168.2.10
```

```
root@frrr-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=61 time=0.600 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=0.105 ms
64 bytes from 192.168.2.10: icmp_seq=4 ttl=61 time=0.107 ms
^C
--- 192.168.2.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 71ms
rtt min/avg/max/mdev = 0.098/0.227/0.600/0.215 ms
root@frrr-pc:~#
```

Figure 45. Connectivity test using **ping** command.

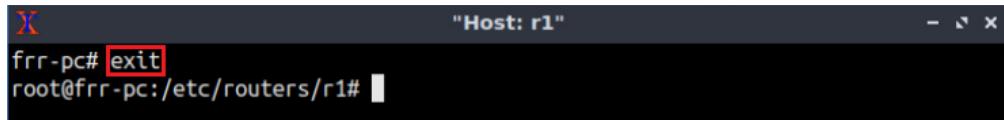
5 Configure and verify EBGP

In this section, you will configure and verify EBGP on routers r1 and r2.

5.1 Configure EBGP on routers r1 and r2

Step 1. To configure BGP routing protocol, you need to enable the BGP daemon first. In router r1 terminal, type the following command to exit the vtysh session:

```
exit
```

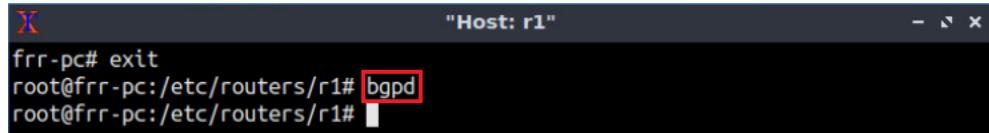


```
"Host: r1"
frr-pc# exit
root@frr-pc:/etc/routers/r1#
```

Figure 46. Exiting the vtysh session.

Step 2. Type the following command on r1 terminal to enable and start BGP routing protocol.

```
bgpd
```

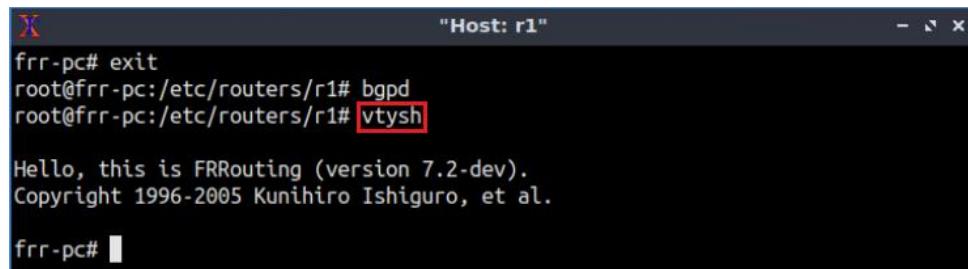


```
"Host: r1"
frr-pc# exit
root@frr-pc:/etc/routers/r1# bgpd
root@frr-pc:/etc/routers/r1#
```

Figure 47. Starting BGP daemon.

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

```
vtysh
```



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

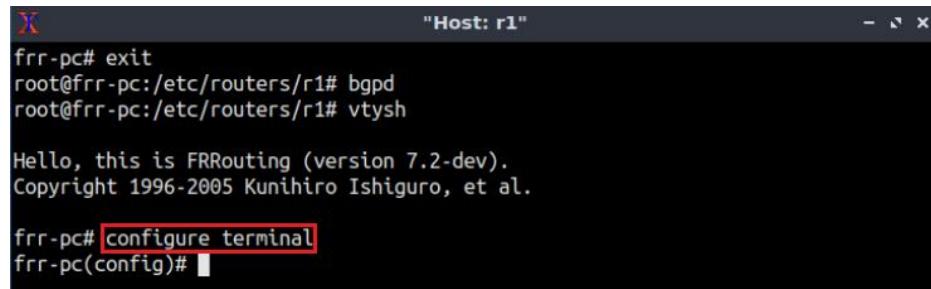
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc#
```

Figure 48. Starting vtysh in router r1.

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

```
configure terminal
```



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

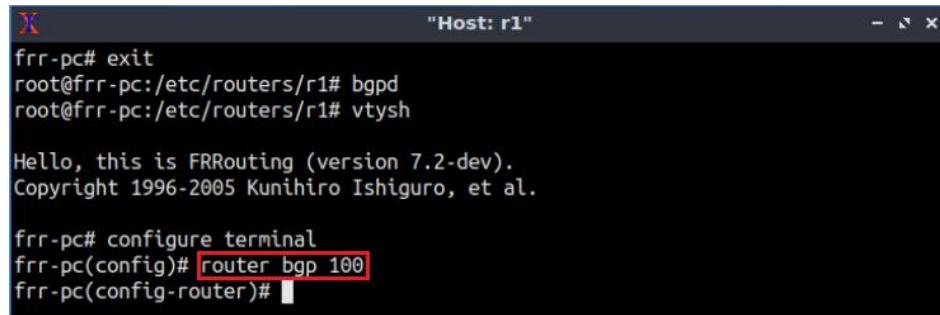
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

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

Figure 49. Enabling configuration mode in router r1.

Step 5. The ASN assigned for router r1 is 100. In order to configure BGP, type the following command:

```
router bgp 100
```



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

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

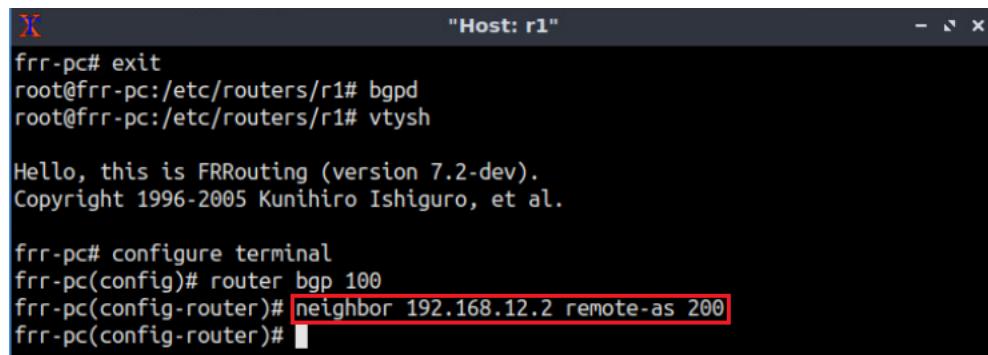
frr-pc# configure terminal
frr-pc(config)# router bgp 100
frr-pc(config-router)#

```

Figure 50. Configuring BGP on router r1.

Step 6. To configure a BGP neighbor to router r1 (AS 100), type the command shown below. This command specifies the neighbor IP address (192.168.12.2) and the ASN of the remote BGP peer (AS 200).

```
neighbor 192.168.12.2 remote-as 200
```



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

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

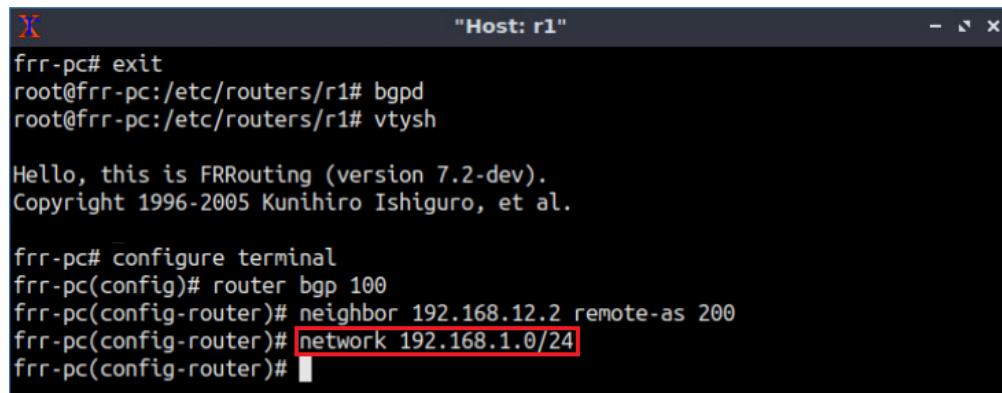
frr-pc# configure terminal
frr-pc(config)# router bgp 100
frr-pc(config-router)# neighbor 192.168.12.2 remote-as 200
frr-pc(config-router)#

```

Figure 51. Assigning BGP neighbor to router r1.

Step 7. In this step, router r1 will advertise the LAN 192.168.1.0/24 to its BGP peers. To do so, issue the following command:

```
network 192.168.1.0/24
```



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

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

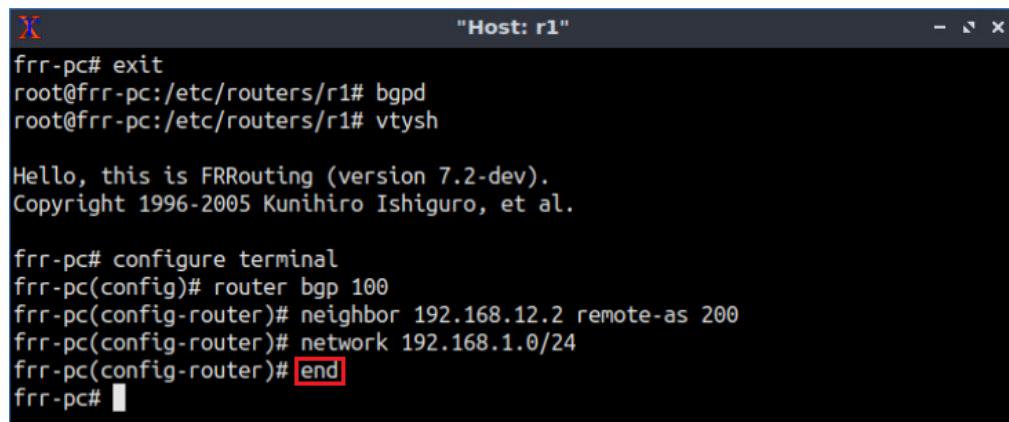
frr-pc# configure terminal
frr-pc(config)# router bgp 100
frr-pc(config-router)# neighbor 192.168.12.2 remote-as 200
frr-pc(config-router)# network 192.168.1.0/24
frr-pc(config-router)#

```

Figure 52. Advertising local network in router r1.

Step 8. Type the following command to exit from configuration mode.

```
end
```



```
frr-pc# exit
root@frr-pc:/etc/routers/r1# bgpd
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 bgp 100
frr-pc(config-router)# neighbor 192.168.12.2 remote-as 200
frr-pc(config-router)# network 192.168.1.0/24
frr-pc(config-router)# end
frr-pc#
```

Figure 53. Exiting from configuration mode.

Step 9. Follow from step 4 to step 6 but with different metrics in order to configure EBGP on router r2. All the steps are summarized in the following figure.

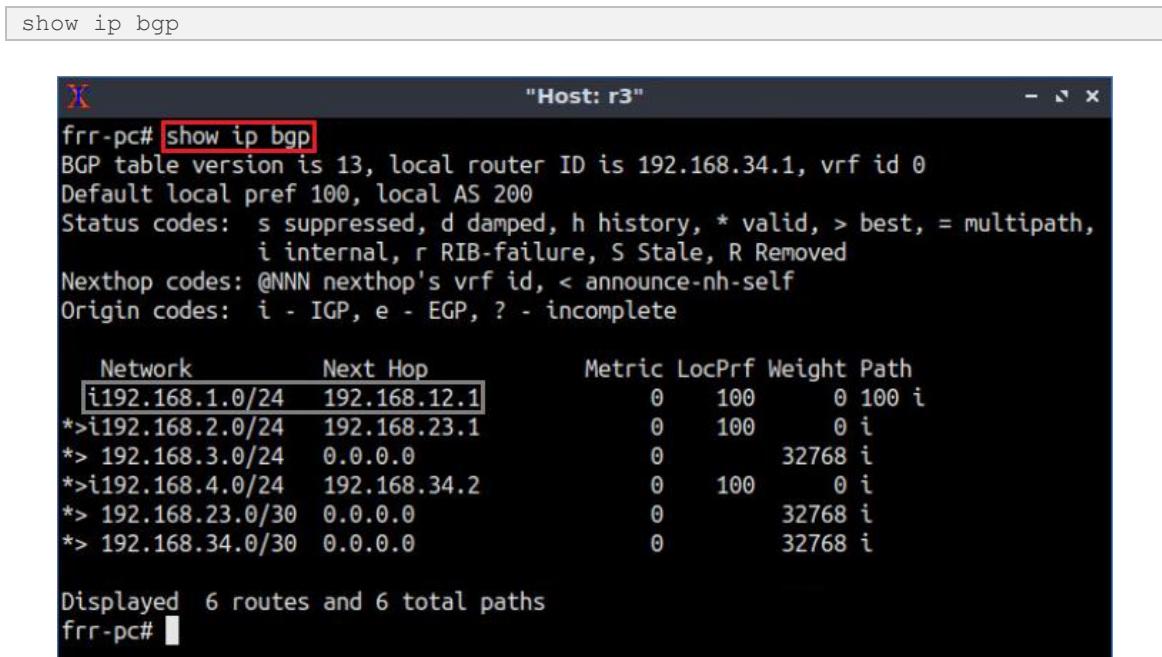


```
frr-pc# configure terminal
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.12.1 remote-as 100
frr-pc(config-router)#
```

Figure 54. Configuring BGP on router r2.

At this point, routers r1 and r2 are EBGP neighbors and the network 192.168.1.0/24 should be advertised from router r1 to router r2. Additionally, router r2 should advertise EBGP learned routes to its IBGP neighbor routers (router r3).

Step 10. In router r3 terminal, type the following command to verify BGP networks.



```
show ip bgp
```

```
frr-pc# show ip bgp
BGP table version is 13, local router ID is 192.168.34.1, vrf id 0
Default local pref 100, local AS 200
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
              i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop           Metric LocPrf Weight Path
i192.168.1.0/24    192.168.12.1        0     100      0 100 i
*>i192.168.2.0/24  192.168.23.1        0     100      0 i
*> 192.168.3.0/24  0.0.0.0            0          32768 i
*>i192.168.4.0/24  192.168.34.2        0     100      0 i
*> 192.168.23.0/30 0.0.0.0            0          32768 i
*> 192.168.34.0/30 0.0.0.0            0          32768 i

Displayed 6 routes and 6 total paths
frr-pc#
```

Figure 55. Verifying BGP networks on router r2.

Consider Figure 55. The BGP table of router r3 has a route to the network 192.168.1.0/24. The code `*>` is not shown next to this network, which means that it is not reachable. Router r3 uses the next hop 192.168.12.1 to reach the network 192.168.1.0/24, however, router r3 does not have a route to the network 192.168.12.0/30. Router r2 will be configured so that it updates the next hop to itself when it advertises EBGP routes to IBGP neighbors.

Step 11. In router r2 terminal, configure BGP so that the IBGP neighbor 192.168.23.2 (router r3) uses router r2 as the next hop to the network 192.168.1.0/24. To do so, type the following command:

```
neighbor 192.168.23.2 next-hop-self
```

```
frr-pc# configure terminal
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.12.1 remote-as 100
frr-pc(config-router)# neighbor 192.168.23.2 next-hop-self
frr-pc(config-router)#

```

Figure 56. Configuring BGP in router r2.

5.2 Verify the connectivity of the configured topology

Step 1. Type the following command to verify the BGP table of router r3. You will notice that the BGP table contains a route to the network 192.168.1.0/24 via the next hop 192.168.23.1 (router r2).

```
show ip route
```

```
frr-pc# show ip bgp
BGP table version is 14, local router ID is 192.168.34.1, vrf id 0
Default local pref 100, local AS 200
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
              i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

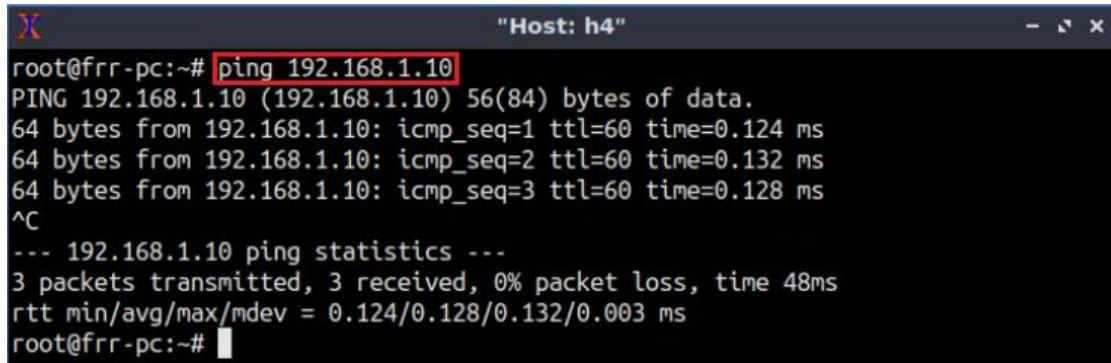
      Network          Next Hop           Metric LocPrf Weight Path
*->i192.168.1.0/24  192.168.23.1        0       100      0 100 i
*->i192.168.2.0/24  192.168.23.1        0       100      0 i
*> 192.168.3.0/24   0.0.0.0            0           32768 i
*->i192.168.4.0/24  192.168.34.2        0       100      0 i
*> 192.168.23.0/30  0.0.0.0            0           32768 i
*> 192.168.34.0/30  0.0.0.0            0           32768 i

Displayed 6 routes and 6 total paths
frr-pc#
```

Figure 57. Displaying the routing table of router r3.

Step 2. In host h4 terminal, perform a connectivity between host h4 and host h1 by issuing the command shown below. To stop the test, press **Ctrl+c**. The result will show a successful connectivity test.

```
ping 192.168.1.10
```



The screenshot shows a terminal window titled "Host: h4". The command `ping 192.168.1.10` is entered at the root prompt. The output shows three ICMP echo requests being sent to the target host, with their sequence numbers, TTL values, and round-trip times. After the third request, the user presses `Ctrl+C` to stop the ping. The final statistics line shows 3 packets transmitted, 3 received, 0% packet loss, and a minimum/average/max round-trip time of 48ms.

```
root@frrr-pc:~# ping 192.168.1.10
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_seq=1 ttl=60 time=0.124 ms
64 bytes from 192.168.1.10: icmp_seq=2 ttl=60 time=0.132 ms
64 bytes from 192.168.1.10: icmp_seq=3 ttl=60 time=0.128 ms
^C
--- 192.168.1.10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 48ms
rtt min/avg/max/mdev = 0.124/0.128/0.132/0.003 ms
root@frrr-pc:~#
```

Figure 58. Connectivity test using `ping` command.

This concludes Lab 10. Stop the emulation and then exit out of MiniEdit.

References

1. A. Tanenbaum, D. Wetherall, "Computer networks", 5th Edition, Pearson, 2012.
2. T. Bates, E. Chen, R. Chandra, "BGP route reflection: an alternative to full mesh internal BGP (IBGP)", 2006. [Online]. Available: <https://www.ietf.org/rfc/rfc4456.txt>



BORDER GATEWAY PROTOCOL

Lab 11: Configuring Multiprotocol BGP

Document Version: **03-18-2020**



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 IPv4 and IPv6 addresses	4
1.2 Intradomain and Interdomain routing protocols.....	4
1.3 MP-BGP	5
2 Lab topology.....	6
2.1 Lab settings.....	7
2.2 Open topology and load the configuration.....	8
2.3 Configure and verify the hosts	11
2.4 Load zebra daemon and verify the Connectivity	14
3 Configure and verify OSPF on router r2 and router r3	17
3.1 Configure OSPFv2 for IPv4 networks	19
3.2 Configure OSPFv3 for IPv6 networks	22
3.3 Verify connectivity between router r2 and router r3	25
4 Configure and verify BGP for IPv4 networks	27
4.1 Configure and verify EBGP on router r1.....	27
4.2 Configure and verify EBGP and IBGP on router r2	31
4.3 Configure and verify IBGP on router r3.....	35
5 Configure and verify BGP for IPv6 networks	36
5.1 Configure and verify EBGP on router r1.....	36
5.2 Configure and verify EBGP and IBGP on router r2	39
5.3 Configure and verify IBGP on router r3.....	44
6 Verify BGP configuration.....	45
References	46

Overview

This lab introduces Multiprotocol Border Gateway Protocol (MP-BGP). This protocol makes BGP available for other network layer protocols, including Internet Protocol version 6 (IPv6). For Interior Gateway Protocol (IGP), Open Shortest Path First version 2 (OSPFv2) and OSPFv3 will be configured to advertise IPv4 and IPv6 addresses. For Exterior Gateway Protocol (EGP), Internal BGP (IBGP) and External BGP (EBGP) will be configured to advertise IPv4 and IPv6 across Autonomous Systems (ASes).

Objectives

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

1. Assign IPv6 addresses on hosts.
2. Configure OSPFv2 and OSPFv3.
3. Configure and verify IBGP and EBGP.
4. Use MP-BGP to distribute IPv4 and IPv6 in parallel.

Lab settings

The information in Table 1 provides the credentials to access Client1 machine.

Table 1. Credentials to access Client1 machine.

Device	Account	Password
Client1	admin	password

Lab roadmap

This lab is organized as follows:

1. Section 1: Introduction.
2. Section 2: Lab topology.
3. Section 3: Configure and verify OSPF on router r2 and router r3.
4. Section 4: Configure and verify BGP for IPv4 networks.
5. Section 5: Configure and verify BGP for IPv6 networks.
6. Section 6: Verify BGP configuration.

1 Introduction

1.1 IPv4 and IPv6 addresses

The IP transmits blocks of data called datagrams from the source to the destination that are identified by a fixed length address. IPv4 is the dominant protocol of the Internet that identifies devices on a network. It uses a 32-bit address space which allows to store more than 4 billion addresses¹.

As the Internet evolves, more IP addresses are required than the IPv4 offers. Thus, IPv6 was designed to fulfill the need for more Internet addresses. IPv6 uses a 64-bit address space which allows to store a huge number of addresses (more than 340 undecillion)².

IPv6 has multiple address types. The link local address is used on a single link, or within a Local Area Network (LAN). Link local addresses do not have to be globally unique, thus, they are not routable. The prefix of the link local address is fe80::/10. Global unicast address is globally unique, and it is used to identify a single interface on the Internet. The prefix of the global unicast address is 2000::/3³.

Consider Figure 1. In IPv6, the link local address is used between routers that are directly connected, whereas the global address is used between routers that do not share a common link.

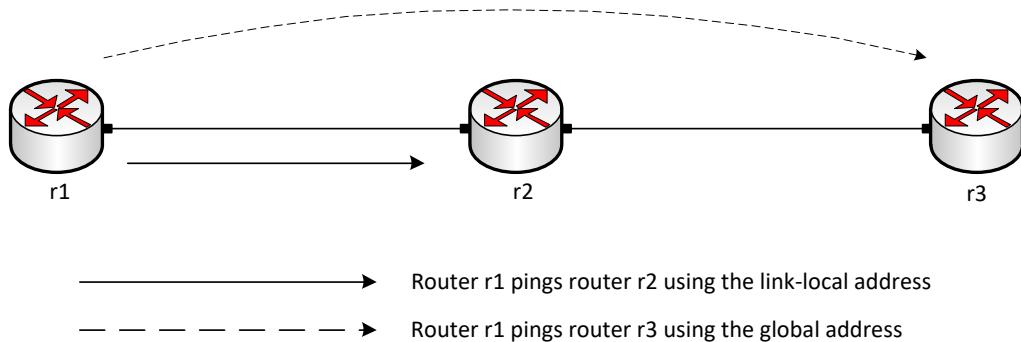


Figure 1. IPv6 link local and global addresses.

1.2 Intradomain and Interdomain routing protocols

The Internet consists of many independent administrative domains, referred to as ASes. ASes are operated by different organizations, which can run their own internal routing protocols. A routing protocol that runs within an AS is referred to as intradomain routing protocol. One of the most widely used intradomain protocols is OSPF. Since an AS may be large and nontrivial to manage, OSPF allows an AS to be divided into numbered areas⁴. An area is a logical collection of networks, routers, and links. All routers in the same area have detailed information of the topology within their area⁵. Traditionally, OSPF only supported IPv4 addresses (OSPFv2); however, it was redesigned to support IPv6 as well (OSPFv3)⁶.

A routing protocol that runs between ASes is referred to as interdomain routing protocol. ASes may use different intradomain routing protocols; however, they must use the same

interdomain routing protocol, i.e., BGP. BGP allows the enforcement of different routing policies on the traffic from one AS to another. For example, a security policy can prevent the dissemination of routing information from one AS to another⁴. BGP has been extended to carry routing information for multiple protocols, such as IPv6⁷.

BGP is referred to as EBGP when it is running between different ASes, whereas it is referred to as IBGP when it is running within an AS⁴. IBGP is usually used to distribute the EBGP learned routes among the routers within the same AS⁴.

Consider Figure 2. The intradomain routing protocol within AS 100 is OSPF, and the interdomain routing protocol between AS 100 and AS 200 is BGP (EBGP). Routers within the same AS advertise their EBGP learned routes among each other through IBGP.

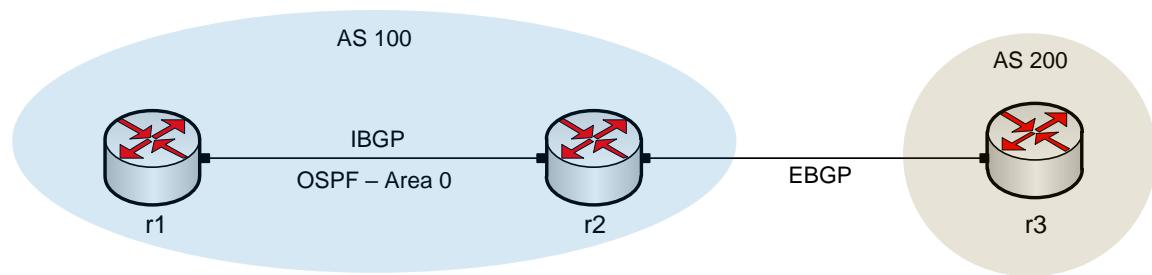


Figure 2. Routers that exchange information within the same AS use OSPF and IBGP, while routers that exchange information between different ASes use EBGP.

1.3 MP-BGP

In an IPv4 environment, BGP establishes sessions using IPv4, i.e., BGP peers are configured with IPv4 addresses. The routes that are advertised by BGP also have IPv4 addresses⁸.

MP-BGP was introduced to make BGP available for other network layer protocols, including IPv6. In an environment where IPv4 and IPv6 are both configured, MP-BGP routers can become neighbors using IPv4 addresses and exchange IPv6 prefixes or the other way around⁸.

Consider Figure 3. The BGP session established between router r1 and router r2 is done using IPv4. Using MP-BGP, router r1 can advertise the address of the attached IPv6 LAN through the current BGP session.

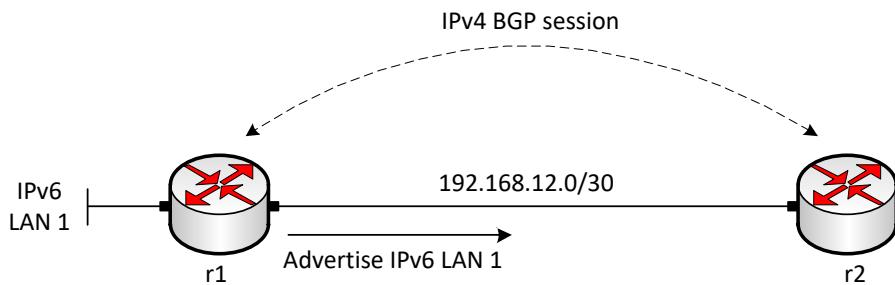


Figure 3. Advertising an IPv6 LAN address through a BGP IPv4 session.

Similarly, consider Figure 4. The BGP session established between router r1 and router r2 is done using IPv6. Using MP-BGP, router r1 can advertise the address of the attached IPv4 LAN through the current BGP session.

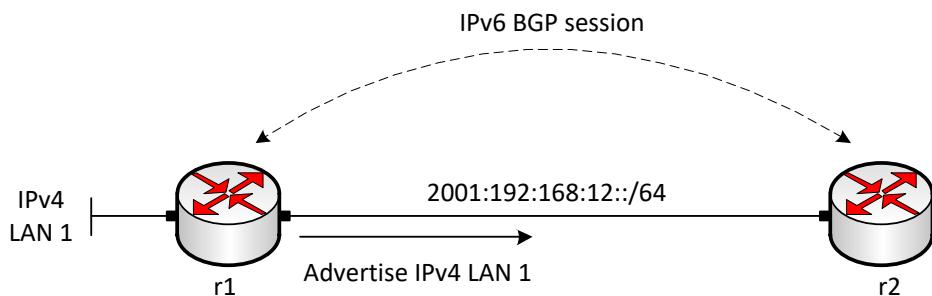


Figure 4. Advertising an IPv4 LAN address through a BGP IPv6 session.

2 Lab topology

Consider Figure 5. The topology consists of two ASes. The Internet Service Provider (ISP), i.e., router r1, provides Internet service to the Campus network (routers r2 and r3). The ASN assigned to the ISP and the Campus network are 100 and 200, respectively. The ISP communicates with the Campus via EBGP routing protocol, and the routers within the Campus network communicate using IBGP and OSPF.

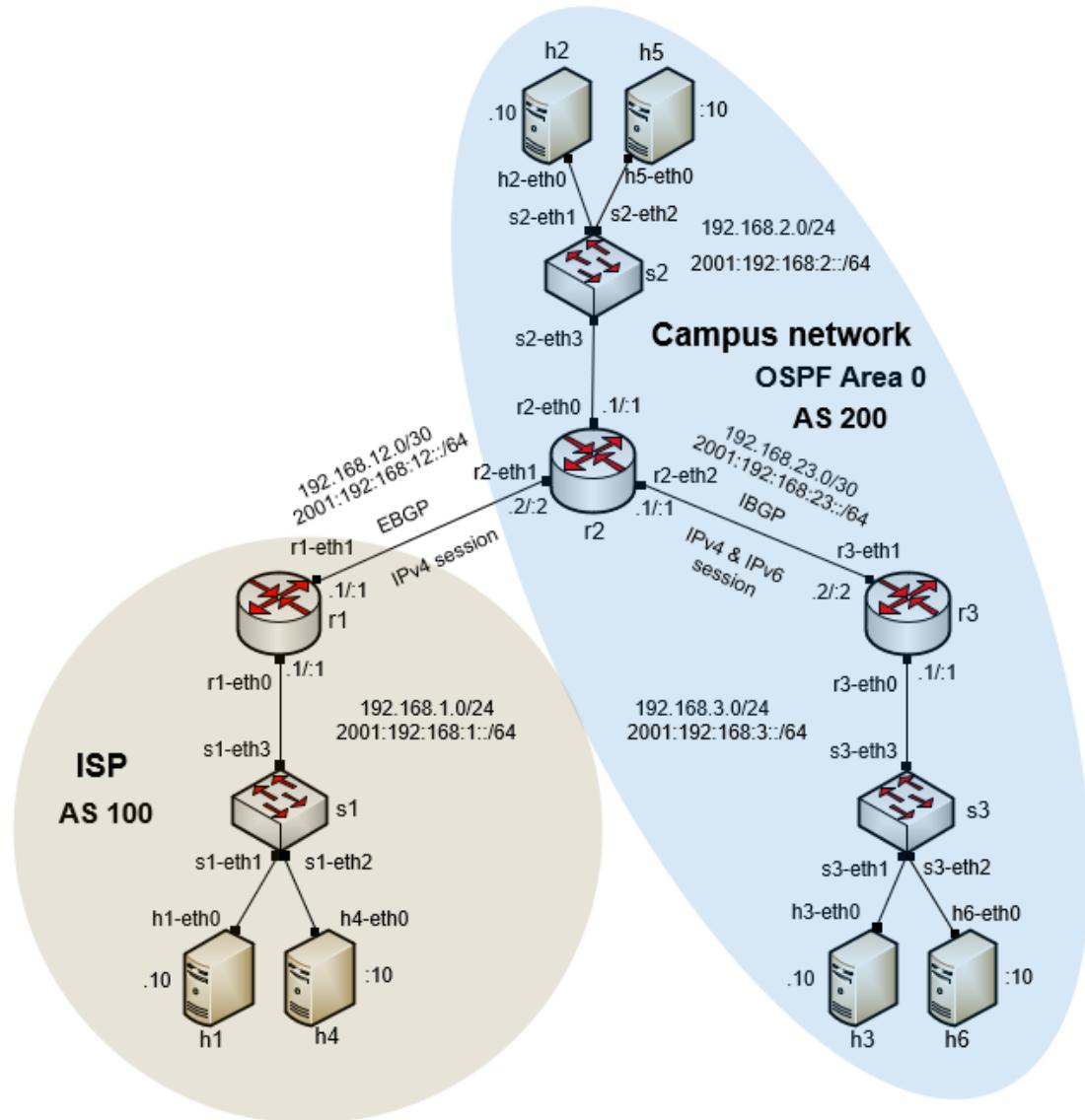


Figure 5. Lab topology.

2.1 Lab settings

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

Table 2. Topology information.

Device	Interface	IPV4 Address	IPV6 Address	Default gateway
r1 (ISP)	r1-eth0	192.168.1.1/24	2001:192:168:1::1/64	N/A
	r1-eth1	192.168.12.1/30	2001:192:168:12::1/64	N/A
	lo	192.168.11.1/32	2001:192:168:11::1/64	N/A
	r2-eth0	192.168.2.1/24	2001:192:168:2::1/64	N/A

r2 (Campus network)	r2-eth1	192.168.12.2/30	2001:192:168:12::2/64	N/A
	r2-eth2	192.168.23.1/30	2001:192:168:23::1/64	N/A
	lo	192.168.22.1/32	2001:192:168:22::1/64	N/A
r3 (Campus network)	r3-eth0	192.168.3.1/24	2001:192:168:3::1/64	N/A
	r3-eth1	192.168.23.2/30	2001:192:168:23::2/64	N/A
	lo	192.168.33.1/32	2001:192:168:33::1/64	N/A
h1	h1-eth0	192.168.1.10/24	N/A	192.168.1.1
h2	h2-eth0	192.168.2.10/24	N/A	192.168.2.1
h3	h3-eth0	192.168.3.10/24	N/A	192.168.3.1
h4	h4-eth0	N/A	2001:192:168:1::10/64	2001:192:168:1::1
h5	h5-eth0	N/A	2001:192:168:2::10/64	2001:192:168:1::1
h6	h6-eth0	N/A	2001:192:168:3::10/64	2001:192:168:1::1

2.2 Open topology and load the configuration

Step 1. Start by launching Miniedit by clicking on Desktop's shortcut. When prompted for a password, type `password`.



Figure 6. MiniEdit shortcut.

Step 2. On Miniedit's menu bar, click on *File* then *open* to load the lab's topology. Locate the *Lab11.mn* topology file in the default directory, */home/frr/BGP_Labs/lab11* and click on *Open*.

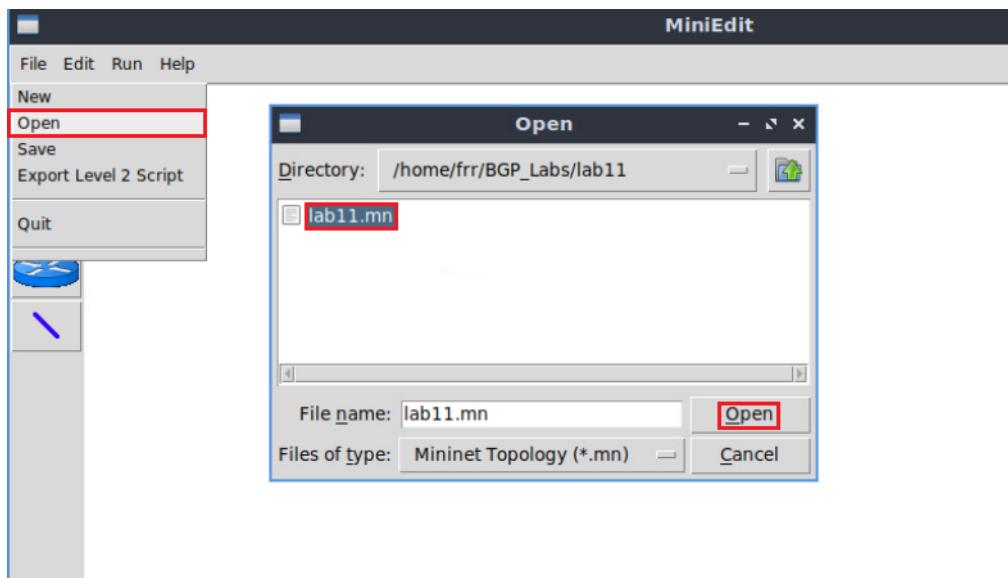


Figure 7. MiniEdit's Open dialog.

At this point the topology is loaded with all the required network components. You will execute a script that will load the configuration of the routers.

Step 3. Open the Linux Terminal.

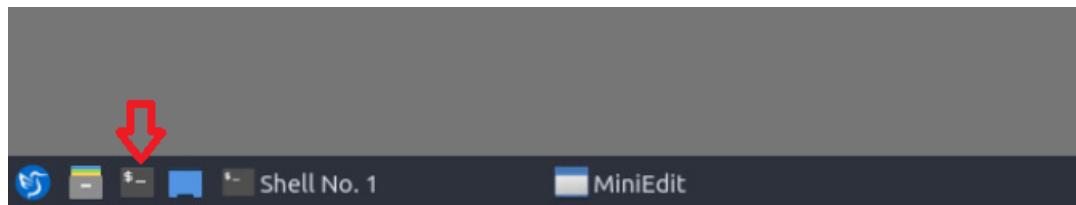


Figure 8. Opening Linux terminal

Step 4. Click on the Linux's terminal and navigate into *BGP_Labs/lab11* 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 BGP_Labs/lab11
```

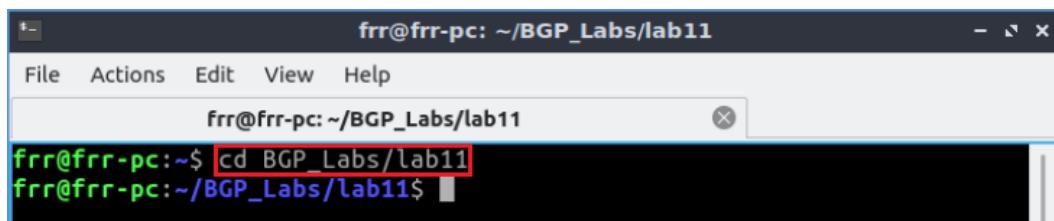
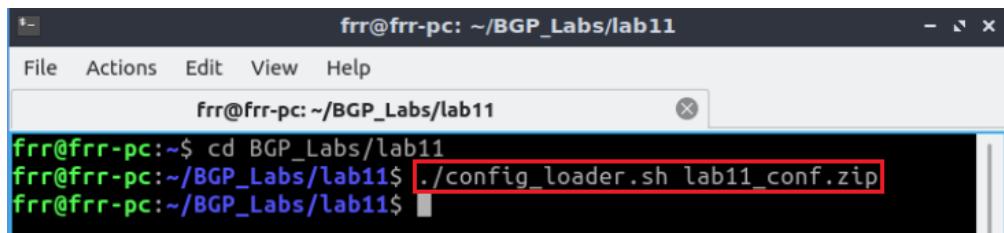


Figure 9. Entering to the *BGP_Labs/lab11* directory.

Step 5. To execute the shell script, type the following command. The argument of the program corresponds to the configuration zip file that will be loaded in all the routers in the topology.

```
./config_loader.sh lab11_conf.zip
```

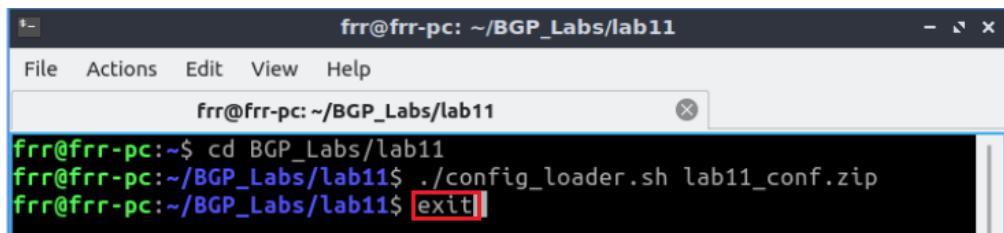


```
frr@frr-pc:~/BGP_Labs/lab11
frr@frr-pc:~/BGP_Labs/lab11$ cd BGP_Labs/lab11
frr@frr-pc:~/BGP_Labs/lab11$ ./config_loader.sh lab11_conf.zip
frr@frr-pc:~/BGP_Labs/lab11$
```

Figure 10. Executing the shell script to load the configuration.

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

```
exit
```



```
frr@frr-pc:~/BGP_Labs/lab11
frr@frr-pc:~/BGP_Labs/lab11$ cd BGP_Labs/lab11
frr@frr-pc:~/BGP_Labs/lab11$ ./config_loader.sh lab11_conf.zip
frr@frr-pc:~/BGP_Labs/lab11$ exit
```

Figure 11. Exiting from the terminal.

Step 7. At this point hosts h1, h2 and h3 interfaces are configured. To proceed with the emulation, click on the *Run* button located in lower left-hand side.



Figure 12. Starting the emulation.

Step 8. Click on Mininet's terminal, i.e., the one launched when MiniEdit was started.

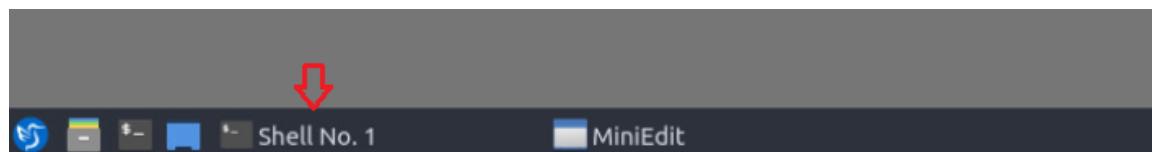
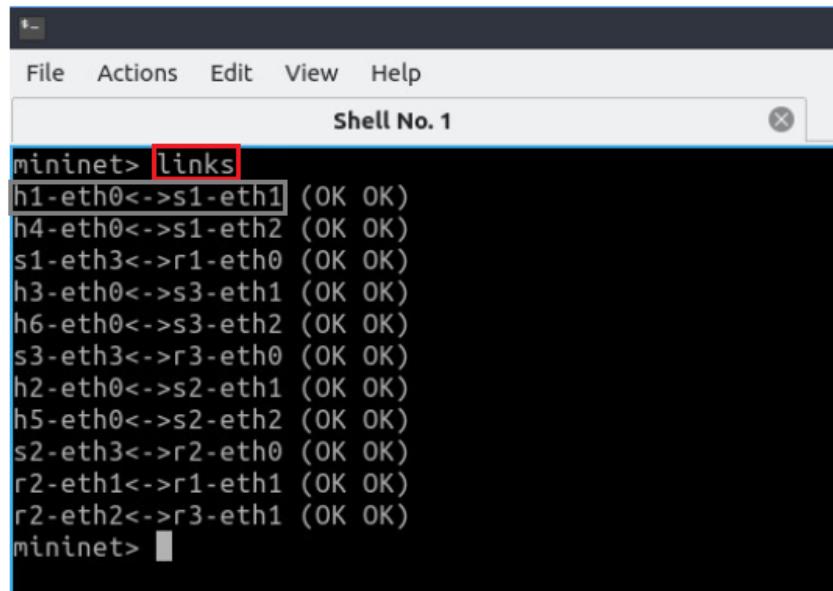


Figure 13. Opening Mininet's terminal.

Step 9. Issue the following command to display the interface names and connections.

```
links
```



```
File Actions Edit View Help
Shell No. 1
mininet> links
h1-eth0<->s1-eth1 (OK OK)
h4-eth0<->s1-eth2 (OK OK)
s1-eth3<->r1-eth0 (OK OK)
h3-eth0<->s3-eth1 (OK OK)
h6-eth0<->s3-eth2 (OK OK)
s3-eth3<->r3-eth0 (OK OK)
h2-eth0<->s2-eth1 (OK OK)
h5-eth0<->s2-eth2 (OK OK)
s2-eth3<->r2-eth0 (OK OK)
r2-eth1<->r1-eth1 (OK OK)
r2-eth2<->r3-eth1 (OK OK)
mininet> █
```

Figure 14. Displaying network interfaces.

In Figure 14, the link displayed within the gray box indicates that interface eth0 of host h1 connects to interface eth1 of switch s1 (i.e., $h1\text{-}eth0 <-> s1\text{-}eth1$).

2.3 Configure and verify the hosts

You will verify IPv4 addresses of each host (h1, h2 and h3) following Table 2. Additionally, you will assign the IP addresses and the default gateway to IPv6 hosts (h4, h5 and h6).

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.

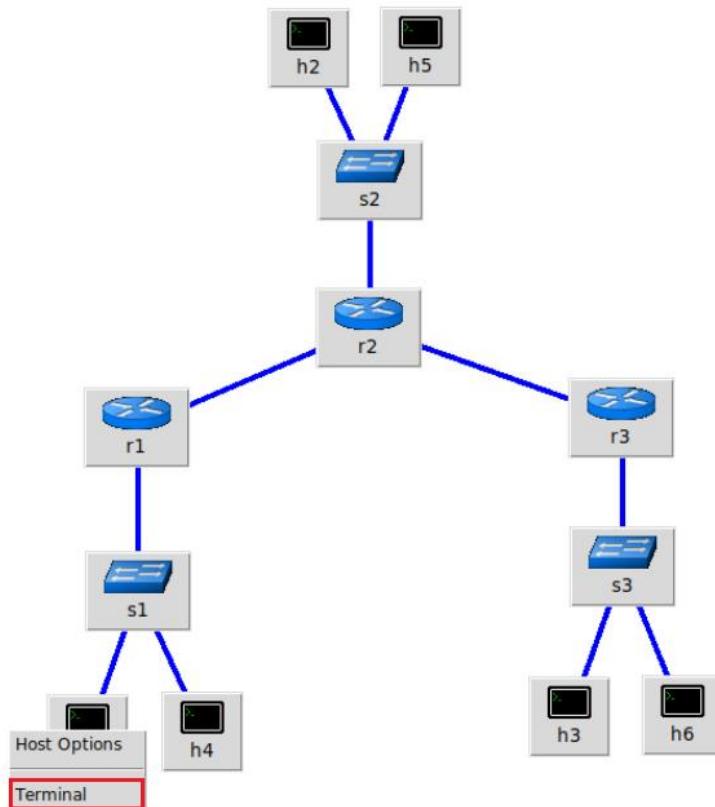


Figure 15. Opening terminal on host h1.

Step 2. On 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.

```
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
                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 16. Output of `ifconfig` command.

Step 3. On 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 17. Output of `route` command.

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

Step 5. On host h4 terminal, type the following command to assign IPv6 address.

```
ifconfig h4-eth0 inet6 add 2001:192:168:1::10/64
```

```
"Host: h4"
root@frr-pc:~# ifconfig h4-eth0 inet6 add 2001:192:168:1::10/64
root@frr-pc:~#
```

Figure 18. Assigning IPv6 address on host h4.

Step 6. Type the following command to assign IPv6 default gateway on host h4.

```
ip -6 route add default via 2001:192:168:1::1
```

```
"Host: h4"
root@frr-pc:~# ifconfig h4-eth0 inet6 add 2001:192:168:1::10/64
root@frr-pc:~# ip -6 route add default via 2001:192:168:1::1
root@frr-pc:~#
```

Figure 19. Assigning IPv6 default gateway on host h4.

Step 7. Type the following command to verify IPv6 addresses on host h4. The following figure contains two IPv6 addresses.

```
ifconfig
```

```

root@frr-pc:~# ifconfig
h4-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::d02c:46ff:fedb:e073 brd fe80::ff02:1c:46ff:fedb:e073/64 scopeid 0x20<link>
        inet6 2001:192:168:1::10 brd 2001:192:168:1::ff/64 scopeid 0x0<global>
            ether d2:2c:46:db:e0:73 txqueuelen 1000 (Ethernet)
            RX packets 38 bytes 4203 (4.2 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 14 bytes 1172 (1.1 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 20. Output of `ifconfig` command.

Consider Figure 20. There are two IPv6 addresses assigned to host h4. IPv6 addresses fe80::d02c:46ff:fedb:e073 and 2001:192:168:1::10 are assigned as link local address and global address, respectively. You may notice a different link local address as they are assigned randomly in FRR. Link local address is used to communicate with directly connected networks. IPv6 global addresses are like IPv4 public addresses which is globally unique.

Step 8. Follow from step 5 to step 6 to assign IPv6 addresses on host h5. These steps are summarized in the figure below.

```

root@frr-pc:~# ifconfig h5-eth0 inet6 add 2001:192:168:2::10/64
root@frr-pc:~# ip -6 route add default via 2001:192:168:2::1
root@frr-pc:~#

```

Figure 21. Assigning IPv6 address on host h5.

Step 9. Follow from step 5 to step 6 to assign IPv6 addresses on host h6. These steps are summarized in the figure below.

```

root@frr-pc:~# ifconfig h6-eth0 inet6 add 2001:192:168:3::10/64
root@frr-pc:~# ip -6 route add default via 2001:192:168:3::1
root@frr-pc:~#

```

Figure 22. Assigning IPv6 address on host h6.

2.4 Load zebra daemon and verify the Connectivity

In this section, you will verify the routing table of routers r1, r2, and r3.

Step 1. You will validate that the router interfaces are configured correctly according to Table 2. In order to verify router r1, hold right-click on router r1 and select *Terminal*.

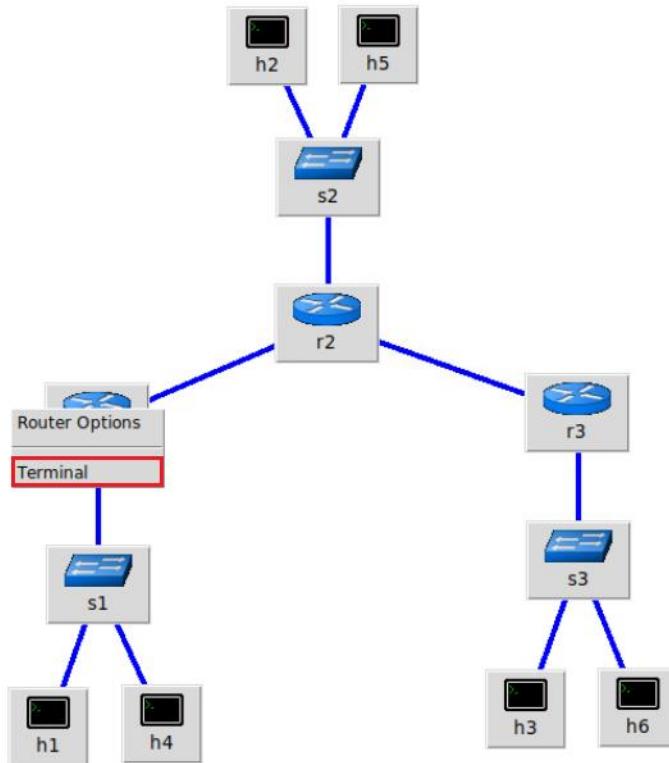


Figure 23. Opening terminal on router r1.

Step 2. In this step, you will start zebra daemon, which is a multi-server routing software that provides TCP/IP based routing protocols. The configuration will not be working if you do not enable zebra daemon initially. In order 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 24. 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:

```
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 25. Starting vtysh on router r1.

Step 4. Type the following command on router r1 terminal to verify the routing table of router r1. It will list all the connected IPv4 networks. The routing table of router r1 does not contain any route to the networks attached to router r3 (192.168.23.0/30, 192.168.3.0/24) 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, 01:20:22
C>* 192.168.11.1/32 is directly connected, lo, 01:20:22
C>* 192.168.12.0/30 is directly connected, r1-eth1, 01:20:22
frr-pc#
```

Figure 26. Displaying the routing table for IPv4 routes of router r1.

Step 5. Type the following command on router r1 terminal to verify the routing table for IPv6 routes. You will verify link local addresses (fe80::/64) for each link which is automatically configured in FRR to communicate with directly connected networks. The routing table of router r1 does not contain any route to the networks attached to router r3 (2001:192:168:23::/64, 2001:192:168:3::/64) as there is no routing protocol configured yet.

```
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,
       > - selected route, * - FIB route, q - queued route, r - rejected route

C>* 2001:192:168:1::/64 is directly connected, r1-eth0, 01:17:38
C>* 2001:192:168:11::/64 is directly connected, lo, 01:17:40
C>* 2001:192:168:12::/64 is directly connected, r1-eth1, 01:17:38
C * fe80::/64 is directly connected, r1-eth1, 01:17:40
C>* fe80::/64 is directly connected, r1-eth0, 01:17:40
frr-pc#
```

Figure 27. Displaying the routing table for IPv6 routes of router r1.

Step 6. 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.

```

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 ipv6 route
Codes: K - kernel route, C - connected, S - static, R - RIPng,
      0 - 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>* 2001:192:168:2::/64 is directly connected, r2-eth0, 00:05:51
C>* 2001:192:168:12::/64 is directly connected, r2-eth1, 00:05:51
C>* 2001:192:168:22::/64 is directly connected, lo, 00:05:52
C>* 2001:192:168:23::/64 is directly connected, r2-eth2, 00:05:50
C * fe80::/64 is directly connected, r2-eth2, 00:05:52
C * fe80::/64 is directly connected, r2-eth1, 00:05:52
C>* fe80::/64 is directly connected, r2-eth0, 00:05:52
frr-pc#

```

Figure 28. Displaying the routing table for IPv6 routes of router r2.

Step 7. Router r3 is configured similarly to router r1 but, with different IP addresses (see Table 2). Those steps are summarized in the following figure. To proceed, in router r3 terminal, issue the commands depicted below. At the end, you verify all the directly connected networks of router r3.

```

root@frr-pc:/etc/routers/r3# zebra
root@frr-pc:/etc/routers/r3# vtysh

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

frr-pc# show ipv6 route
Codes: K - kernel route, C - connected, S - static, R - RIPng,
      0 - 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>* 2001:192:168:3::/64 is directly connected, r3-eth0, 00:00:05
C>* 2001:192:168:23::/64 is directly connected, r3-eth1, 00:00:05
C>* 2001:192:168:33::/64 is directly connected, lo, 00:00:06
C * fe80::/64 is directly connected, r3-eth1, 00:00:06
C>* fe80::/64 is directly connected, r3-eth0, 00:00:06
frr-pc#

```

Figure 29. Displaying the routing table for IPv6 routes of router r3.

3 Configure and verify OSPF on router r2 and router r3

In this section, you will configure OSPF routing protocol in router r2 and router r3. First, you will enable the OSPF daemon on the routers. Second, you will establish a single area OSPF, which is classified as area 0 or backbone area. Finally, all the directly connected networks (except 192.168.12.0/30 and 2001:192:168:12::/64) will be advertised in area 0 between routers r2, r3. Networks 192.168.12.0/30 and 2001:192:168:12::/64 will be used to configure EBGP between routers r1 and r2 in the following section.

Step 1. To configure OSPF routing protocol, you need to enable the OSPF daemon first. In router r2, type the following command to exit the vtysh session.

```
exit
```

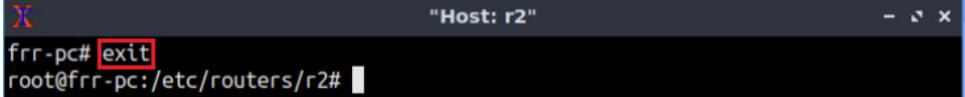


Figure 30. Exiting the vtysh session.

Step 2. Type the following command on router r2 to enable OSPF daemon for IPv4 networks:

```
ospfd
```

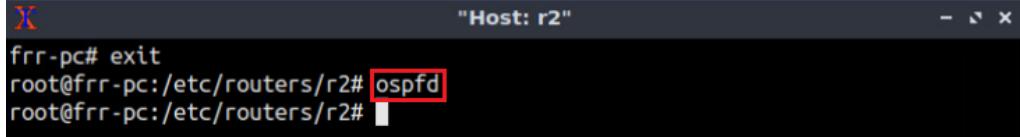


Figure 31. Starting OSPF daemon for IPv4 networks.

Step 3. Type the following command on router r2 to enable OSPF daemon for IPv6 networks:

```
ospf6d
```

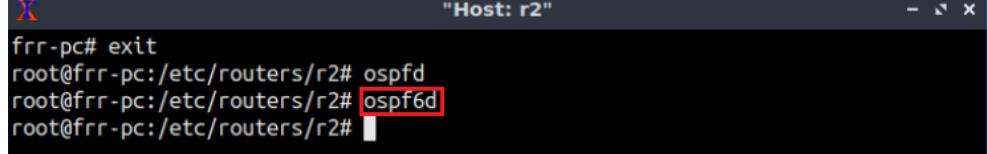


Figure 32. Starting OSPF daemon for IPv6 networks.

Step 4. In order to enter to router r2 terminal, issue the following command:

```
vtysh
```



```
"Host: r2"
frr-pc# exit
root@frr-pc:/etc/routers/r2# ospfd
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.

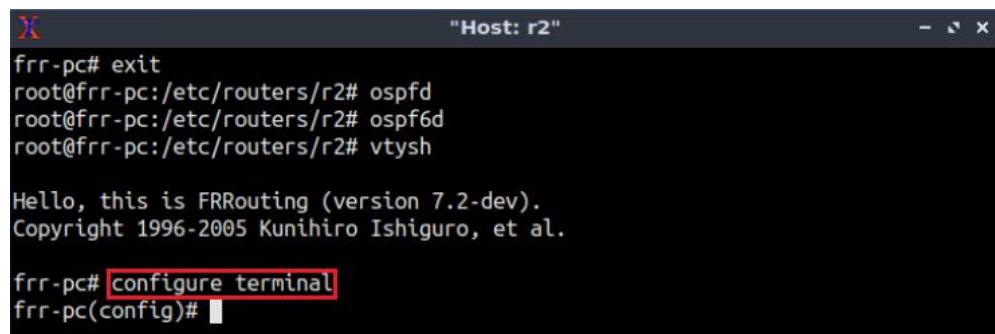
frr-pc#
```

Figure 33. Starting vtysh on router r2.

3.1 Configure OSPFv2 for IPv4 networks

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

```
configure terminal
```



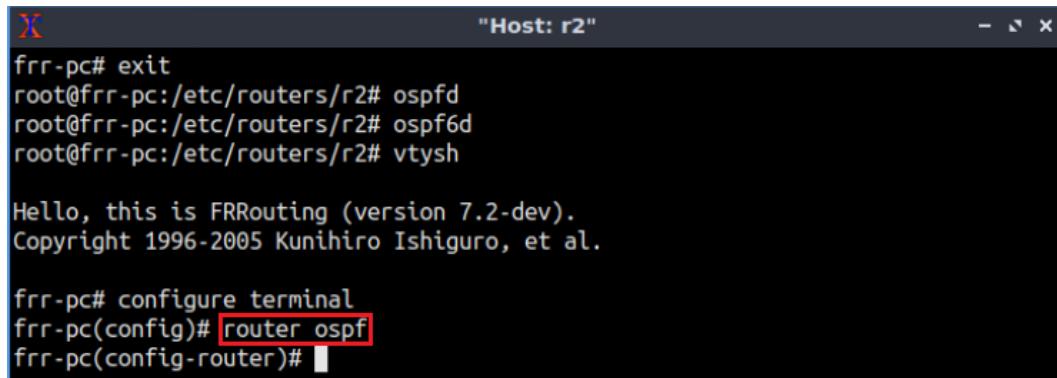
```
"Host: r2"
frr-pc# exit
root@frr-pc:/etc/routers/r2# ospfd
root@frr-pc:/etc/routers/r2# ospf6d
root@frr-pc:/etc/routers/r2# vtysh
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

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

Figure 34. Enabling configuration mode on router r2.

Step 2. In order to configure OSPF routing protocol, type the command shown below. This command will enable OSPF configuration mode where you can advertise the IPv4 networks directly connected to the router r2.

```
router ospf
```



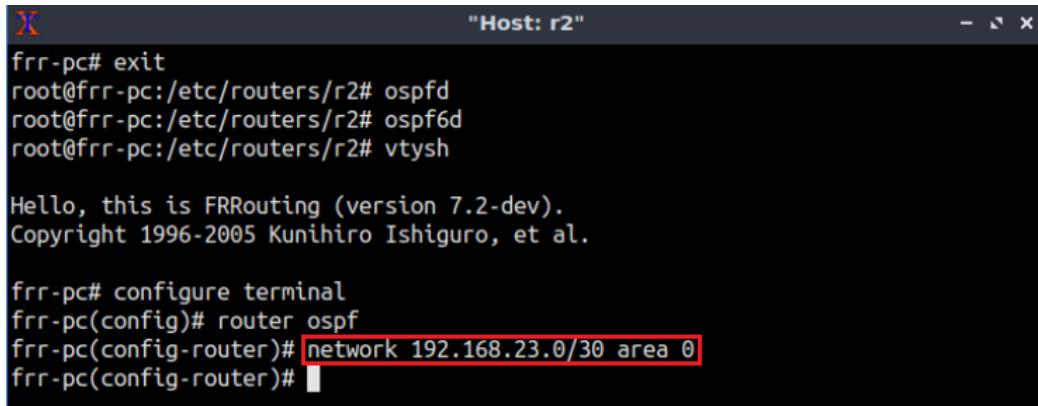
```
"Host: r2"
frr-pc# exit
root@frr-pc:/etc/routers/r2# ospfd
root@frr-pc:/etc/routers/r2# ospf6d
root@frr-pc:/etc/routers/r2# vtysh
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

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

Figure 35. Configuring OSPF for IPv4 networks on router r2.

Step 3. In this step, type the following command to enable the interface *r2-eth2*, corresponding to the network 192.168.23.0/30, to participate in the routing process. This network is associated with area 0.

```
network 192.168.23.0/30 area 0
```



The terminal window shows the following session:

```
frr-pc# exit
root@frr-pc:/etc/routers/r2# ospfd
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.

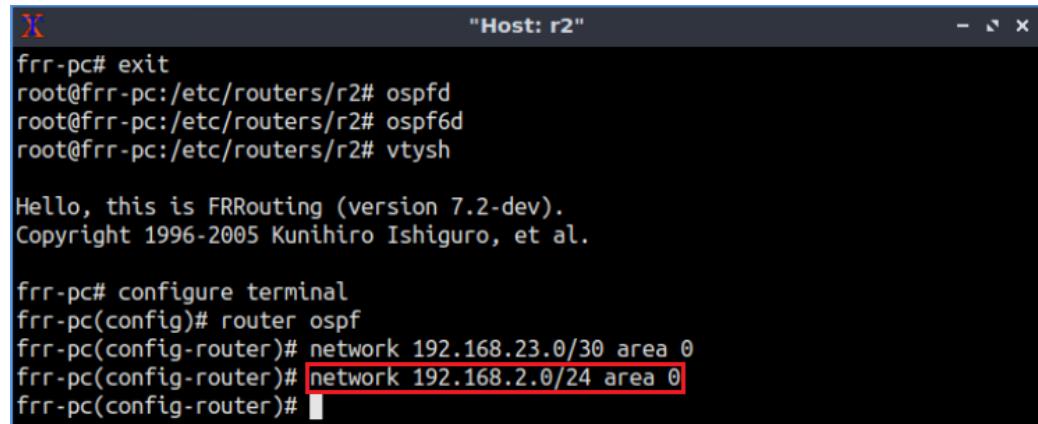
frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# network 192.168.23.0/30 area 0
frr-pc(config-router)#

```

Figure 36. Enabling the interface corresponding to 192.168.23.0/30 to participate in the OSPF routing process.

Step 4. Similarly, type the following command in router r2 terminal to enable the interface *r2-eth0* to participate in the OSPF routing process.

```
network 192.168.2.0/24 area 0
```



The terminal window shows the following session:

```
frr-pc# exit
root@frr-pc:/etc/routers/r2# ospfd
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.

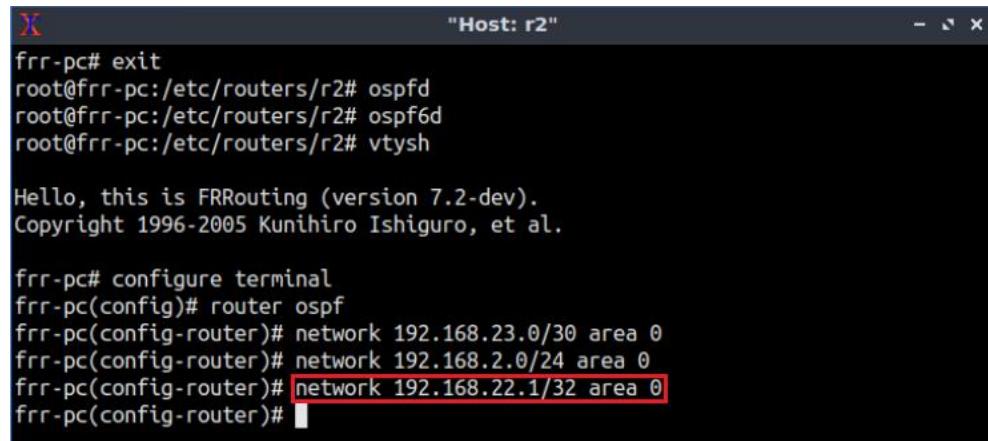
frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# network 192.168.23.0/30 area 0
frr-pc(config-router)# network 192.168.2.0/24 area 0
frr-pc(config-router)#

```

Figure 37. Enabling the interface corresponding to 192.168.2.0/24 to participate in the OSPF routing process.

Step 5. Type the following command to enable the loopback interface 192.168.22.1/32 to participate in the routing process.

```
network 192.168.22.1/32 area 0
```



```
"Host: r2"
frr-pc# exit
root@frr-pc:/etc/routers/r2# ospfd
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.

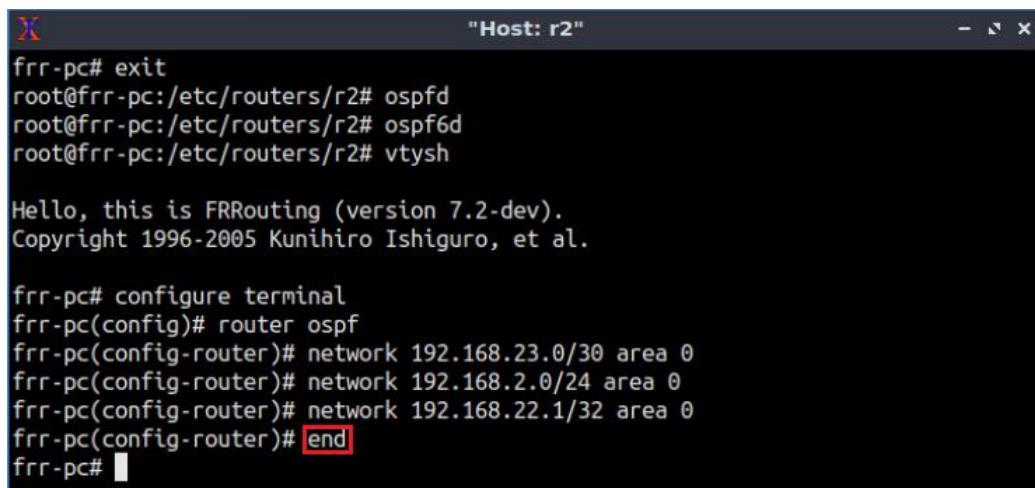
frr-pc# configure terminal
frr-pc(config)# router ospf
frr-pc(config-router)# network 192.168.23.0/30 area 0
frr-pc(config-router)# network 192.168.2.0/24 area 0
frr-pc(config-router)# network 192.168.22.1/32 area 0
frr-pc(config-router)#

```

Figure 38. Enabling the interface corresponding to 192.168.22.1/32 to participate in the OSPF routing process.

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

```
end
```



```
"Host: r2"
frr-pc# exit
root@frr-pc:/etc/routers/r2# ospfd
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 ospf
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 192.168.22.1/32 area 0
frr-pc(config-router)# end
frr-pc#
```

Figure 39. Exiting from configuration mode.

Step 7. 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, on router r3 terminal issue the commands depicted below.

```

frr-pc# exit
root@frr-pc:/etc/routers/r3# ospfd
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 ospf
frr-pc(config-router)# network 192.168.23.0/30 area 0
frr-pc(config-router)# network 192.168.3.0/24 area 0
frr-pc(config-router)# network 192.168.33.1/32 area 0
frr-pc(config-router)# end
frr-pc#

```

Figure 40. Configuring OSPF for IPv4 networks in router r3.

Step 8. Type the following command to verify the routing table of router r3.

```

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 192.168.23.1, r3-eth1, 00:08:41
O  192.168.3.0/24 [110/10] is directly connected, r3-eth0, 00:08:05
C>* 192.168.3.0/24 is directly connected, r3-eth0, 02:38:18
O>* 192.168.22.1/32 [110/10] via 192.168.23.1, r3-eth1, 00:08:41
O  192.168.23.0/30 [110/10] is directly connected, r3-eth1, 00:08:44
C>* 192.168.23.0/30 is directly connected, r3-eth1, 02:38:18
O  192.168.33.1/32 [110/0] is directly connected, lo, 00:07:55
C>* 192.168.33.1/32 is directly connected, lo, 02:38:18
frr-pc#

```

Figure 41. Verifying IPv4 routes on router r3.

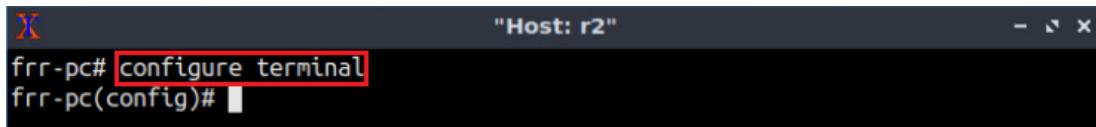
Consider Figure 41. Router r3 reaches the network 192.168.2.0/24 via the IP address 192.168.23.1. Networks 192.168.3.0/24 and 192.168.23.0/30 have two available paths from router r3. The administrative distance (AD) of the paths advertised through OSPF is 110. The AD is a value used by routers to select the best path when there are multiple available routes to the same destination. A smaller AD is always preferable to the routers. The characters **[*]** indicates that the following path is used to reach a specific network. Router r3 prefers directly connected networks over OSPF since the former has a lower AD than the latter.

3.2 Configure OSPFv3 for IPv6 networks

In this section, you will configure OSPFv3 for IPv6 networks to participate in the OSPF process. To do the configuration, you will enable the interface associated with area ID and specify area range parameters on the router. In this lab, you will use area 0.0.0.0 which is default in FRR.

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

```
configure terminal
```

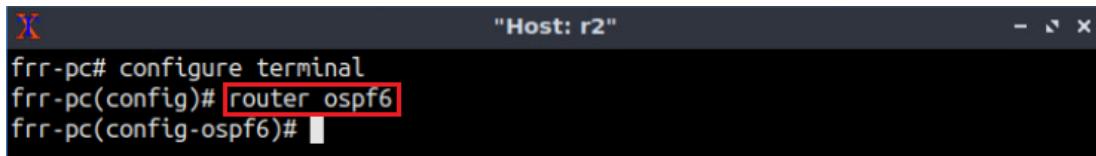


```
"Host: r2"
frr-pc# configure terminal
frr-pc(config)#
```

Figure 42. Enabling configuration mode on router r2.

Step 2. In order to configure OSPF routing protocol, type the command shown below. This command will enable OSPF configuration mode where you can advertise the networks directly connected to the router r2.

```
router ospf6
```

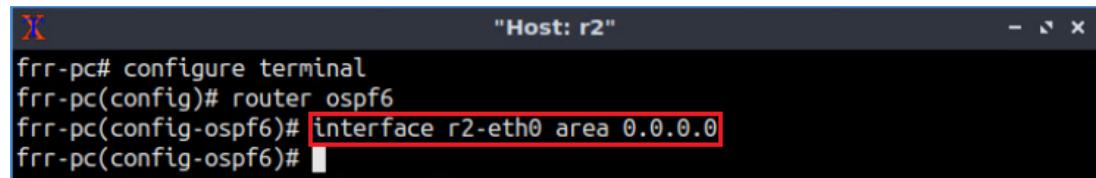


```
"Host: r2"
frr-pc# configure terminal
frr-pc(config)# router ospf6
frr-pc(config-ospf6)#
```

Figure 43. Configuring OSPF for IPv6 networks on router r3.

Step 3. In this step, you will enable the interface of router r2 to participate in the routing process. Type the following command to enable the interface *r2-eth0* along with area 0.0.0.0.

```
interface r2-eth0 area 0.0.0.0
```

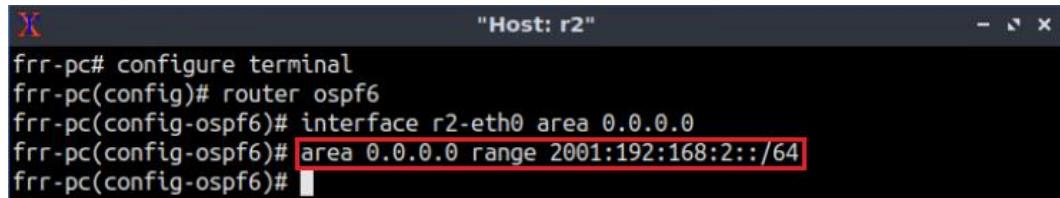


```
"Host: r2"
frr-pc# configure terminal
frr-pc(config)# router ospf6
frr-pc(config-ospf6)# interface r2-eth0 area 0.0.0.0
frr-pc(config-ospf6)#
```

Figure 44. Enabling the interface corresponds to r2-eth0 to participate in the OSPF routing process.

Step 4. In this step, you will specify the area range so that the network 2001:192:168:2::/64 will be advertised through OSPF.

```
area 0.0.0.0 range 2001:192:168:2::/64
```



```
"Host: r2"
frr-pc# configure terminal
frr-pc(config)# router ospf6
frr-pc(config-ospf6)# interface r2-eth0 area 0.0.0.0
frr-pc(config-ospf6)# area 0.0.0.0 range 2001:192:168:2::/64
frr-pc(config-ospf6)#
```

Figure 45. Advertising the network corresponding to r2-eth0 to participate in the OSPF routing process.

Step 5. Type the following command to enable the interface *r2-eth2* along with area 0.0.0.0.

```
interface r2-eth2 area 0.0.0.0
```

```
frr-pc# configure terminal
frr-pc(config)# router ospf6
frr-pc(config-ospf6)# interface r2-eth0 area 0.0.0.0
frr-pc(config-ospf6)# area 0.0.0.0 range 2001:192:168:2::/64
frr-pc(config-ospf6)# interface r2-eth2 area 0.0.0.0
frr-pc(config-ospf6)#

```

Figure 46. Enabling the interface corresponding to *r2-eth2* to participate in the OSPF routing process.

Step 6. Type the following command to advertise the network 2001:192:168:23::/64 through OSPF.

```
area 0.0.0.0 range 2001:192:168:23::/64
```

```
frr-pc# configure terminal
frr-pc(config)# router ospf6
frr-pc(config-ospf6)# interface r2-eth0 area 0.0.0.0
frr-pc(config-ospf6)# area 0.0.0.0 range 2001:192:168:2::/64
frr-pc(config-ospf6)# interface r2-eth2 area 0.0.0.0
frr-pc(config-ospf6)# area 0.0.0.0 range 2001:192:168:23::/64
frr-pc(config-ospf6)#

```

Figure 47. Advertising the network corresponding to *r2-eth2* to participate in the OSPF routing process.

Step 7. Type the following command to enable the interface *lo* along with area 0.0.0.0.

```
interface lo area 0.0.0.0
```

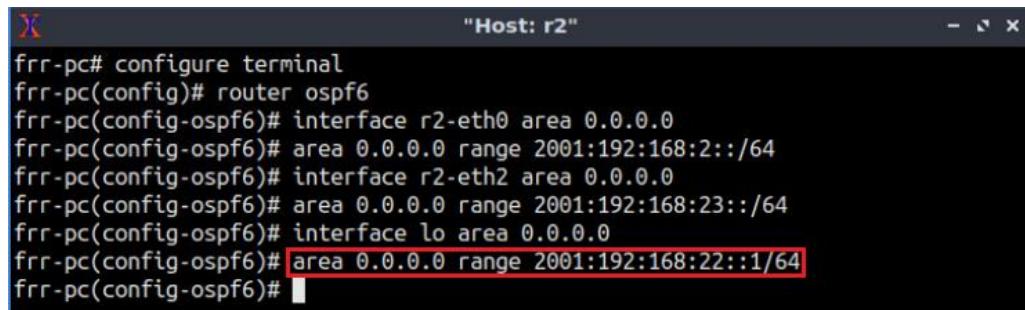
```
frr-pc# configure terminal
frr-pc(config)# router ospf6
frr-pc(config-ospf6)# interface r2-eth0 area 0.0.0.0
frr-pc(config-ospf6)# area 0.0.0.0 range 2001:192:168:2::/64
frr-pc(config-ospf6)# interface r2-eth2 area 0.0.0.0
frr-pc(config-ospf6)# area 0.0.0.0 range 2001:192:168:23::/64
frr-pc(config-ospf6)# interface lo area 0.0.0.0
frr-pc(config-ospf6)#

```

Figure 48. Enabling the interface corresponding to *lo* to participate in the OSPF routing process.

Step 8. Type the following command to advertise the loopback address 2001:192:168:22::1/64 through OSPF.

```
area 0.0.0.0 range 2001:192:168:22::1/64
```



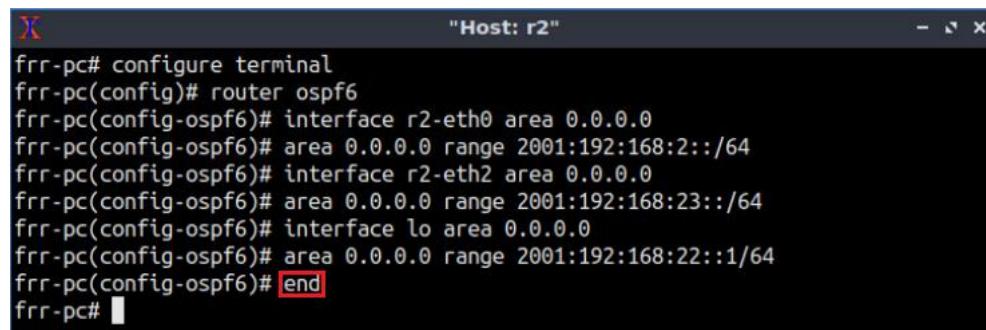
```
"Host: r2"
frr-pc# configure terminal
frr-pc(config)# router ospf6
frr-pc(config-ospf6)# interface r2-eth0 area 0.0.0.0
frr-pc(config-ospf6)# area 0.0.0.0 range 2001:192:168:2::/64
frr-pc(config-ospf6)# interface r2-eth2 area 0.0.0.0
frr-pc(config-ospf6)# area 0.0.0.0 range 2001:192:168:23::/64
frr-pc(config-ospf6)# interface lo area 0.0.0.0
frr-pc(config-ospf6)# area 0.0.0.0 range 2001:192:168:22::1/64
frr-pc(config-ospf6)#

```

Figure 49. Advertising the loopback address to participate in the OSPF routing process.

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

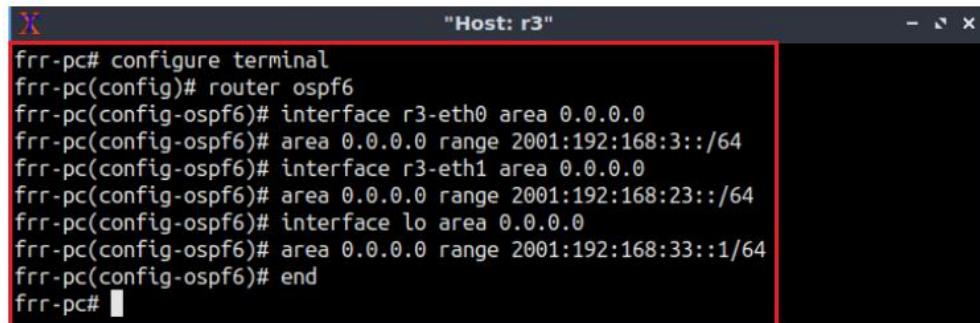
```
end
```



```
"Host: r2"
frr-pc# configure terminal
frr-pc(config)# router ospf6
frr-pc(config-ospf6)# interface r2-eth0 area 0.0.0.0
frr-pc(config-ospf6)# area 0.0.0.0 range 2001:192:168:2::/64
frr-pc(config-ospf6)# interface r2-eth2 area 0.0.0.0
frr-pc(config-ospf6)# area 0.0.0.0 range 2001:192:168:23::/64
frr-pc(config-ospf6)# interface lo area 0.0.0.0
frr-pc(config-ospf6)# area 0.0.0.0 range 2001:192:168:22::1/64
frr-pc(config-ospf6)# end
frr-pc#
```

Figure 50. Exiting from configuration mode.

Step 10. 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, on router r3 terminal issue the commands depicted below.



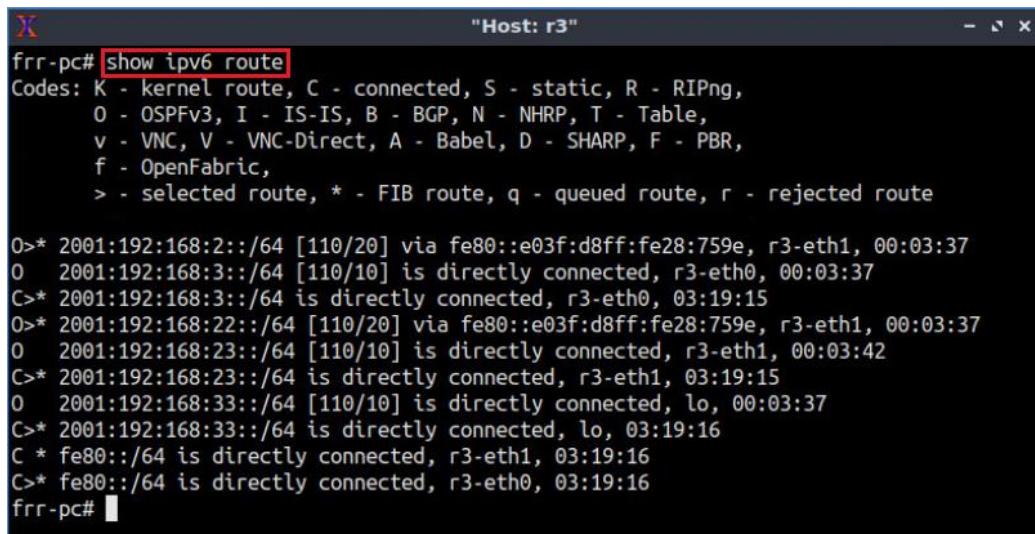
```
"Host: r3"
frr-pc# configure terminal
frr-pc(config)# router ospf6
frr-pc(config-ospf6)# interface r3-eth0 area 0.0.0.0
frr-pc(config-ospf6)# area 0.0.0.0 range 2001:192:168:3::/64
frr-pc(config-ospf6)# interface r3-eth1 area 0.0.0.0
frr-pc(config-ospf6)# area 0.0.0.0 range 2001:192:168:23::/64
frr-pc(config-ospf6)# interface lo area 0.0.0.0
frr-pc(config-ospf6)# area 0.0.0.0 range 2001:192:168:33::1/64
frr-pc(config-ospf6)# end
frr-pc#
```

Figure 51. Configuring OSPFv3 on router r3.

3.3 Verify connectivity between router r2 and router r3

Step 1. Type the following command to verify the routing table of router r3. Router r3 will communicate with the networks attached to router r2 (2001:192:168:2::/64 and 2001:192:168:22::1/64) through OSPF.

```
show ipv6 route
```



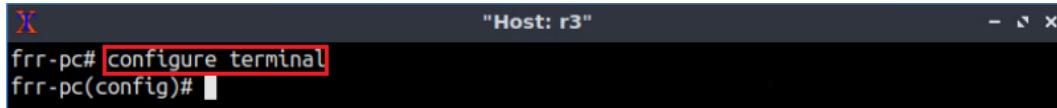
```
"Host: r3"
frr-pc# show ipv6 route
Codes: K - kernel route, C - connected, S - static, R - RIPng,
      O - OSPFv3, I - IS-IS, B - BGP, N - NHRP, T - Table,
      v - VNC, V - VNC-Direct, A - Babel, D - SHARP, F - PBR,
      f - OpenFabric,
      > - selected route, * - FIB route, q - queued route, r - rejected route

O>* 2001:192:168:2::/64 [110/20] via fe80::e03f:d8ff:fe28:759e, r3-eth1, 00:03:37
O  2001:192:168:3::/64 [110/10] is directly connected, r3-eth0, 00:03:37
C>* 2001:192:168:3::/64 is directly connected, r3-eth0, 03:19:15
O>* 2001:192:168:22::/64 [110/20] via fe80::e03f:d8ff:fe28:759e, r3-eth1, 00:03:37
O  2001:192:168:23::/64 [110/10] is directly connected, r3-eth1, 00:03:42
C>* 2001:192:168:23::/64 is directly connected, r3-eth1, 03:19:15
O  2001:192:168:33::/64 [110/10] is directly connected, lo, 00:03:37
C>* 2001:192:168:33::/64 is directly connected, lo, 03:19:16
C * fe80::/64 is directly connected, r3-eth1, 03:19:16
C>* fe80::/64 is directly connected, r3-eth0, 03:19:16
frr-pc#
```

Figure 52. Verifying IPv6 routes on router r3.

Step 2. At this point, router r2 and router r3 will exchange routes. By default, IPv6 forwarding is disabled in FRR. You will enable IPv6 forwarding so that all the IPv6 hosts can participate in the routing process. In order to enable router r3 configuration mode, issue the following command:

```
configure terminal
```

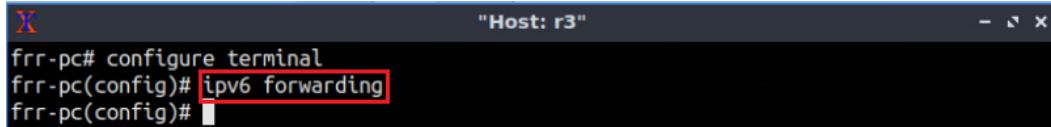


```
"Host: r3"
frr-pc# configure terminal
frr-pc(config)#
```

Figure 53. Enabling configuration mode on router r2.

Step 3. Type the following command to enable IPv6 forwarding.

```
ipv6 forwarding
```

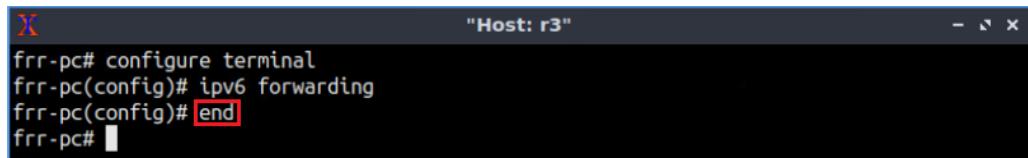


```
"Host: r3"
frr-pc# configure terminal
frr-pc(config)# ipv6 forwarding
frr-pc(config)#
```

Figure 54. Enabling IPv6 forwarding in router r3.

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

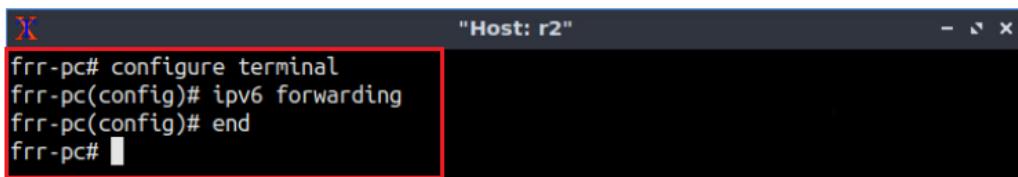
```
end
```



```
"Host: r3"
frr-pc# configure terminal
frr-pc(config)# ipv6 forwarding
frr-pc(config)# end
frr-pc#
```

Figure 55. Exiting from configuration mode.

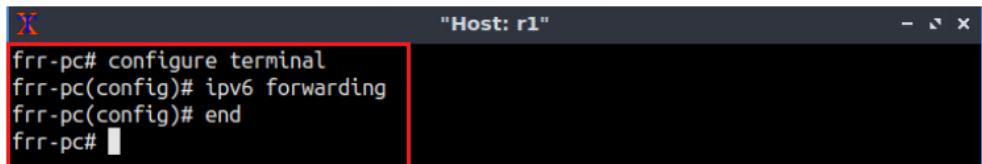
Step 5. Repeat from step 2 to step 4 in order to enable IPv6 forwarding in router r2.



```
frr-pc# configure terminal
frr-pc(config)# ipv6 forwarding
frr-pc(config)# end
frr-pc#
```

Figure 56. Enabling IPv6 forwarding on router r2.

Step 6. Repeat from step 2 to step 4 in order to enable IPv6 forwarding in router r1.

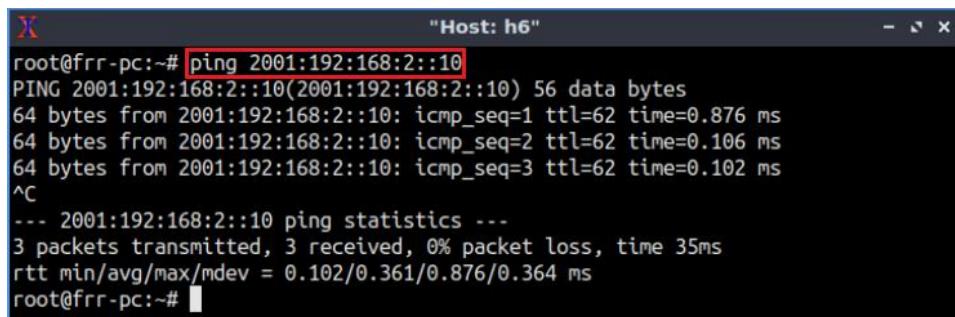


```
frr-pc# configure terminal
frr-pc(config)# ipv6 forwarding
frr-pc(config)# end
frr-pc#
```

Figure 57. Enabling IPv6 forwarding on router r1.

Step 7. On host h6 terminal, perform a connectivity between host h6 and host h5 by issuing the command shown below. To stop the test, press **Ctrl+c**. The result will show a successful connectivity test.

```
ping 2001:192:168:2::10
```



```
root@frr-pc:~# ping 2001:192:168:2::10
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=62 time=0.876 ms
64 bytes from 2001:192:168:2::10: icmp_seq=2 ttl=62 time=0.106 ms
64 bytes from 2001:192:168:2::10: icmp_seq=3 ttl=62 time=0.102 ms
^C
--- 2001:192:168:2::10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 35ms
rtt min/avg/max/mdev = 0.102/0.361/0.876/0.364 ms
root@frr-pc:~#
```

Figure 58. Connectivity test using **ping** command.

4 Configure and verify BGP for IPv4 networks

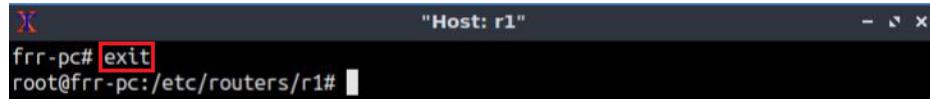
In this section, you will configure BGP on all routers. Routers r2 and r3 communicate with router r1 through EBGP, while router r2 communicates with router r3 through IBGP. You will assign BGP neighbors to allow the routers to exchange BGP routes. Furthermore, routers r1, r2, and r3 will advertise their LANs via BGP so that the LANs are learned by peer routers.

You will configure EBGP so that router r1 uses IPv4 as the BGP transport for IPv4 sessions. For IBGP, you will configure router r2 so that IPv4 routing information is transported by IPv4 TCP sessions.

4.1 Configure and verify EBGP on router r1

Step 1. To configure BGP routing protocol, you need to enable the BGP daemon first. On router r1 terminal, type the following command to exit the vtysh session:

```
exit
```

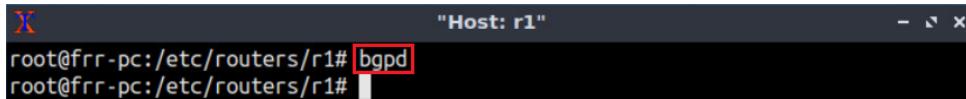


```
frr-pc# exit
root@frr-pc:/etc/routers/r1#
```

Figure 59. Exiting the vtysh session.

Step 2. Type the following command on r1 terminal to enable and start BGP routing protocol.

```
bgpd
```

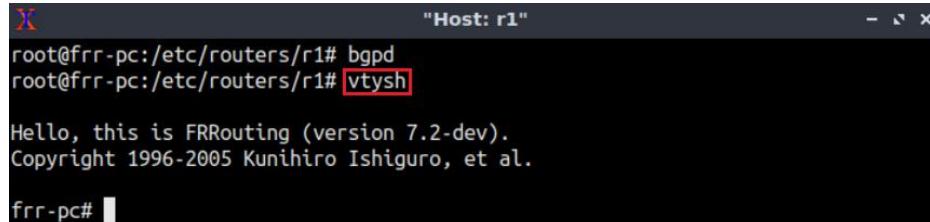


```
root@frr-pc:/etc/routers/r1# bgpd
root@frr-pc:/etc/routers/r1#
```

Figure 60. Starting BGP daemon.

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

```
vtysh
```



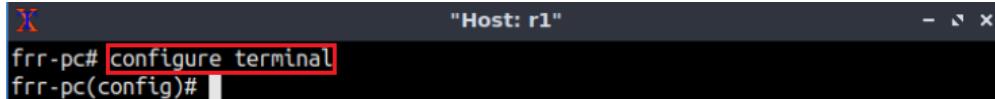
```
root@frr-pc:/etc/routers/r1# bgpd
root@frr-pc:/etc/routers/r1# vtysh
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc#
```

Figure 61. Starting vtysh on router r1.

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

```
configure terminal
```

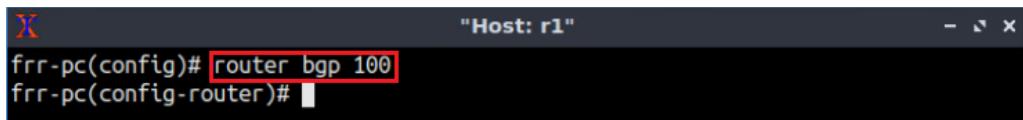


```
frr-pc# configure terminal
frr-pc(config)#
```

Figure 62. Enabling configuration mode on router r1.

Step 5. The ASN assigned for router r1 is 100. In order to apply the configuration, type the following command:

```
router bgp 100
```

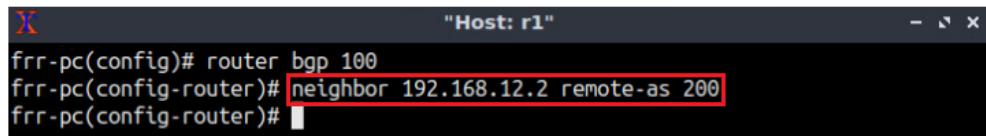


```
"Host: r1"
frr-pc(config)# router bgp 100
frr-pc(config-router)#[ ]
```

Figure 63. Configuring BGP on router r1.

Step 6. To configure a BGP neighbor to router r1 (AS 100), type the command shown below. This command specifies the neighbor IP address (192.168.12.2) and the ASN of the remote BGP peer (AS 200). This neighbor will act as the BGP transport for both IPv4 and IPv6 networks.

```
neighbor 192.168.12.2 remote-as 200
```

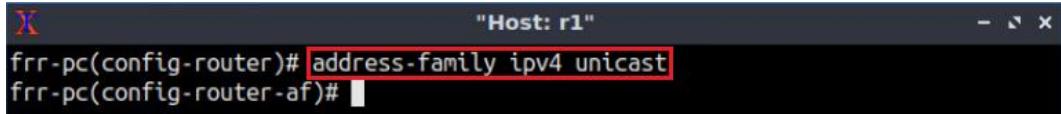


```
"Host: r1"
frr-pc(config)# router bgp 100
frr-pc(config-router)# neighbor 192.168.12.2 remote-as 200
frr-pc(config-router)#[ ]
```

Figure 64. Assigning BGP neighbor to router r1.

Step 7. Type the following command to enter address-family mode where you can configure routing sessions that use standard IPv4 address prefixes.

```
address-family ipv4 unicast
```

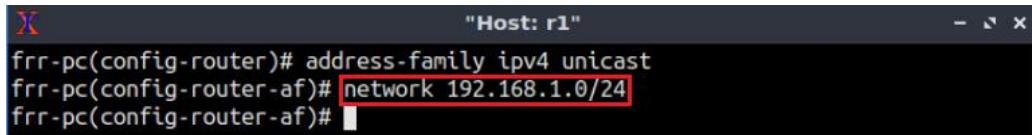


```
"Host: r1"
frr-pc(config-router)# address-family ipv4 unicast
frr-pc(config-router-af)#[ ]
```

Figure 65. Enabling address-family IPv4 configuration mode on router r1.

Step 8. In this step, router r1 will advertise the LAN 192.168.1.0/24 to its BGP peers. To do so, issue the following command:

```
network 192.168.1.0/24
```

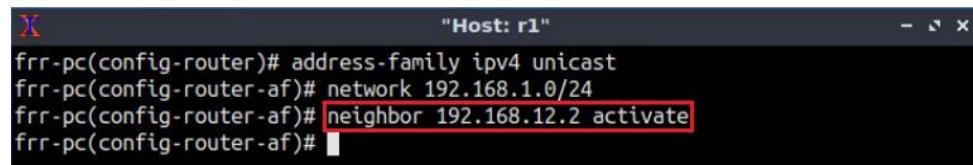


```
"Host: r1"
frr-pc(config-router)# address-family ipv4 unicast
frr-pc(config-router-af)# network 192.168.1.0/24
frr-pc(config-router-af)#[ ]
```

Figure 66. Advertising IPv4 LAN on router r1.

Step 9. Type the following command to activate the neighbor 192.168.12.2 so that router r1 uses this neighbor to advertise the IPv4 LAN.

```
neighbor 192.168.12.2 activate
```

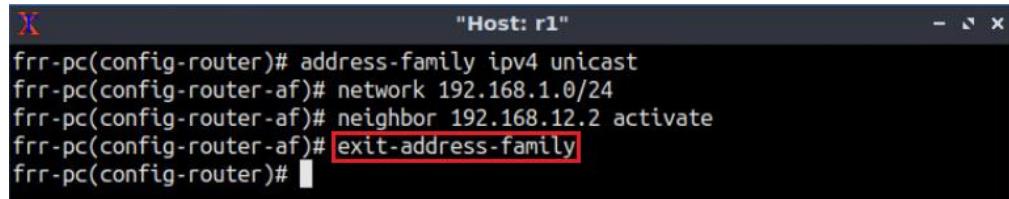


```
frr-pc(config-router)# address-family ipv4 unicast
frr-pc(config-router-af)# network 192.168.1.0/24
frr-pc(config-router-af)# neighbor 192.168.12.2 activate
frr-pc(config-router-af)#[ ]
```

Figure 67. Activating neighbor to advertise IPv4 network.

Step 10. Type the following command to exit from the address-family mode.

```
exit-address-family
```

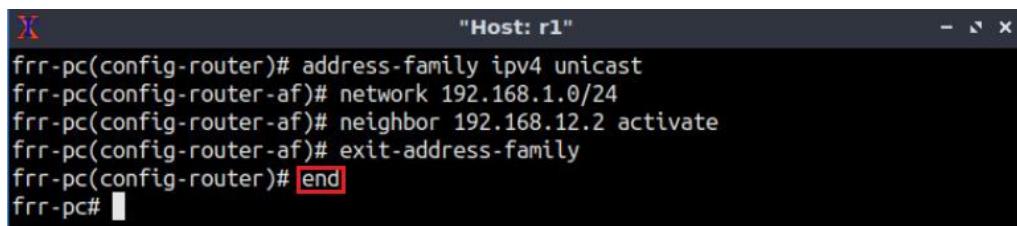


```
frr-pc(config-router)# address-family ipv4 unicast
frr-pc(config-router-af)# network 192.168.1.0/24
frr-pc(config-router-af)# neighbor 192.168.12.2 activate
frr-pc(config-router-af)# exit-address-family
frr-pc(config-router)#[ ]
```

Figure 68. Exiting from address-family mode.

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

```
end
```

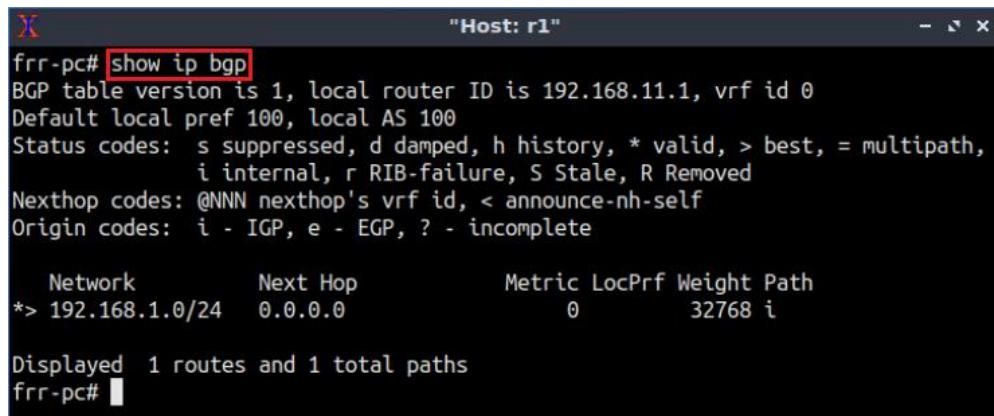


```
frr-pc(config-router)# address-family ipv4 unicast
frr-pc(config-router-af)# network 192.168.1.0/24
frr-pc(config-router-af)# neighbor 192.168.12.2 activate
frr-pc(config-router-af)# exit-address-family
frr-pc(config-router)# end
frr-pc#[ ]
```

Figure 69. Exiting from configuration mode.

Step 12. Type the following command to verify BGP networks. You will observe the LAN network of router r1.

```
show ip bgp
```



```
frr-pc# show ip bgp
BGP table version is 1, local router ID is 192.168.11.1, vrf id 0
Default local pref 100, local AS 100
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
* 192.168.1.0/24    0.0.0.0                  0        32768 i

Displayed 1 routes and 1 total paths
frr-pc# [ ]
```

Figure 70. Verifying BGP networks on router r1.

Step 13. Type the following command on router r1 to verify IPv4 peering information with router r2. You will notice that BGP connectivity for IPv4 is over an IPv4 BGP transport session, using the neighbor address 192.168.12.2.

```
show bgp ipv4 summary
```

```
"Host: r1"
frr-pc# show bgp ipv4 summary

IPv4 Unicast Summary:
BGP router identifier 192.168.11.1, local AS number 100 vrf-id 0
BGP table version 5
RIB entries 5, using 920 bytes of memory
Peers 1, using 21 KiB of memory

Neighbor      V      AS MsgRcvd MsgSent   TblVer InQ OutQ Up/Down State/P
fxRcd
192.168.12.2  4      200      78      77          0     0    0 00:38:16
2

Total number of neighbors 1
frr-pc#
```

Figure 71. Verifying IPv4 BGP summary on router r1.

4.2 Configure and verify EBGP and IBGP on router r2

Step 1. On router r2, exit vtysh session and enable the BGP daemon. Enable router into configuration mode to configure BGP on router r2. All the steps are summarized in the following figure.

```
"Host: r2"
frr-pc# exit
root@frr-pc:/etc/routers/r2# bgpd
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 72. Starting BGP daemon on router r2.

Step 2. The ASN assigned for router r2 is 200. In order to configure BGP, type the following command:

```
router bgp 200
```

```
"Host: r2"
frr-pc(config)# router bgp 200
frr-pc(config-router)#
```

Figure 73. Configuring BGP on router r2.

Step 3. To configure EBGP neighbor to router r2 (AS 200), type the command shown below. This command specifies the neighbor IP address (192.168.12.1) and the ASN of the remote BGP peer (AS 100). This neighbor will act as the BGP transport for both IPv4 and IPv6 networks.

```
neighbor 192.168.12.1 remote-as 100
```

```
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.12.1 remote-as 100
frr-pc(config-router)#
```

Figure 74. Assigning EBGP neighbor to router r2.

Step 4. Type the following command to assign the IPv4 neighbor so that IPv4 network uses IPv4 BGP transport while communicating with router r3. For IBGP peering between router r2 and router r3, assign the loopback address of router r3 as the neighbor of router r2.

```
neighbor 192.168.33.1 remote-as 200
```

```
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.12.1 remote-as 100
frr-pc(config-router)# neighbor 192.168.33.1 remote-as 200
frr-pc(config-router)#
```

Figure 75. Assigning IBGP neighbor to router r2 for IPv4 network.

Step 5. Type the following command to assign *lo* as the source IP in router r2.

```
neighbor 192.168.33.1 update-source lo
```

```
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.12.1 remote-as 100
frr-pc(config-router)# neighbor 192.168.33.1 remote-as 200
frr-pc(config-router)# neighbor 192.168.33.1 update-source lo
frr-pc(config-router)#
```

Figure 76. Assigning loopback as source IP for the neighbor 192.168.33.1.

Step 6. Type the following command to enter address-family mode where you can configure routing sessions that use standard IPv4 address prefixes.

```
address-family ipv4 unicast
```

```
frr-pc(config-router)# address-family ipv4 unicast
frr-pc(config-router-af)#
```

Figure 77. Enabling address-family IPv4 configuration mode on router r2.

Step 7. In this step, router r2 will advertise the LAN 192.168.2.0/24 to its BGP peers. To do so, issue the following command:

```
network 192.168.2.0/24
```

```
frr-pc(config-router)# address-family ipv4 unicast
frr-pc(config-router-af)# network 192.168.2.0/24
frr-pc(config-router-af)#

```

Figure 78. Advertising IPv4 LAN on router r2.

Step 8. Type the following command to activate the neighbor 192.168.12.1 so that this neighbor is used to exchange IPv4 routes with router r1.

```
neighbor 192.168.12.1 activate
```

```
frr-pc(config-router)# address-family ipv4 unicast
frr-pc(config-router-af)# network 192.168.2.0/24
frr-pc(config-router-af)# neighbor 192.168.12.1 activate
frr-pc(config-router-af)#

```

Figure 79. Activating EBGP neighbor to advertise IPv4 network.

Step 9. Type the following command to activate the neighbor 192.168.33.1 so that this neighbor is used to exchange IPv4 routes with router r3.

```
neighbor 192.168.33.1 activate
```

```
frr-pc(config-router)# address-family ipv4 unicast
frr-pc(config-router-af)# network 192.168.2.0/24
frr-pc(config-router-af)# neighbor 192.168.12.1 activate
frr-pc(config-router-af)# neighbor 192.168.33.1 activate
frr-pc(config-router-af)#

```

Figure 80. Activating IBGP neighbor to advertise IPv4 network.

Step 10. Type the following command on router r2 so that the interface lo is used as the next hop address of router r2. It will allow router r3 to receive the route to router r1 as the next hop address (192.168.22.1) is known to router r3.

```
neighbor 192.168.33.1 next-hop-self
```

```
frr-pc(config-router)# address-family ipv4 unicast
frr-pc(config-router-af)# network 192.168.2.0/24
frr-pc(config-router-af)# neighbor 192.168.12.1 activate
frr-pc(config-router-af)# neighbor 192.168.33.1 activate
frr-pc(config-router-af)# neighbor 192.168.33.1 next-hop-self
frr-pc(config-router-af)#

```

Figure 81. Assigning next hop address on router r2.

Step 11. Type the following command to exit from the address-family mode.

```
exit-address-family
```

```
"Host: r2"
frr-pc(config-router)# address-family ipv4 unicast
frr-pc(config-router-af)# network 192.168.2.0/24
frr-pc(config-router-af)# neighbor 192.168.12.1 activate
frr-pc(config-router-af)# neighbor 192.168.33.1 activate
frr-pc(config-router-af)# neighbor 192.168.33.1 next-hop-self
frr-pc(config-router-af)# exit-address-family
frr-pc(config-router)#[ ]
```

Figure 82. Exiting from address-family mode.

Step 12. Type the following command to exit from configuration mode.

```
end
```

```
"Host: r2"
frr-pc(config-router)# address-family ipv4 unicast
frr-pc(config-router-af)# network 192.168.2.0/24
frr-pc(config-router-af)# neighbor 192.168.12.1 activate
frr-pc(config-router-af)# neighbor 192.168.33.1 activate
frr-pc(config-router-af)# neighbor 192.168.33.1 next-hop-self
frr-pc(config-router-af)# exit-address-family
frr-pc(config-router)# end
frr-pc#[ ]
```

Figure 83. Exiting from configuration mode.

Step 13. Type the following command on router r2 to verify IPv4 peering information with routers r1 and r3. You will notice that BGP connectivity for IPv4 is over IPv4 BGP transport session, using the neighbor address 192.168.12.1 and 192.168.33.1 respectively.

```
show bgp ipv4 summary
```

```
"Host: r2"
frr-pc# show bgp ipv4 summary

IPv4 Unicast Summary:
BGP router identifier 192.168.22.1, local AS number 200 vrf-id 0
BGP table version 2
RIB entries 3, using 552 bytes of memory
Peers 2, using 41 KiB of memory

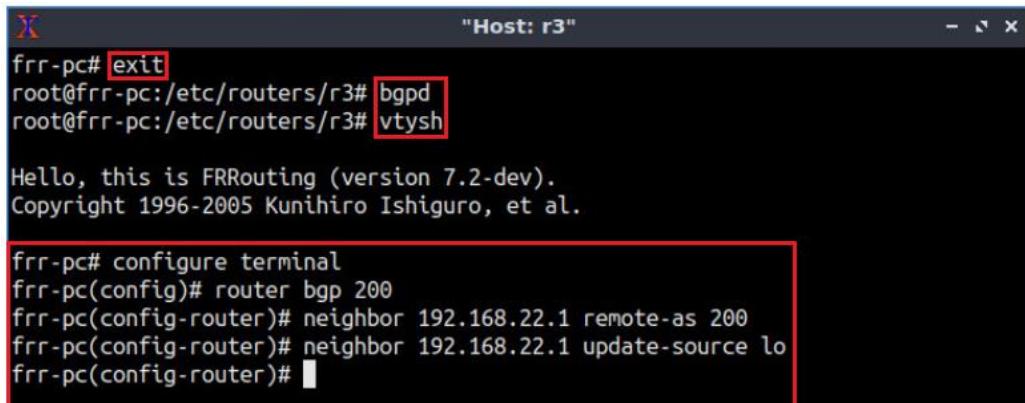
Neighbor      V      AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down State/P
fxRcd
192.168.12.1  4      100    13     13       0      0      0  00:08:16
1
192.168.33.1  4      200    0      0       0      0      0  never      A
ctive

Total number of neighbors 2
frr-pc#[ ]
```

Figure 84. Verifying IPv4 BGP summary on router r2.

4.3 Configure and verify IBGP on router r3

Step 1. Router r3 is configured similarly to router r2 but, with different metrics in order to establish IBGP peering with router r2. All the steps are summarized in the following figure.



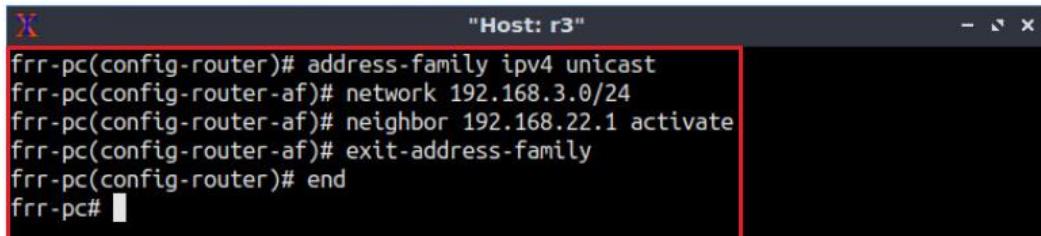
```
frr-pc# exit
root@frr-pc:/etc/routers/r3# bgpd
root@frr-pc:/etc/routers/r3# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.22.1 remote-as 200
frr-pc(config-router)# neighbor 192.168.22.1 update-source lo
frr-pc(config-router)#
```

Figure 85. Configuring BGP on router r3.

Step 2. Configure IPv4 address-family on router r3. There is no need to assign the next hop when configuring BGP, since router r3 is not participating in any EBGP session. All the steps are summarized in the following figure.

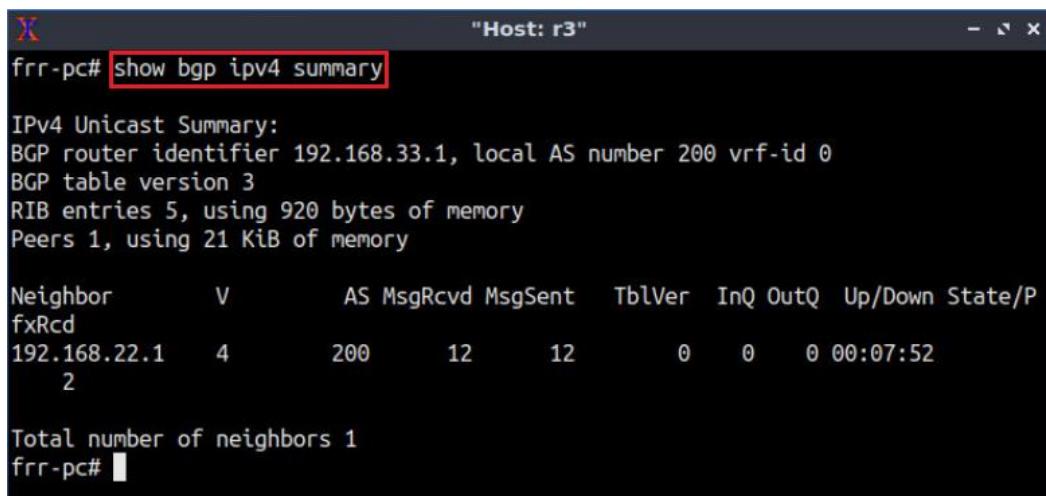


```
frr-pc(config-router)# address-family ipv4 unicast
frr-pc(config-router-af)# network 192.168.3.0/24
frr-pc(config-router-af)# neighbor 192.168.22.1 activate
frr-pc(config-router-af)# exit-address-family
frr-pc(config-router)# end
frr-pc#
```

Figure 86. Configuring IPv4 address-family on router r3.

Step 3. Type the following command on router r3 to verify IPv4 peering information with router r2. You will notice that BGP connectivity for IPv4 is over an IPv4 BGP transport session, using the neighbor address 192.168.22.1.

```
show bgp ipv4 summary
```



```
"Host: r3"
frr-pc# show bgp ipv4 summary

IPv4 Unicast Summary:
BGP router identifier 192.168.33.1, local AS number 200 vrf-id 0
BGP table version 3
RIB entries 5, using 920 bytes of memory
Peers 1, using 21 KiB of memory

Neighbor      V      AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down State/P
fxRcd
192.168.22.1  4      200       12       12       0       0     0 00:07:52
2

Total number of neighbors 1
frr-pc#
```

Figure 87. Verifying IPv4 BGP summary on router r3.

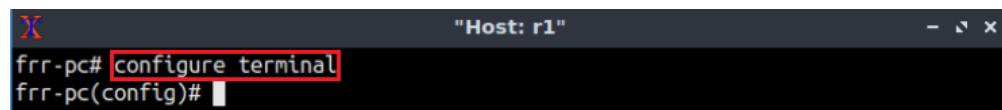
5 Configure and verify BGP for IPv6 networks

In this section, you will configure EBGP so that router r1 uses IPv4 as the BGP transport for IPv6 sessions. Create a route-map next-hop-IPv6 to attach to the BGP neighbor in the outbound direction so that the next-hop parameter overwrites with the appropriate IPv6 next-hop address. For IBGP, you will configure router r2 so that IPv6 routing information is transported by IPv6 TCP sessions.

5.1 Configure and verify EBGP on router r1

Step 1. To enable router r1 into configuration mode, issue the following command:

```
configure terminal
```

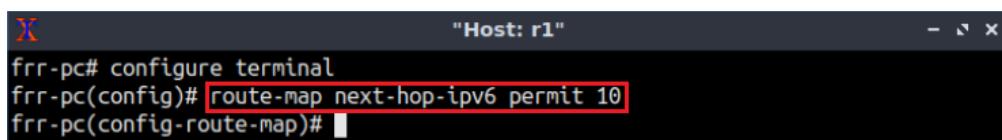


```
"Host: r1"
frr-pc# configure terminal
frr-pc(config)#
```

Figure 88. Enabling configuration mode on router r1.

Step 2. Type the following command to create a route-map named next-hop-IPv6 with permit clause. The permit clause will allow BGP to use the route map policy. The sequence number allows the identification and editing of multiple statements. You will use default sequence number which is 10. You will be entering the configuration mode where you can set the route-map policy.

```
route-map next-hop-ipv6 permit 10
```

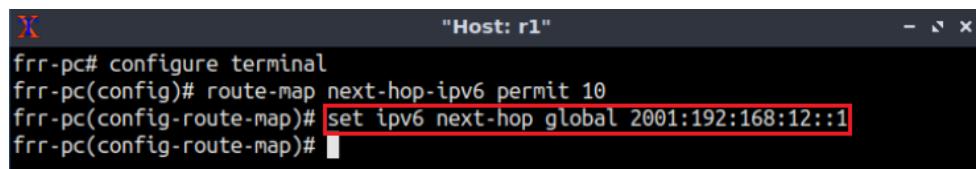


```
"Host: r1"
frr-pc# configure terminal
frr-pc(config)# route-map next-hop-ipv6 permit 10
frr-pc(config-route-map)#
```

Figure 89. Creating next-hop-IPv6 route-map.

Step 3. Type the following command to set the route-map policy. As the IPv6 address will be transported by IPv4 TCP session, you need an IPv6 address so that the next-hop parameter overwrites with the appropriate IPv6 next-hop address. By the route-map policy, you will set the global IPv6 address 2001:192:168:12::1 for router r1 which will be the next hop address for router r2 and the link local address will appear as the next hop address in the BGP table of router r2.

```
set ipv6 next-hop global 2001:192:168:12::1
```



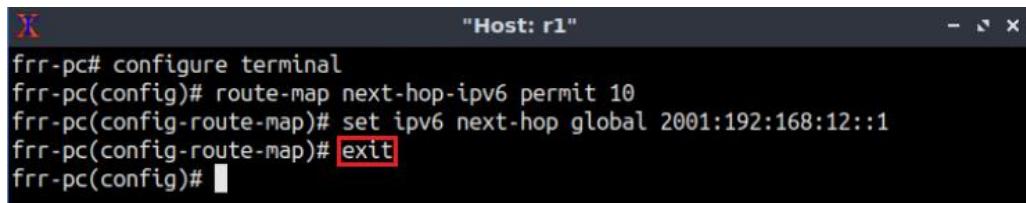
A terminal window titled "Host: r1" showing the configuration of a route-map. The commands entered are:

```
frr-pc# configure terminal
frr-pc(config)# route-map next-hop-ipv6 permit 10
frr-pc(config-route-map)# set ipv6 next-hop global 2001:192:168:12::1
frr-pc(config-route-map)#
```

Figure 90. Setting route-map policy in router r1.

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

```
exit
```



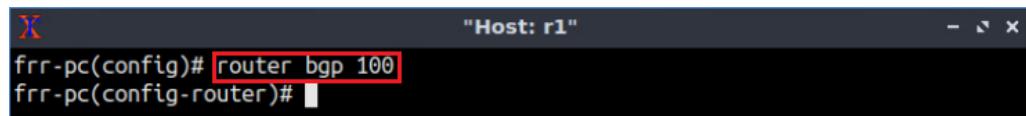
A terminal window titled "Host: r1" showing the exit from route-map mode. The commands entered are:

```
frr-pc# configure terminal
frr-pc(config)# route-map next-hop-ipv6 permit 10
frr-pc(config-route-map)# set ipv6 next-hop global 2001:192:168:12::1
frr-pc(config-route-map)# exit
frr-pc(config)#
```

Figure 91. Exiting from route-map mode.

Step 5. The ASN assigned for router r1 is 100. In order to apply the configuration, type the following command:

```
router bgp 100
```



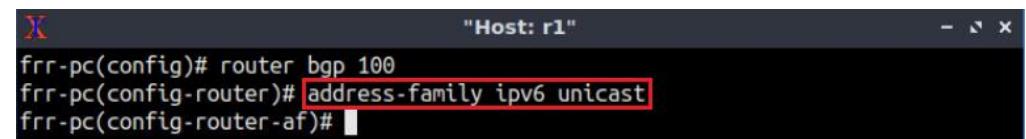
A terminal window titled "Host: r1" showing the configuration of BGP. The command entered is:

```
frr-pc(config)# router bgp 100
frr-pc(config-router)#
```

Figure 92. Configuring BGP on router r1.

Step 6. Type the following command to enter address-family mode where you can configure routing sessions that use standard IPv6 address prefixes.

```
address-family ipv6 unicast
```



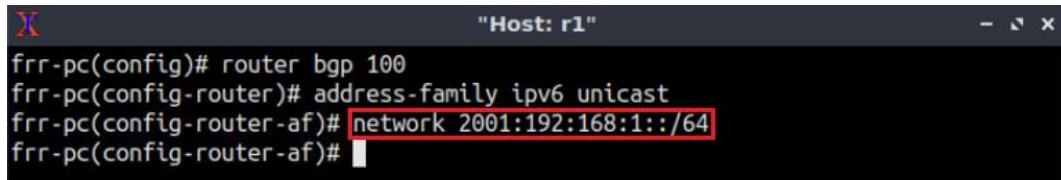
A terminal window titled "Host: r1" showing the enabling of address-family IPv6 configuration mode. The command entered is:

```
frr-pc(config)# router bgp 100
frr-pc(config-router)# address-family ipv6 unicast
frr-pc(config-router-af)#
```

Figure 93. Enabling address-family IPv6 configuration mode on router r1.

Step 7. In this step, router r1 will advertise the IPv6 LAN 2001:192:168:1::/64 to its BGP peers. To do so, issue the following command:

```
network 2001:192:168:1::/64
```



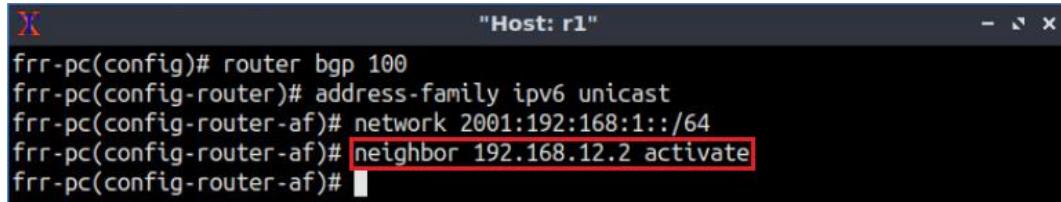
```
"Host: r1"
frr-pc(config)# router bgp 100
frr-pc(config-router)# address-family ipv6 unicast
frr-pc(config-router-af)# network 2001:192:168:1::/64
frr-pc(config-router-af)#

```

Figure 94. Advertising IPv6 LAN on router r1.

Step 8. Since you are using IPv4 neighbor as BGP transport, you will activate the IPv4 neighbor within the IPv6 address-family. To do so, type the following command.

```
neighbor 192.168.12.2 activate
```



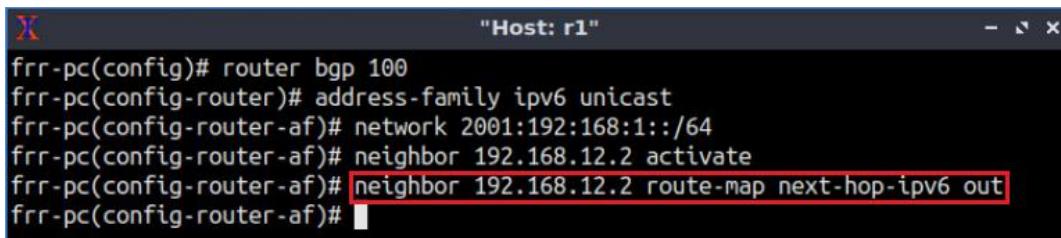
```
"Host: r1"
frr-pc(config)# router bgp 100
frr-pc(config-router)# address-family ipv6 unicast
frr-pc(config-router-af)# network 2001:192:168:1::/64
frr-pc(config-router-af)# neighbor 192.168.12.2 activate
frr-pc(config-router-af)#

```

Figure 95. Activating neighbor to advertise IPv6 network.

Step 9. Type the following command to attach the route-map to BGP neighbor in the outbound direction. Outbound direction means that this information in the route-map will be applied to IPv6 BGP updates as they are sent to router r2. In the BGP table of router r2, 2001:192:168:12::1 will be used as next-hop address. The next-hop address will be the link local address of 2001:192:168:12::1 because link local addresses are used as next-hop address in FRR.

```
neighbor 192.168.12.2 route-map next-hop-ipv6 out
```



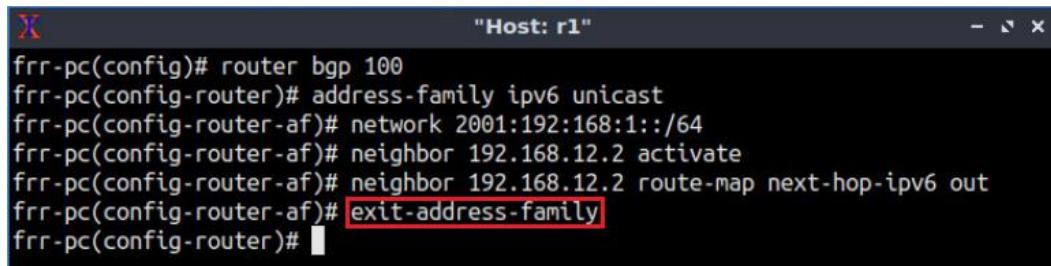
```
"Host: r1"
frr-pc(config)# router bgp 100
frr-pc(config-router)# address-family ipv6 unicast
frr-pc(config-router-af)# network 2001:192:168:1::/64
frr-pc(config-router-af)# neighbor 192.168.12.2 activate
frr-pc(config-router-af)# neighbor 192.168.12.2 route-map next-hop-ipv6 out
frr-pc(config-router-af)#

```

Figure 96. Attaching the route-map to BGP neighbor.

Step 10. Type the following command to exit from the address-family mode.

```
exit-address-family
```



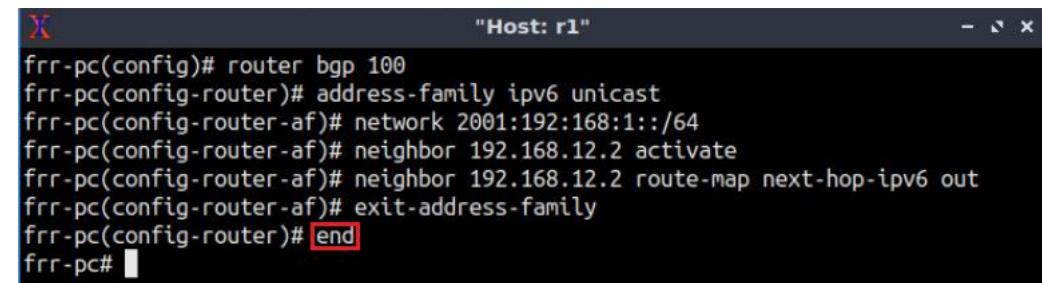
```
"Host: r1"
frr-pc(config)# router bgp 100
frr-pc(config-router)# address-family ipv6 unicast
frr-pc(config-router-af)# network 2001:192:168:1::/64
frr-pc(config-router-af)# neighbor 192.168.12.2 activate
frr-pc(config-router-af)# neighbor 192.168.12.2 route-map next-hop-ipv6 out
frr-pc(config-router-af)# exit-address-family
frr-pc(config-router)#

```

Figure 97. Exiting from address-family mode.

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

```
end
```

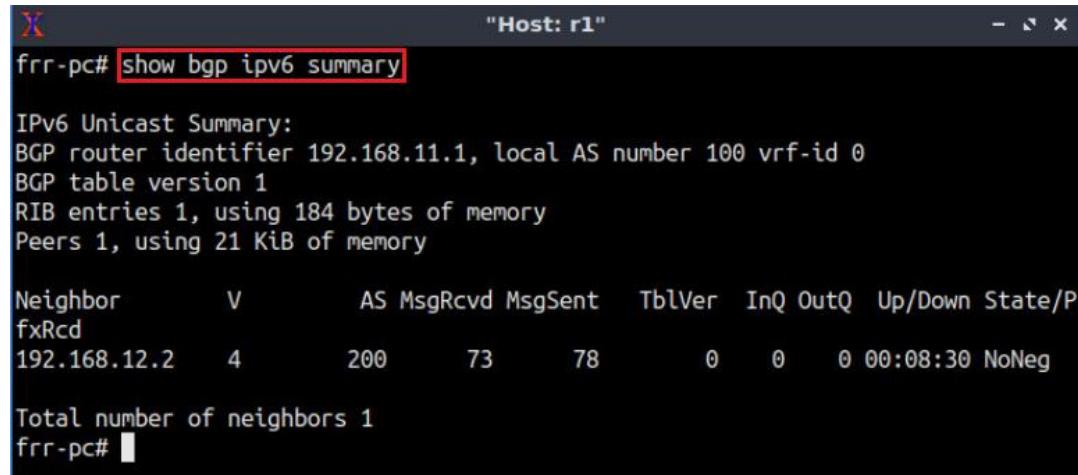


```
"Host: r1"
frr-pc(config)# router bgp 100
frr-pc(config-router)# address-family ipv6 unicast
frr-pc(config-router-af)# network 2001:192:168:1::/64
frr-pc(config-router-af)# neighbor 192.168.12.2 activate
frr-pc(config-router-af)# neighbor 192.168.12.2 route-map next-hop-ipv6 out
frr-pc(config-router-af)# exit-address-family
frr-pc(config-router)# end
frr-pc#
```

Figure 98. Exiting from configuration mode.

Step 12. Type the following command on router r1 to verify IPv6 peering information with router r2. You will notice that BGP connectivity for IPv6 is over an IPv4 BGP transport session, using the neighbor address 192.168.12.2.

```
show bgp ipv6 summary
```



```
"Host: r1"
frr-pc# show bgp ipv6 summary

IPv6 Unicast Summary:
BGP router identifier 192.168.11.1, local AS number 100 vrf-id 0
BGP table version 1
RIB entries 1, using 184 bytes of memory
Peers 1, using 21 KiB of memory

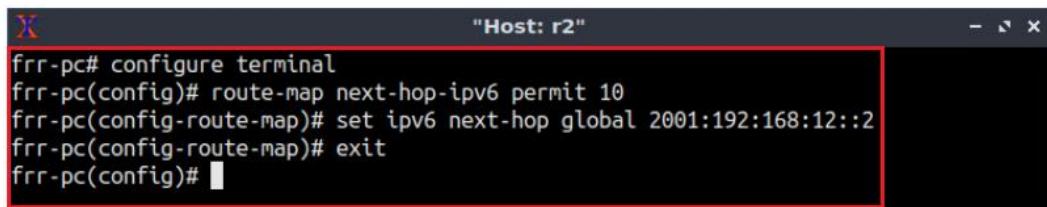
Neighbor      V      AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down State/P
fxRcd
192.168.12.2  4      200     73      78       0      0      0 00:08:30 NoNeg

Total number of neighbors 1
frr-pc#
```

Figure 99. Verifying IPv6 BGP summary on router r1.

5.2 Configure and verify EBGP and IBGP on router r2

Step 1. Enable BGP daemon and create a route-map so that you can attach the route-map to the BGP neighbor of router r2. All the steps are summarized in the figure below.

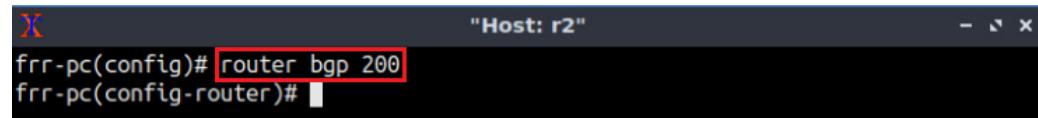


```
"Host: r2"
frr-pc# configure terminal
frr-pc(config)# route-map next-hop-ipv6 permit 10
frr-pc(config-route-map)# set ipv6 next-hop global 2001:192:168:12::2
frr-pc(config-route-map)# exit
frr-pc(config)#
```

Figure 100. Creating next-hop-IPv6 route-map on router r2.

Step 2. The ASN assigned for router r2 is 200. In order to configure BGP, type the following command:

```
router bgp 200
```

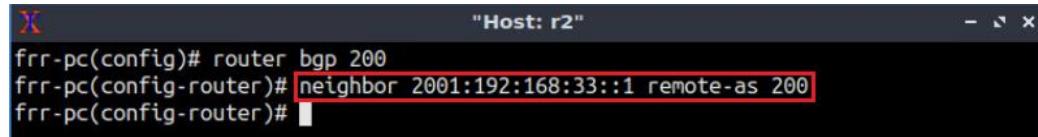


```
"Host: r2"
frr-pc(config)# router bgp 200
frr-pc(config-router)#
```

Figure 101. Configuring BGP on router r2.

Step 3. In this step, you will configure IBGP neighbor to router r2. Type the following command to assign the IPv6 neighbor so that IPv6 network uses IPv6 BGP transport. For IBGP peering between router r2 and router r3, assign the loopback address of router r3 as the neighbor of router r2.

```
neighbor 2001:192:168:33::1 remote-as 200
```

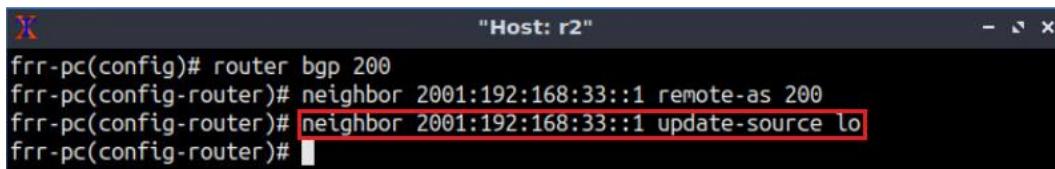


```
"Host: r2"
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 2001:192:168:33::1 remote-as 200
frr-pc(config-router)#
```

Figure 102. Assigning IBGP neighbor to router r2 for IPv6 network.

Step 4. In BGP, the source IP address of BGP packets sent by the router must be the same as neighbor IP address set on the neighboring router. As you are assigning the loopback as neighbor address, you must use loopback address as the source of BGP packets sent to the neighbor. Type the following command to assign *lo* as source IP in router r2.

```
neighbor 2001:192:168:33::1 update-source lo
```

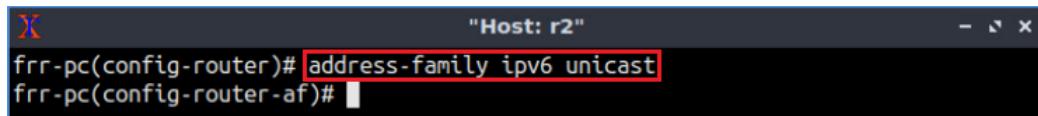


```
"Host: r2"
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 2001:192:168:33::1 remote-as 200
frr-pc(config-router)# neighbor 2001:192:168:33::1 update-source lo
frr-pc(config-router)#
```

Figure 103. Assigning loopback as source IP for the neighbor 2001:192:168:33::1.

Step 5. Type the following command to enter address-family mode where you can configure routing sessions that use standard IPv6 address prefixes.

```
address-family ipv6 unicast
```

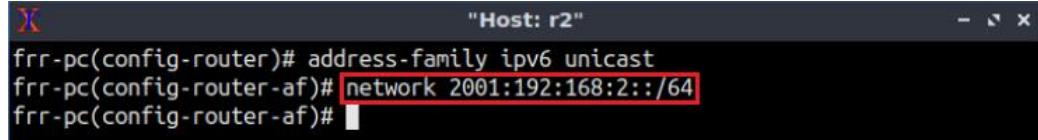


```
"Host: r2"
frr-pc(config-router)# address-family ipv6 unicast
frr-pc(config-router-af)#[ ]
```

Figure 104. Enabling address-family IPv6 configuration mode on router r2.

Step 6. In this step, router r2 will advertise the LAN 2001:192:168:2::/64 to its BGP peers. To do so, issue the following command:

```
network 2001:192:168:2::/64
```

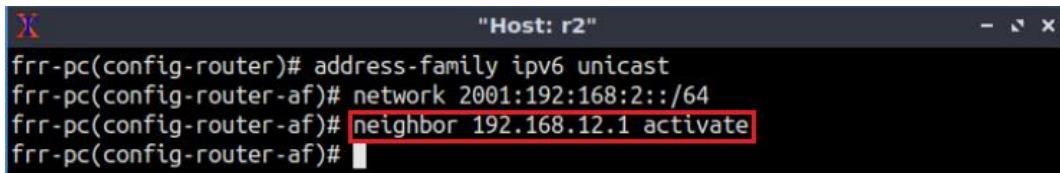


```
"Host: r2"
frr-pc(config-router)# address-family ipv6 unicast
frr-pc(config-router-af)# network 2001:192:168:2::/64
frr-pc(config-router-af)#[ ]
```

Figure 105. Advertising IPv6 LAN on router r2.

Step 7. Type the following command to activate the neighbor 192.168.12.1 so that router r2 uses this neighbor to exchange IPv6 routes with router r1.

```
neighbor 192.168.12.1 activate
```

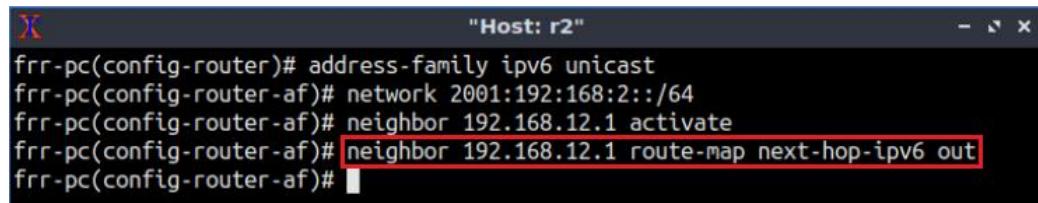


```
"Host: r2"
frr-pc(config-router)# address-family ipv6 unicast
frr-pc(config-router-af)# network 2001:192:168:2::/64
frr-pc(config-router-af)# neighbor 192.168.12.1 activate
frr-pc(config-router-af)#[ ]
```

Figure 106. Activating neighbor to advertise IPv6 network.

Step 8. Type the following command to attach the route-map to BGP neighbor in the outbound direction. Outbound direction means that this information in the route-map will be applied to IPv6 BGP updates as they are sent to router r1. In the BGP table of router r1, 2001:192:168:12::2 will be used as next-hop address. The next-hop address will be the link local address of 2001:192:168:12::2 because FRR always uses link local address as next-hop address.

```
neighbor 192.168.12.1 route-map next-hop-ipv6 out
```



```
"Host: r2"
frr-pc(config-router)# address-family ipv6 unicast
frr-pc(config-router-af)# network 2001:192:168:2::/64
frr-pc(config-router-af)# neighbor 192.168.12.1 activate
frr-pc(config-router-af)# neighbor 192.168.12.1 route-map next-hop-ipv6 out
frr-pc(config-router-af)#[ ]
```

Figure 107. Attaching the route-map to BGP neighbor of router r2.

Step 9. Type the following command to activate the neighbor 2001:192:168:33::1 so that router r2 uses this neighbor to exchange IPv6 routes with router r3.

```
neighbor 2001:192:168:33::1 activate
```

```
frr-pc(config-router)# address-family ipv6 unicast
frr-pc(config-router-af)# network 2001:192:168:2::/64
frr-pc(config-router-af)# neighbor 192.168.12.1 activate
frr-pc(config-router-af)# neighbor 192.168.12.1 route-map next-hop-ipv6 out
frr-pc(config-router-af)# neighbor 2001:192:168:33::1 activate
frr-pc(config-router-af)#[ ]
```

Figure 108. Activating neighbor to advertise IPv6 network.

Step 10. Type the following command on router r2 so that the interface lo is used as the next hop address of router r2. It will allow router r3 to receive the route to router r1 as the next hop address (2001:192:168:22::1) is known to router r3.

```
neighbor 2001:192:168:33::1 next-hop-self
```

```
frr-pc(config-router)# address-family ipv6 unicast
frr-pc(config-router-af)# network 2001:192:168:2::/64
frr-pc(config-router-af)# neighbor 192.168.12.1 activate
frr-pc(config-router-af)# neighbor 192.168.12.1 route-map next-hop-ipv6 out
frr-pc(config-router-af)# neighbor 2001:192:168:33::1 activate
frr-pc(config-router-af)# neighbor 2001:192:168:33::1 next-hop-self
frr-pc(config-router-af)#[ ]
```

Figure 109. Assigning next hop address on router r2.

Step 11. Type the following command to exit from the address-family mode.

```
exit-address-family
```

```
frr-pc(config-router)# address-family ipv6 unicast
frr-pc(config-router-af)# network 2001:192:168:2::/64
frr-pc(config-router-af)# neighbor 192.168.12.1 activate
frr-pc(config-router-af)# neighbor 192.168.12.1 route-map next-hop-ipv6 out
frr-pc(config-router-af)# neighbor 2001:192:168:33::1 activate
frr-pc(config-router-af)# neighbor 2001:192:168:33::1 next-hop-self
frr-pc(config-router-af)# exit-address-family
frr-pc(config-router)#[ ]
```

Figure 110. Exiting from address-family mode.

Step 12. Type the following command to exit from configuration mode.

```
end
```

```
frr-pc(config-router)# address-family ipv6 unicast
frr-pc(config-router-af)# network 2001:192:168:2::/64
frr-pc(config-router-af)# neighbor 192.168.12.1 activate
frr-pc(config-router-af)# neighbor 192.168.12.1 route-map next-hop-ipv6 out
frr-pc(config-router-af)# neighbor 2001:192:168:33::1 activate
frr-pc(config-router-af)# neighbor 2001:192:168:33::1 next-hop-self
frr-pc(config-router-af)# exit-address-family
frr-pc(config-router)# end
frr-pc#
```

Figure 111. Exiting from configuration mode.

Step 13. Type the following command on router r2 to verify IPv6 neighbors. The BGP table shows that router r2 communicates with router r1 through IPv4 neighbor (192.168.12.1) and communicates with router r3 via IPv6 neighbor (2001:192:168:33::1).

```
show bgp ipv6 unicast summary
```

```
frr-pc# show bgp ipv6 unicast summary
BGP router identifier 192.168.22.1, local AS number 200 vrf-id 0
BGP table version 2
RIB entries 3, using 552 bytes of memory
Peers 2, using 41 KiB of memory

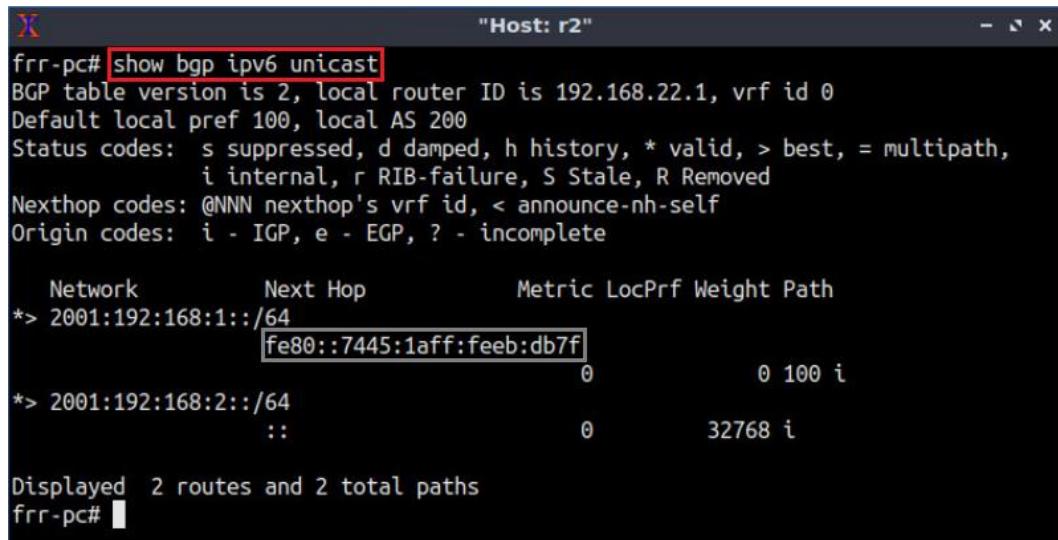
Neighbor          V      AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down State/P
fxRcd
192.168.12.1     4      100    109      113        0      0      0 00:08:49
1
2001:192:168:33::1 4      200      0       7        0      0      0 never      A
ctive

Total number of neighbors 2
frr-pc#
```

Figure 112. Verifying IPv6 BGP neighbors on router r2.

Step 14. Type the following command on router r2 to verify IPv6 routes. You will notice the link local address of 2001:192:168:12::2 as the next hop address for the network 2001:192:168:1::/64 which was used in the route-map.

```
show bgp ipv6 unicast
```



```
"Host: r2"
frr-pc# show bgp ipv6 unicast
BGP table version is 2, local router ID is 192.168.22.1, vrf id 0
Default local pref 100, local AS 200
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Next-hop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

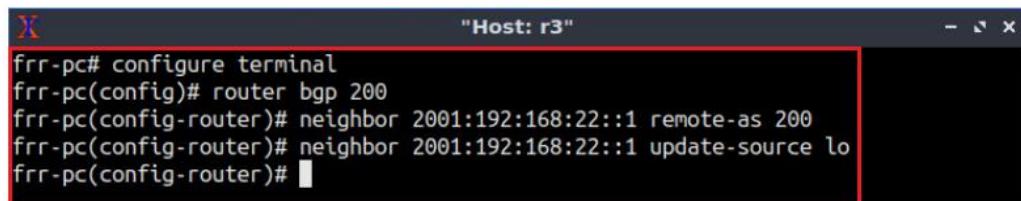
      Network          Next Hop           Metric LocPrf Weight Path
*-> 2001:192:168:1::/64        fe80::7445:1aff:feeb:db7f
                                0                  0 100 i
*-> 2001:192:168:2::/64        ::
                                0                  32768 i

Displayed 2 routes and 2 total paths
frr-pc#"
```

Figure 113. Verifying IPv6 routes on router r2.

5.3 Configure and verify IBGP on router r3

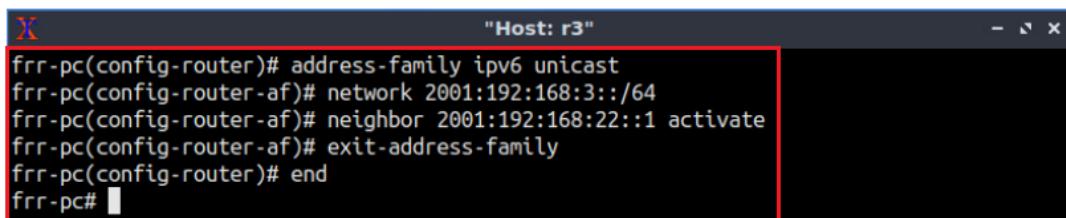
Step 1. Router r3 is configured similarly to router r2 but, with different metrics in order to establish IBGP peering with router r2. All the steps are summarized in the following figure.



```
"Host: r3"
frr-pc# configure terminal
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 2001:192:168:22::1 remote-as 200
frr-pc(config-router)# neighbor 2001:192:168:22::1 update-source lo
frr-pc(config-router)#"
```

Figure 114. Configuring BGP on router r3.

Step 2. Configure IPv6 address-family on router r3. All the steps are summarized in the following figure.



```
"Host: r3"
frr-pc(config-router)# address-family ipv6 unicast
frr-pc(config-router-af)# network 2001:192:168:3::/64
frr-pc(config-router-af)# neighbor 2001:192:168:22::1 activate
frr-pc(config-router-af)# exit-address-family
frr-pc(config-router)# end
frr-pc#"
```

Figure 115. Configuring IPv6 address-family on router r3.

Step 3. Type the following command on router r3 to verify IPv6 neighbors. The BGP table shows that router r3 communicates with router r2 through IPv6 neighbor (2001:192:168:22::1).

```
show bgp ipv6 summary
```

```

frr-pc# show bgp ipv6 summary

IPv6 Unicast Summary:
BGP router identifier 192.168.33.1, local AS number 200 vrf-id 0
BGP table version 3
RIB entries 5, using 920 bytes of memory
Peers 1, using 21 KiB of memory

Neighbor          V      AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State
/PfxRcd
2001:192:168:22::1 4        200       17      17       0     0    0 00:00:15
                           2

Total number of neighbors 1
frr-pc#

```

Figure 116. Verifying IPv6 BGP neighbors on router r3.

6 Verify BGP configuration

Step 1. Type the following command to verify the routing table of router r3.

```
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

B> 2001:192:168:1::/64 [200/0] via 2001:192:168:22::1 (recursive), 00:08:30
   *           via fe80::d80a:28ff:fe8b:3526, r3-eth1, 00:08:30
0
B  2001:192:168:2::/64 [200/0] via 2001:192:168:22::1 (recursive), 00:08:30
   via fe80::d80a:28ff:fe8b:3526, r3-eth1, 00:08:30
0
0>* 2001:192:168:2::/64 [110/20] via fe80::d80a:28ff:fe8b:3526, r3-eth1, 02:14:38
0  2001:192:168:3::/64 [110/10] is directly connected, r3-eth0, 02:14:16
C>* 2001:192:168:3::/64 is directly connected, r3-eth0, 02:20:50
0>* 2001:192:168:22::/64 [110/20] via fe80::d80a:28ff:fe8b:3526, r3-eth1, 02:14:38
0
0  2001:192:168:23::/64 [110/10] is directly connected, r3-eth1, 02:14:43
C>* 2001:192:168:23::/64 is directly connected, r3-eth1, 02:20:50
0  2001:192:168:33::/64 [110/10] is directly connected, lo, 02:14:16
C>* 2001:192:168:33::/64 is directly connected, lo, 02:20:51
C * fe80::/64 is directly connected, r3-eth1, 02:20:51
C>* fe80::/64 is directly connected, r3-eth0, 02:20:51
frr-pc#

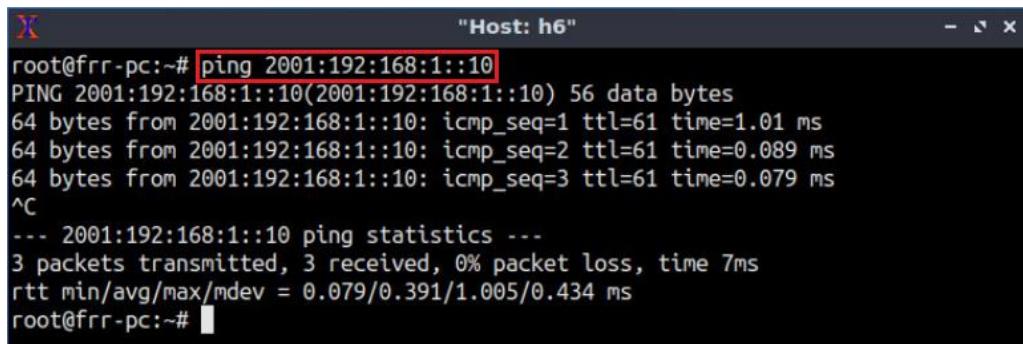
```

Figure 117. Verifying IPv6 routes on router r3.

Consider Figure 117. To reach the network 2001:192:168:1::/64, router r3 uses the link local address of the interface r3-eth1 to communicate with router r2 since they are directly connected. Then, router r2 (2001:192:168:22::1) uses IPv4 BGP transport to reach the LAN 2001:192:168:1::/64.

Step 2. On host h6 terminal, perform a connectivity between host h6 and host h4 by issuing the command shown below. To stop the test, press **Ctrl+c**. The result will show a successful connectivity test.

```
ping 2001:192:168:1::10
```



```
"Host: h6"
root@frr-pc:~# ping 2001:192:168:1::10
PING 2001:192:168:1::10(2001:192:168:1::10) 56 data bytes
64 bytes from 2001:192:168:1::10: icmp_seq=1 ttl=61 time=1.01 ms
64 bytes from 2001:192:168:1::10: icmp_seq=2 ttl=61 time=0.089 ms
64 bytes from 2001:192:168:1::10: icmp_seq=3 ttl=61 time=0.079 ms
^C
--- 2001:192:168:1::10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 7ms
rtt min/avg/max/mdev = 0.079/0.391/1.005/0.434 ms
root@frr-pc:~#
```

Figure 118. Connectivity test using **ping** command.

This concludes Lab 11. Stop the emulation and then exit out of MiniEdit.

References

1. J. Postelet al., “Internet protocol”, 1981. [Online]. Available: <https://tools.ietf.org/html/rfc791>
2. S. Deering, R. Hinden, “Internet Protocol, Version 6 (IPv6) Specification”, 1998. [Online]. Available: <https://tools.ietf.org/html/rfc2460>
3. Cisco, “IPv6 Addressing and Basic Connectivity Configuration Guide, Cisco IOS Release 15M&T”, 2013. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipv6_basic/configuration/15-mt/ip6b-15-mt-book.pdf
4. A. Tanenbaum, D. Wetherall, “Computer networks”, 5th Edition, Pearson, 2012.
5. Cisco, “What Are OSPF Areas and Virtual Links?”, 2016. [Online]. Available: <https://www.cisco.com/c/en/us/support/docs/ip/open-shortest-path-first-ospf/13703-8.html>
6. R. Coltun, D. Ferguson, J. Moy, A. Lindem, “OSPF for IPv6”, 2008.
7. T. Bates, R. Chandra, D. Katz, Y. Rekhter, “Multiprotocol Extensions for BGP-4”, 1998. [Online]. Available: <https://tools.ietf.org/html/rfc2283>
8. Cisco, “Implementing Cisco IP Routing (ROUTE) Foundation Learning Guide”, Pearson, 2015.



BORDER GATEWAY PROTOCOL

Lab 12: IP Spoofing and Mitigation Techniques

Document Version: 3-5-2020



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	4
1.1 BGP overview	4
1.2 IP Spoofing and DoS attacks.....	4
1.3 Anti-Spoofing techniques.....	5
1.3.1 Unicast Reverse Path Forwarding (uRPF)	5
1.3.2 Route filtering.....	6
2 Lab topology.....	6
2.1 Lab settings.....	8
2.2 Open topology and load the configuration.....	8
2.3 Load zebra daemon and Verify IP addresses	11
3 Configure BGP on routers	16
3.1 Configure EBGP on all routers.....	16
3.2 Configure IBGP on routers r2 and r3.....	20
4 Perform IP spoofing and DoS attack.....	23
5 Mitigate DDoS attach by using IP source filtering	24
References	26

Overview

This lab introduces Internet Protocol (IP) address spoofing that occurs on the Internet between routers running Border Gateway Protocol (BGP). In this lab, a compromised host will spoof the IP address and launch a Denial of Service (DoS) on a victim, each in a different Autonomous System (AS). The goal of this lab is to configure the Internet Service Provider (ISP) to mitigate IP spoofing attacks by applying the appropriate filters on the network traffic of its customers.

Objectives

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

1. Configure BGP as the main protocol between ASes.
2. Understand and configure IP spoofing and DoS attack.
3. Understand IP spoofing mitigation techniques.
4. Apply route filters to mitigate IP spoofing.

Lab settings

The information in Table 1 provides the credentials to access Client1 machine.

Table 1. Credentials to access Client1 machine.

Device	Account	Password
Client1	admin	password

Lab roadmap

This lab is organized as follows:

1. Section 1: Introduction.
2. Section 2: Lab topology.
3. Section 3: Configure BGP on routers.
4. Section 4: Perform IP spoofing and DoS attack.
5. Section 5: Mitigate DDoS attack by using IP source filtering.

1 Introduction

1.1 BGP overview

BGP is an exterior gateway protocol designed to exchange routing and reachability information among ASes on the Internet. BGP is relevant to network administrators of large organizations which connect to one or more ISPs, as well as to ISPs who connect to other network providers. In terms of BGP, an AS is referred to as a routing domain, where all networked systems operate common routing protocols and are under the control of a single administration¹.

BGP is a form of distance vector protocol. It requires each router to maintain a table, which stores the distance and the output interface (i.e., vector) to remote networks. BGP makes routing decisions based on paths, network policies, or rule set configured by a network administrator and is involved in making core routing decisions¹.

Two routers that establish a BGP connection are referred to as BGP peers or neighbors. BGP sessions run over Transmission Control Protocol (TCP). If a BGP session is established between two neighbors in different ASes, the session is referred to as an External BGP (EBGP) session. If the session is established between two neighbors in the same AS, the session is referred to as Internal BGP (IBGP)¹. Figure 1 shows a network running BGP protocol. Routers that exchange information within the same AS use IBGP, while routers that exchange information between different ASes use EBGP.

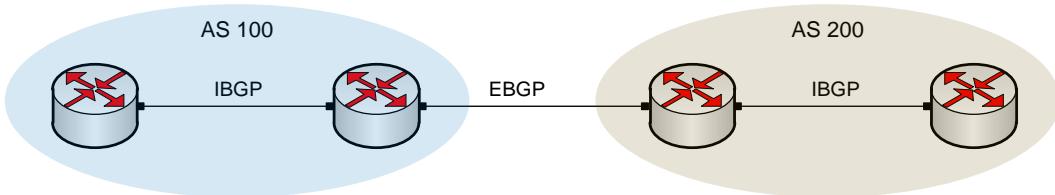


Figure 1. Routers that exchange information within the same AS use IBGP, while routers that exchange information between different ASes use EBGP.

1.2 IP Spoofing and DoS attacks

IP source address spoofing is the process of originating IP packets with source addresses other than those assigned to the origin host. An attacker that spoofs source IP addresses appears as the to be another host². IP spoofing can be exploited in several ways, mainly to launch DoS attacks. The latter is an attack that can exhaust the computing and communication resources of its victim within a short period of time³.

Consider Figure 2. Host A (attacker) spoofs the source IP address of host C (victim) and request host B to send 100 GB of data. Host B will receive the request and sends the data to the spoofed source address, i.e., to host C. Thus, consuming the resources of the victim.

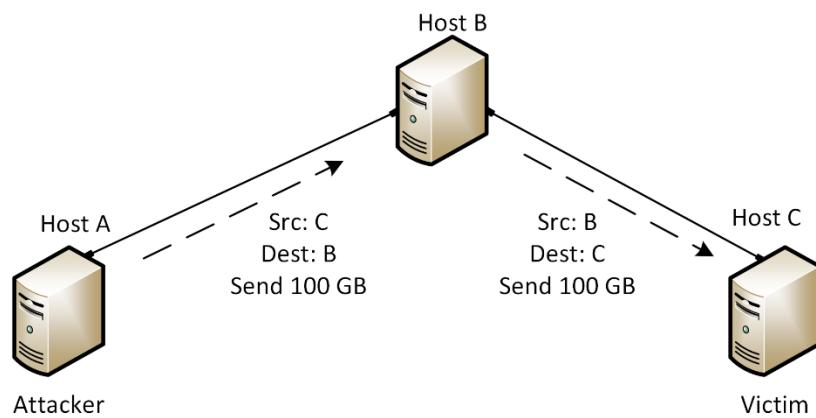


Figure 2. Host A performs DoS attack on host C by spoofing its IP address.

1.3 Anti-Spoofing techniques

Mutually Agreed Norms for Routing Security (MANRS) is a global initiative, supported by the Internet Society, that provides crucial fixes to reduce the most common routing threats. MANRS has many recommendations to prevent IP spoofing by ingress filtering, e.g., checking the source addresses of IP datagrams⁴.

1.3.1 Unicast Reverse Path Forwarding (uRPF)

uRPF is one effective method to prevent IP spoofing. uRPF has multiple modes of operation, among them is the uRPF strict mode, in which the router accepts incoming packets on a specific interface if two conditions satisfy⁴:

1. The source IP address of the incoming packet has an entry in the routing table.
2. The router uses the same interface to reach this source as where it received the packet on.

Consider Figure 3. Router r1 uses uRPF strict mode. Incoming packets on interface *r1-eth0* that have source IP address of Host B will be dropped, since router r1 uses interface *r1-eth1* to reach host B.

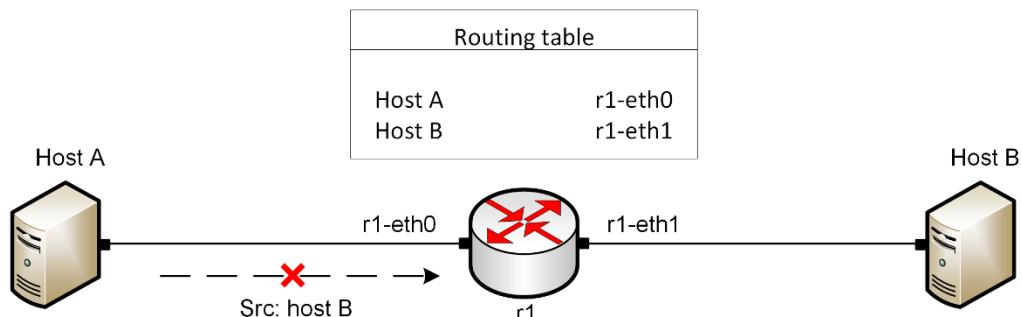


Figure 3. Router r1 uses uRPF to prevent spoofed IP packets.

1.3.2 Route filtering

Route filtering is a method for selectively identifying routes that are advertised or received from neighbor routers. Route filtering may be used to manipulate traffic flows, reduce memory utilization, or to improve security⁵.

Network operators should apply route filters to prevent spoofed IP packets from their customers. Consider Figure 4. The ISP (router r2) filters inbound network traffic of its customers, i.e., traffic sent from routers r1 and r3, based on their assigned IP addresses. Thus, each customer can't generate network traffic with spoofed IP addresses.

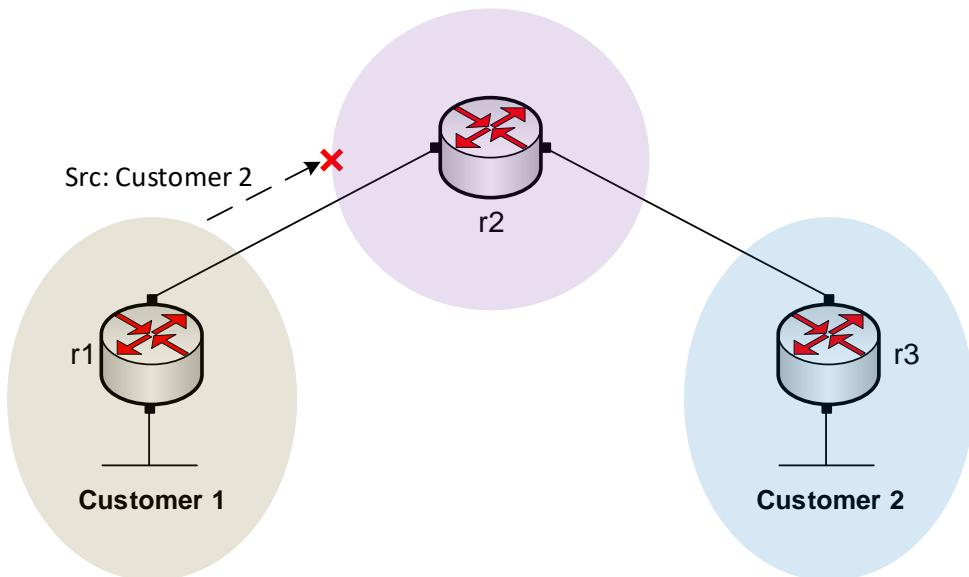


Figure 4. The ISP drops network traffic sent from Customer 1, since their source IP address corresponds to Customer 2.

In this lab, we will apply the route filters to prevent IP spoofing using netfilter. The latter is a framework for packet filtering built in Linux kernel⁶.

2 Lab topology

Consider Figure 5. The topology consists of three ASes. The ISP, consisting of routers r2 and r3, provides Internet service to the Campus-1 (router r1) and Campus-2 (router r4) networks. The Autonomous System Numbers (ASNs) assigned to Campus-1, ISP, and Campus-2 are 100, 200, and 300, respectively. The ISP communicates with the Campus networks via EBGP routing protocol, and the routers within the ISP communicate using IBGP. Host h1 in Campus-1 spoofs the IP address of host h4 in Campus-2. Consequently, host h1 launches a DoS attack on host h4 using hosts h2 and h3. To mitigate IP spoofing, the ISP (router r2) applies the appropriate route filters on the network traffic generated from Campus-1.

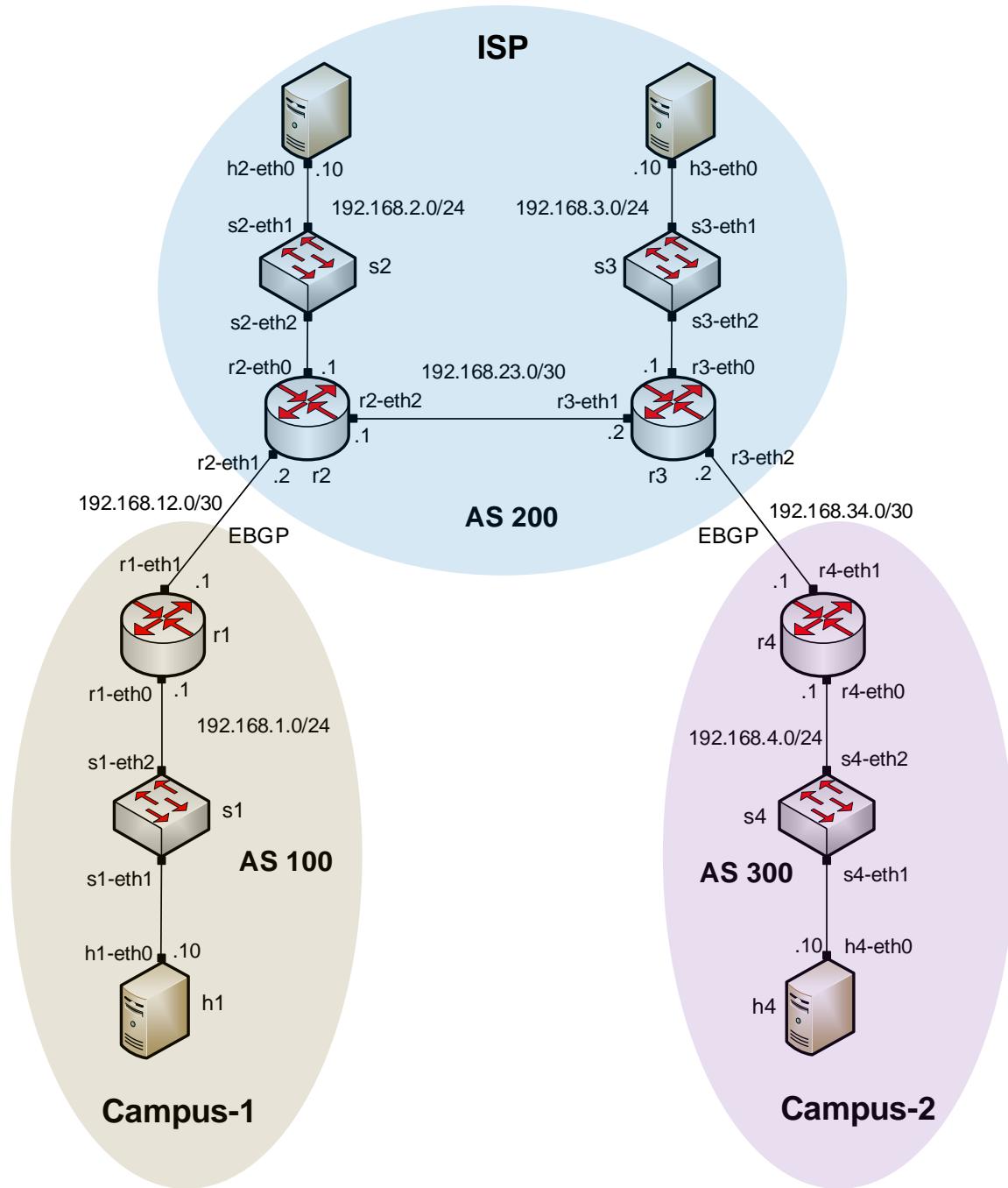


Figure 5. Lab topology.

2.1 Lab settings

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

Table 2. Topology information.

Device	Interface	IPV4 Address	Subnet	Default gateway
r1 (Campus-1)	r1-eth0	192.168.1.1	/24	N/A
	r1-eth1	192.168.12.1	/30	N/A
r2 (ISP)	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
r3 (ISP)	r3-eth0	192.168.3.1	/24	N/A
	r3-eth1	192.168.23.2	/30	N/A
	r3-eth2	192.168.34.1	/30	N/A
r4 (Campus-2)	r4-eth0	192.168.4.1	/24	N/A
	r4-eth1	192.168.34.2	/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
h3	h3-eth0	192.168.3.10	/24	192.168.3.1
h4	h4-eth0	192.168.4.10	/24	192.168.4.1

2.2 Open topology and load the configuration

Step 1. Start by launching Miniedit by clicking on Desktop's shortcut. When prompted for a password, type `password`.



Figure 6. MiniEdit shortcut.

Step 2. On Miniedit's menu bar, click on *File* then *open* to load the lab's topology. Locate the *Lab12.mn* topology file in the default directory, */home/frr/BGP_Labs/lab12* and click on *Open*.

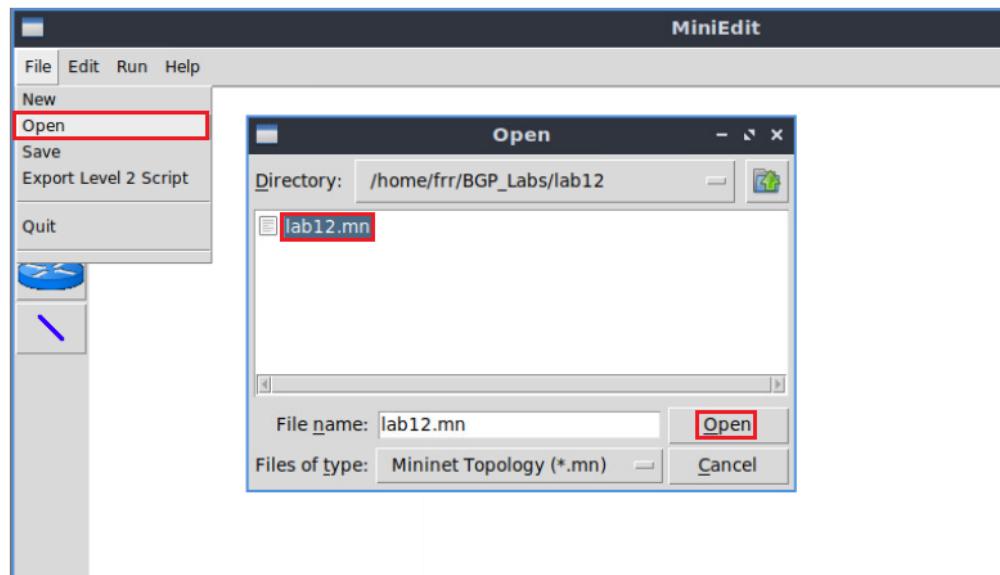


Figure 7. MiniEdit's Open dialog.

At this point the topology is loaded with all the required network components. You will execute a script that will load the configuration of the routers.

Step 3. Open the Linux terminal.

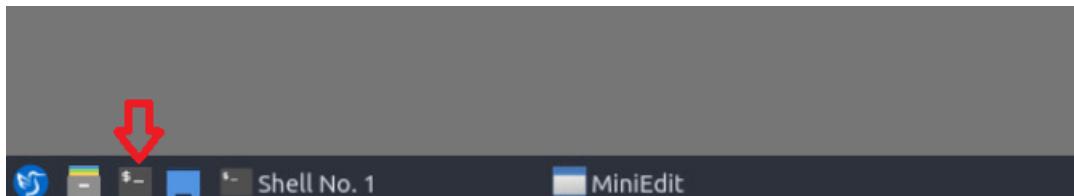
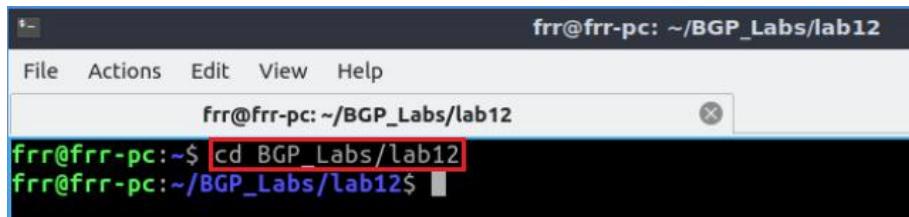


Figure 8. Opening Linux terminal

Step 4. Click on the Linux's terminal and navigate into *BGP_Labs/lab12* 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 BGP_Labs/lab12
```

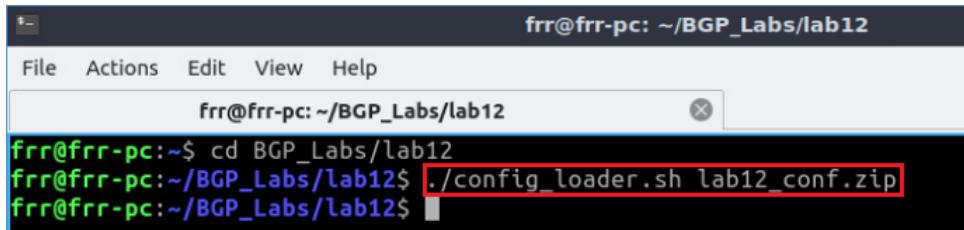


```
frr@frr-pc: ~/BGP_Labs/lab12
frr@frr-pc:~$ cd BGP_Labs/lab12
frr@frr-pc:~/BGP_Labs/lab12$
```

Figure 9. Entering the *BGP_Labs/lab12* directory.

Step 5. To execute the shell script, type the following command. The argument of the program corresponds to the configuration zip file that will be loaded in all the routers in the topology.

```
./config_loader.sh lab12_conf.zip
```

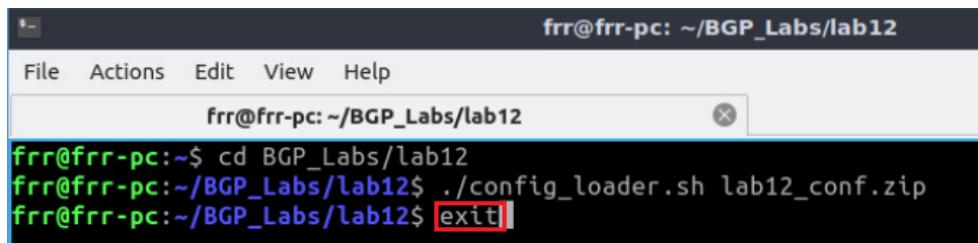


```
frr@frr-pc: ~/BGP_Labs/lab12
frr@frr-pc:~$ cd BGP_Labs/lab12
frr@frr-pc:~/BGP_Labs/lab12$ ./config_loader.sh lab12_conf.zip
frr@frr-pc:~/BGP_Labs/lab12$
```

Figure 10. Executing the shell script to load the configuration.

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

```
exit
```



```
frr@frr-pc: ~/BGP_Labs/lab12
frr@frr-pc:~$ cd BGP_Labs/lab12
frr@frr-pc:~/BGP_Labs/lab12$ ./config_loader.sh lab12_conf.zip
frr@frr-pc:~/BGP_Labs/lab12$ exit
```

Figure 11. Exiting from the terminal.

Step 7. At this point hosts h1, h2, h3 and h4 interfaces are configured. To proceed with the emulation, click on the *Run* button located in lower left-hand side.



Figure 12. Starting the emulation.

Step 8. Click on Mininet's terminal, i.e., the one launched when MiniEdit was started.

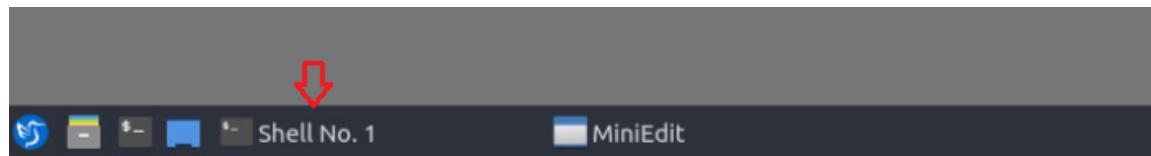


Figure 13. Opening Mininet's terminal.

Step 9. Issue the following command to display the interface names and connections.

```
links
```

```
Shell No. 1
File Actions Edit View Help
Shell No. 1
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)
r3-eth2<->r4-eth1 (OK OK)
mininet> █
```

A screenshot of a terminal window titled "Shell No. 1". The window contains the command "links" followed by a list of network connections. The first connection, "h1-eth0<->s1-eth1", is highlighted with a gray box and a red border. The output shows ten connections in total between hosts h1, s1, r1, r2, r3, and r4, with each connection status as "(OK OK)".

Figure 14. Displaying network interfaces.

In Figure 14, the link displayed within the gray box indicates that interface *eth0* of host *h1* connects to interface *eth1* of switch *s1* (i.e., *h1-eth0<->s1-eth1*).

2.3 Load zebra daemon and Verify IP addresses

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

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.

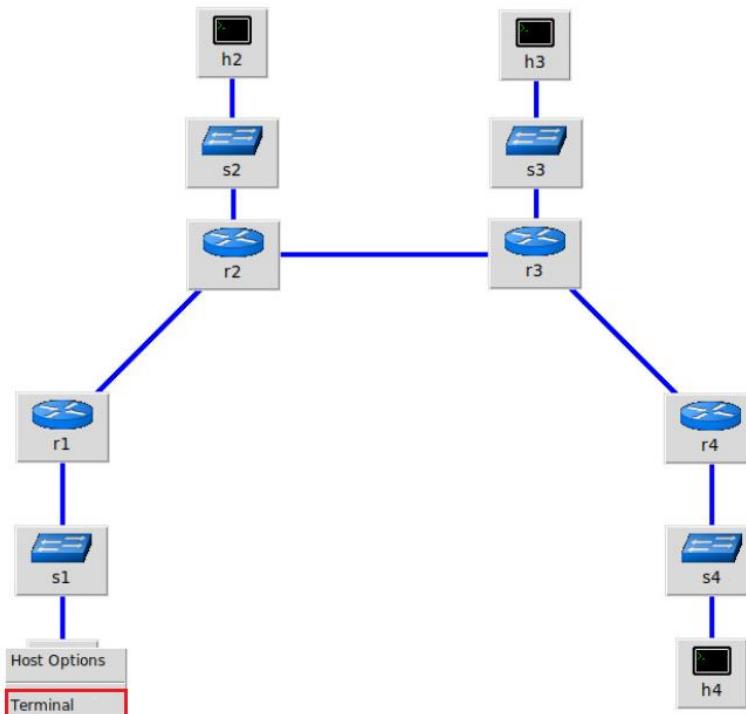


Figure 15. Opening terminal on host h1.

Step 2. On host h1 terminal, type the command shown below to verify that the IP address was assigned successfully. You will verify that host h1 has an interface, *h1-eth0* 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 16. Output of `ifconfig` command.

Step 3. On 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 17. Output of `route` command.

Step 4. In order to verify hosts h2, h3 and h4, proceed similarly by repeating from step 1 to step 3 on host h2, h3 and h4 terminals. Similar results should be observed.

Step 5. You will validate that the router interfaces are configured correctly according to Table 2. In order to verify router r1, hold right-click on router r1 and select Terminal.

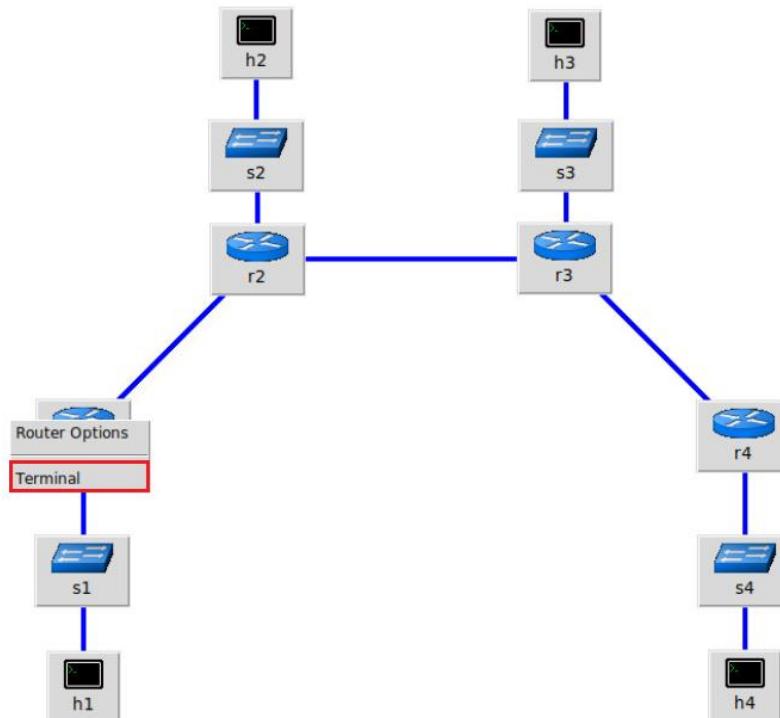


Figure 18. Opening terminal on router r1.

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

```
zebra
```

The terminal window shows the command 'zebra' being typed at the root prompt 'root@frr-pc:/etc/routers/r1#'. The window title is "Host: r1". The command is highlighted with a red box.

Figure 19. Starting zebra daemon.

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

```
vtysh
```

The terminal window shows the command 'vtysh' being typed at the root prompt 'root@frr-pc:/etc/routers/r1#'. The window title is "Host: r1". The command is highlighted with a red box. Below the command, the FRRouting version information is displayed: "Hello, this is FRRouting (version 7.2-dev). Copyright 1996-2005 Kunihiro Ishiguro, et al." The prompt 'frr-pc#' is shown at the bottom.

Figure 20. Starting vtysh on router r1.

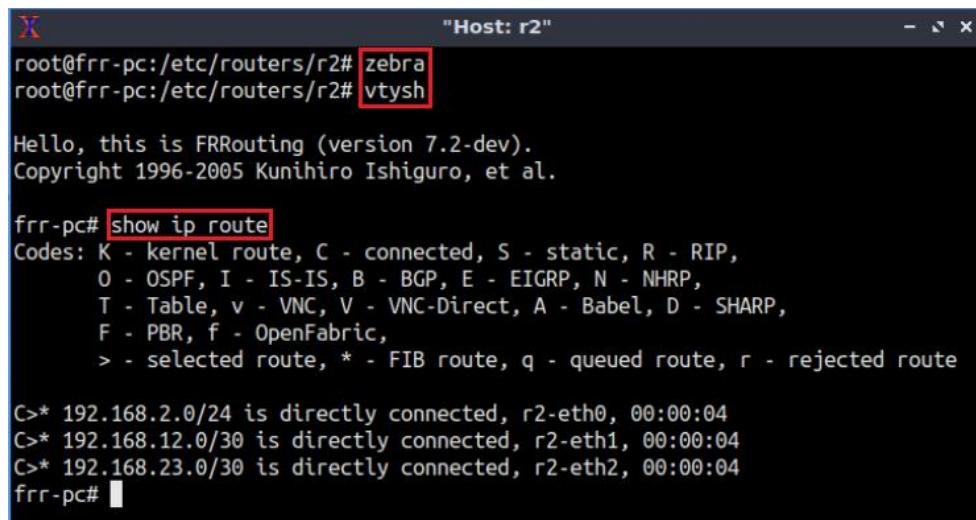
Step 8. Type the following command on router r1 terminal to verify the routing table of router r1. It will list all the directly connected networks. The routing table of router r1 does not contain any route to the networks attached to routers r2 (192.168.2.0/24), r3 (192.168.3.0/24) and r4 (192.168.4.0/24) as there is no routing protocol configured yet.

```
show ip route
```

The terminal window shows the command 'show ip route' being typed at the root prompt 'frr-pc#'. The window title is "Host: r1". The command is highlighted with a red box. The output shows the routing table with two entries: 'C>* 192.168.1.0/24 is directly connected, r1-eth0, 00:00:05' and 'C>* 192.168.12.0/30 is directly connected, r1-eth1, 00:00:05'. The prompt 'frr-pc#' is shown at the bottom.

Figure 21. Displaying the routing table of router r1.

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.



The terminal window shows the command "show ip route" being entered into the Zebra vtysh interface. The output displays the routing table with three directly connected routes: 192.168.2.0/24, 192.168.12.0/30, and 192.168.23.0/30, all via interfaces r2-eth0, r2-eth1, and r2-eth2 respectively.

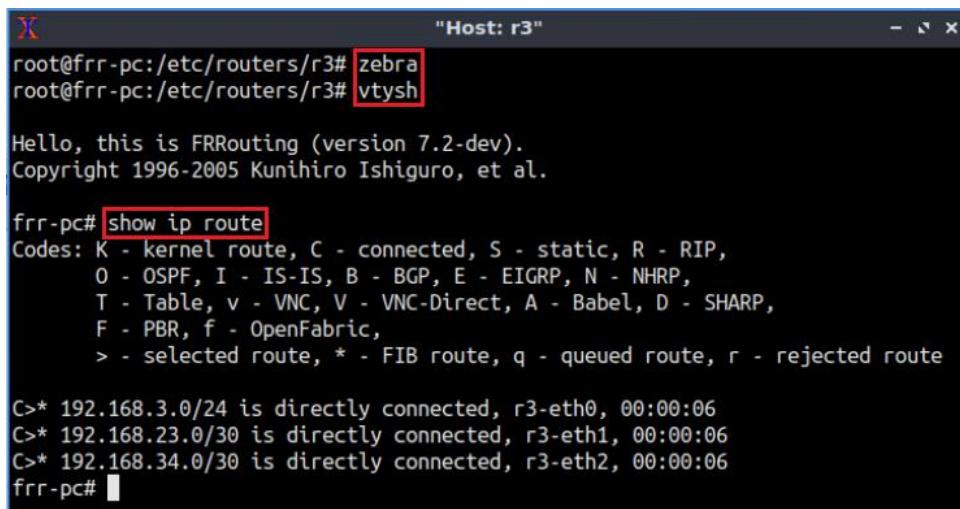
```
"Host: r2"
root@frr-pc:/etc/routers/r2# zebra
root@frr-pc:/etc/routers/r2# vtysh
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

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, 00:00:04
C>* 192.168.12.0/30 is directly connected, r2-eth1, 00:00:04
C>* 192.168.23.0/30 is directly connected, r2-eth2, 00:00:04
frr-pc#
```

Figure 22. Displaying the routing table of router r2.

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



The terminal window shows the command "show ip route" being entered into the Zebra vtysh interface. The output displays the routing table with three directly connected routes: 192.168.3.0/24, 192.168.23.0/30, and 192.168.34.0/30, all via interfaces r3-eth0, r3-eth1, and r3-eth2 respectively.

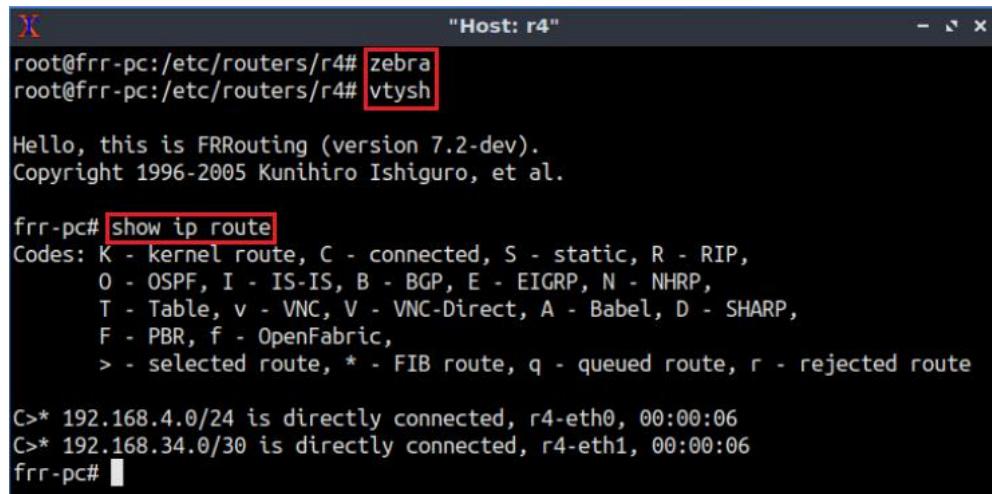
```
"Host: r3"
root@frr-pc:/etc/routers/r3# zebra
root@frr-pc:/etc/routers/r3# vtysh
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

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>* 192.168.3.0/24 is directly connected, r3-eth0, 00:00:06
C>* 192.168.23.0/30 is directly connected, r3-eth1, 00:00:06
C>* 192.168.34.0/30 is directly connected, r3-eth2, 00:00:06
frr-pc#
```

Figure 23. Displaying the routing table of router r3.

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



```
"Host: r4"
root@frr-pc:/etc/routers/r4# zebra
root@frr-pc:/etc/routers/r4# 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,
      0 - 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>* 192.168.4.0/24 is directly connected, r4-eth0, 00:00:06
C>* 192.168.34.0/30 is directly connected, r4-eth1, 00:00:06
frr-pc#
```

Figure 24. Displaying the routing table of router r4.

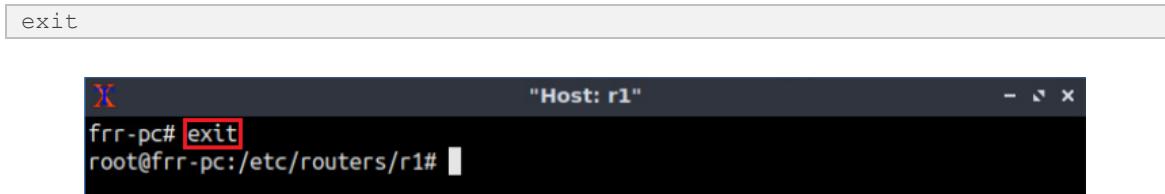
3 Configure BGP on routers

In this section, you will configure BGP on the routers that are hosted in different ASes. You will assign BGP neighbors to allow the routers to exchange BGP routes. Furthermore, all routers will advertise their Local Area Networks (LANs) via BGP.

3.1 Configure EBGP on routers

In this section, you will configure EBGP on all routers.

Step 1. To configure BGP routing protocol, you need to enable the BGP daemon first. In router r1 terminal, type the following command to exit the vtysh session:

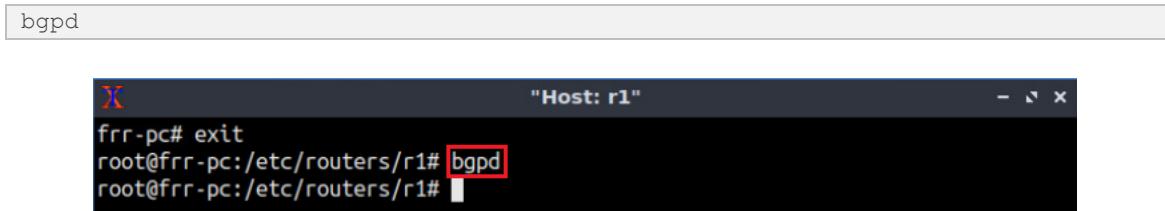


```
exit
```

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

Figure 25. Exiting the vtysh session.

Step 2. Type the following command on r1 terminal to enable and start BGP routing protocol.



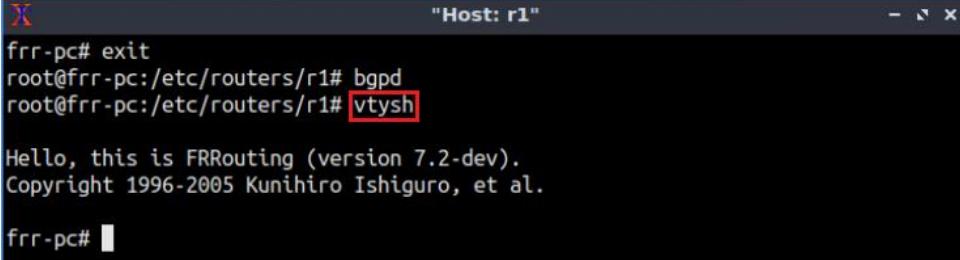
```
bgpd
```

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

Figure 26. Starting BGP daemon.

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

```
vtysh
```



The terminal window shows the FRRouting version 7.2-dev startup message. The command `vtysh` is highlighted in red at the prompt.

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

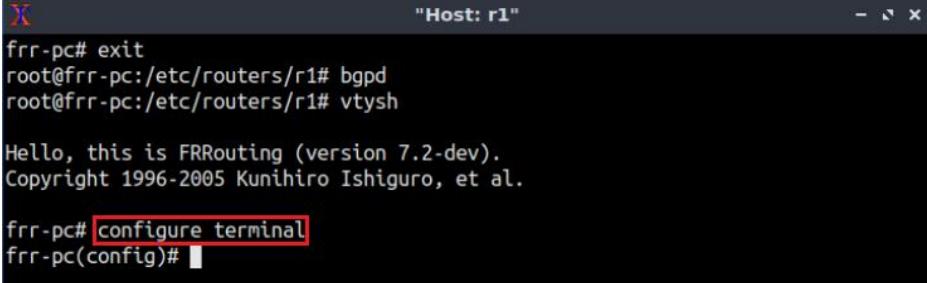
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc#
```

Figure 27. Starting vtysh in router r1.

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

```
configure terminal
```



The terminal window shows the FRRouting version 7.2-dev startup message. The command `configure terminal` is highlighted in red at the prompt.

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

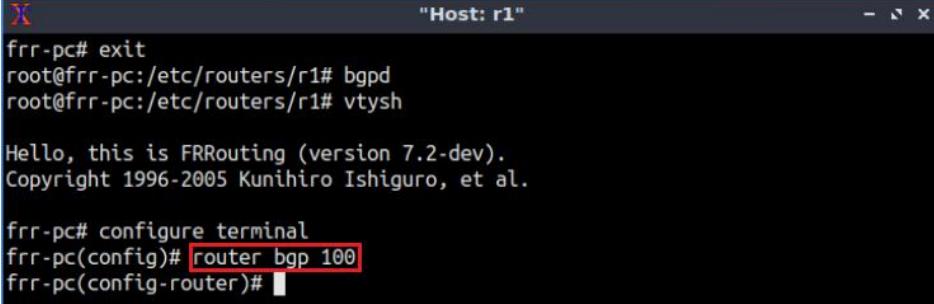
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

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

Figure 28. Enabling configuration mode in router r1.

Step 5. The ASN assigned for router r1 is 100. In order to configure BGP, type the following command:

```
router bgp 100
```



The terminal window shows the FRRouting version 7.2-dev startup message. The command `router bgp 100` is highlighted in red at the prompt.

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

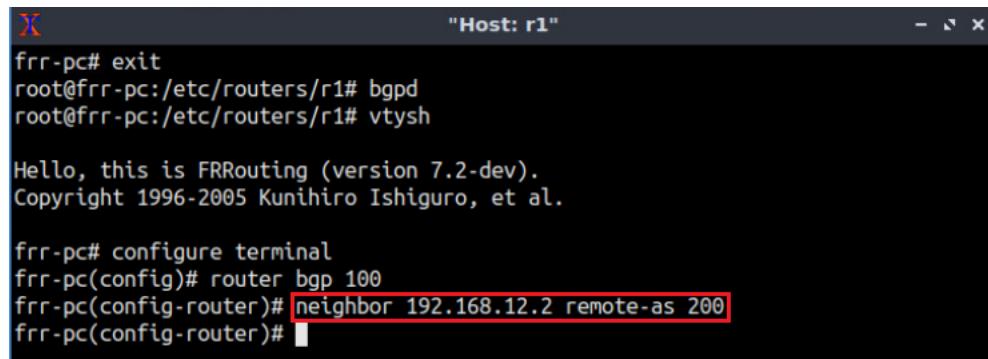
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc# configure terminal
frr-pc(config)# router bgp 100
frr-pc(config-router)#
```

Figure 29. Configuring BGP on router r1.

Step 6. To configure a BGP neighbor to router r1 (AS 100), type the command shown below. This command specifies the neighbor IP address (192.168.12.2) and the ASN of the remote BGP peer (AS 200).

```
neighbor 192.168.12.2 remote-as 200
```



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

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

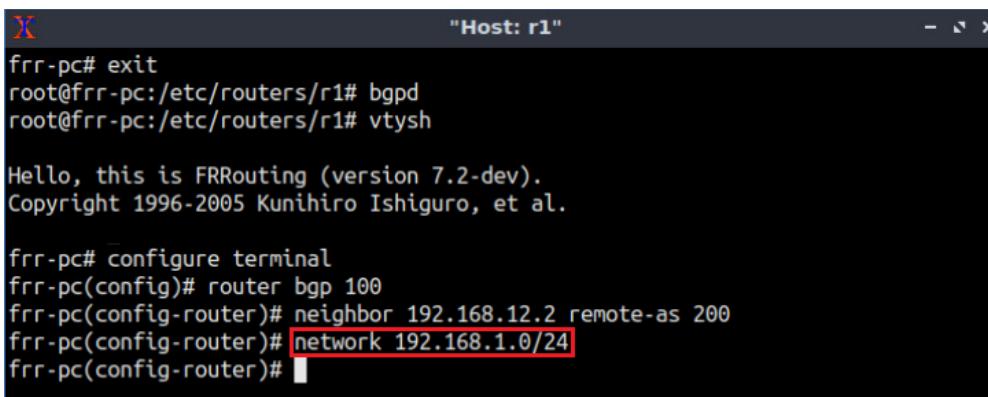
frr-pc# configure terminal
frr-pc(config)# router bgp 100
frr-pc(config-router)# neighbor 192.168.12.2 remote-as 200
frr-pc(config-router)#

```

Figure 30. Assigning BGP neighbor to router r1.

Step 7. In this step, router r1 will advertise the LAN 192.168.1.0/24 to its BGP peers. To do so, issue the following command:

```
network 192.168.1.0/24
```



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

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

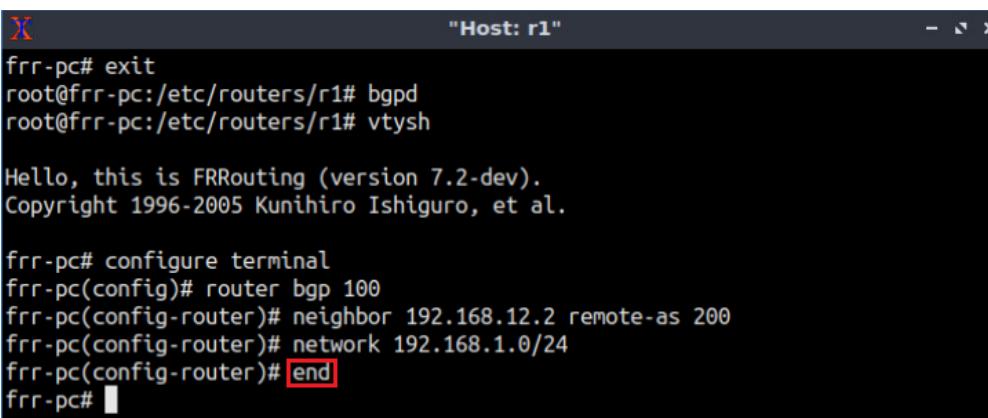
frr-pc# configure terminal
frr-pc(config)# router bgp 100
frr-pc(config-router)# neighbor 192.168.12.2 remote-as 200
frr-pc(config-router)# network 192.168.1.0/24
frr-pc(config-router)#

```

Figure 31. Advertising local network in router r1.

Step 8. Type the following command to exit from configuration mode.

```
end
```



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

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

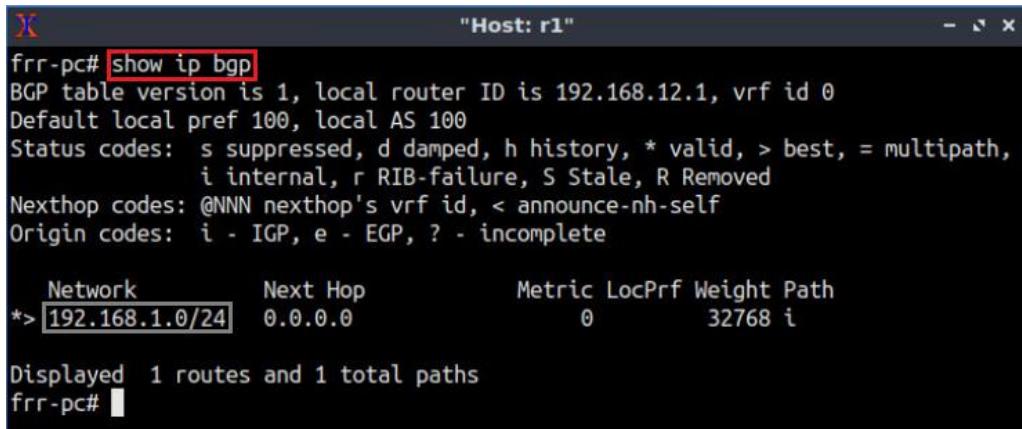
frr-pc# configure terminal
frr-pc(config)# router bgp 100
frr-pc(config-router)# neighbor 192.168.12.2 remote-as 200
frr-pc(config-router)# network 192.168.1.0/24
frr-pc(config-router)# end
frr-pc#

```

Figure 32. Exiting from configuration mode.

Step 9. Type the following command to verify BGP networks. You will observe the LAN of router r1.

```
show ip bgp
```



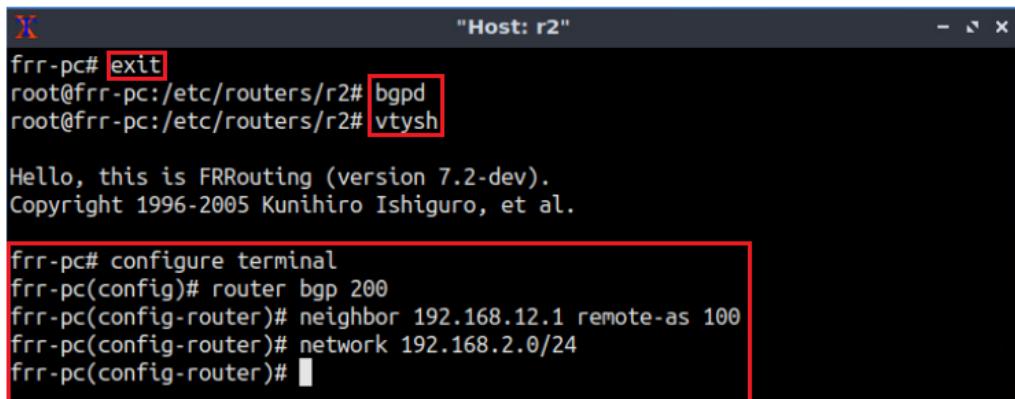
```
"Host: r1"
frr-pc# show ip bgp
BGP table version is 1, local router ID is 192.168.12.1, vrf id 0
Default local pref 100, local AS 100
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

Network          Next Hop           Metric LocPrf Weight Path
*-> 192.168.1.0/24  0.0.0.0            0        32768 i

Displayed 1 routes and 1 total paths
frr-pc#
```

Figure 33. Verifying BGP networks in router r1.

Step 10. Follow from step 1 to step 7 but with different metrics in order to configure BGP on router r2. All the steps are summarized in the following figure.



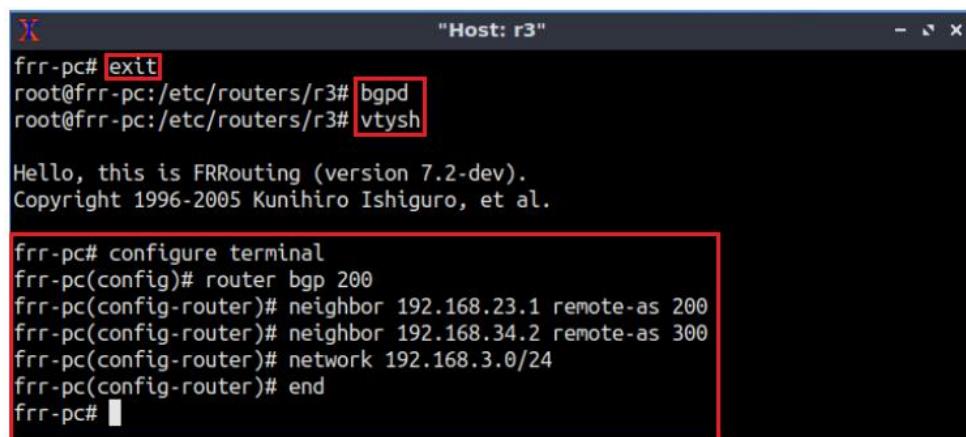
```
"Host: r2"
frr-pc# exit
root@frr-pc:/etc/routers/r2# bgpd
root@frr-pc:/etc/routers/r2# vtysh

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

frr-pc# configure terminal
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.12.1 remote-as 100
frr-pc(config-router)# network 192.168.2.0/24
frr-pc(config-router)#
```

Figure 34. Configuring BGP on router r2.

Step 11. Follow from step 1 to step 8 but with different metrics in order to configure EBGP on router r3 to establish BGP peering with routers r2 and r4. All the steps are summarized in the following figure.



```
"Host: r3"
frr-pc# exit
root@frr-pc:/etc/routers/r3# bgpd
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 bgp 200
frr-pc(config-router)# neighbor 192.168.23.1 remote-as 200
frr-pc(config-router)# neighbor 192.168.34.2 remote-as 300
frr-pc(config-router)# network 192.168.3.0/24
frr-pc(config-router)# end
frr-pc#
```

Figure 35. Configuring EBGP on router r3.

Step 12. Follow from step 1 to step 8 but with different metrics in order to configure EBGP on router r4. All the steps are summarized in the following figure.

The terminal window shows the configuration of BGP on router r4. The user enters 'exit' to leave vtysh, then enters 'bgpd' to start the BGP daemon. Finally, they enter 'vtysh' again. The configuration command block is highlighted with a red rectangle:

```
frr-pc# exit
root@frr-pc:/etc/routers/r4# bgpd
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 bgp 300
frr-pc(config-router)# neighbor 192.168.34.1 remote-as 200
frr-pc(config-router)# network 192.168.4.0/24
frr-pc(config-router)# end
frr-pc#
```

Figure 36. Configuring BGP on router r4.

Step 13. Type the following command to verify the routing table of router r1. Router r1 has a route to the network 192.168.2.0/24 only since IBGP is not configured between routers r2 and r3 yet.

The terminal window shows the routing table of router r1. The user enters 'show ip route'. The output shows a route to 192.168.2.0/24 via router r1's eth1 interface. The command 'show ip route' is highlighted with a red rectangle:

```
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-eth0, 00:36:26
B>* [192.168.2.0/24] [20/0] via 192.168.12.2, r1-eth1, 00:30:24
C>* 192.168.12.0/30 is directly connected, r1-eth1, 00:36:26
frr-pc#
```

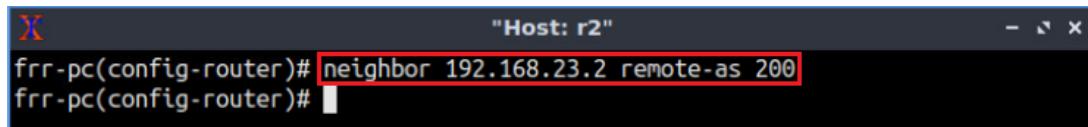
Figure 37. Displaying the routing table of router r1.

3.2 Configure IBGP on routers r2 and r3

In this section, you will configure IBGP on routers r2 and r3. Furthermore, you will configure BGP next-hop-self on routers r2 and r3 so that these routers have valid routes to the EBGP routes that are advertised by their IBGP neighbors.

Step 1. Type the following command on r2 terminal to establish IBGP peering with router r3.

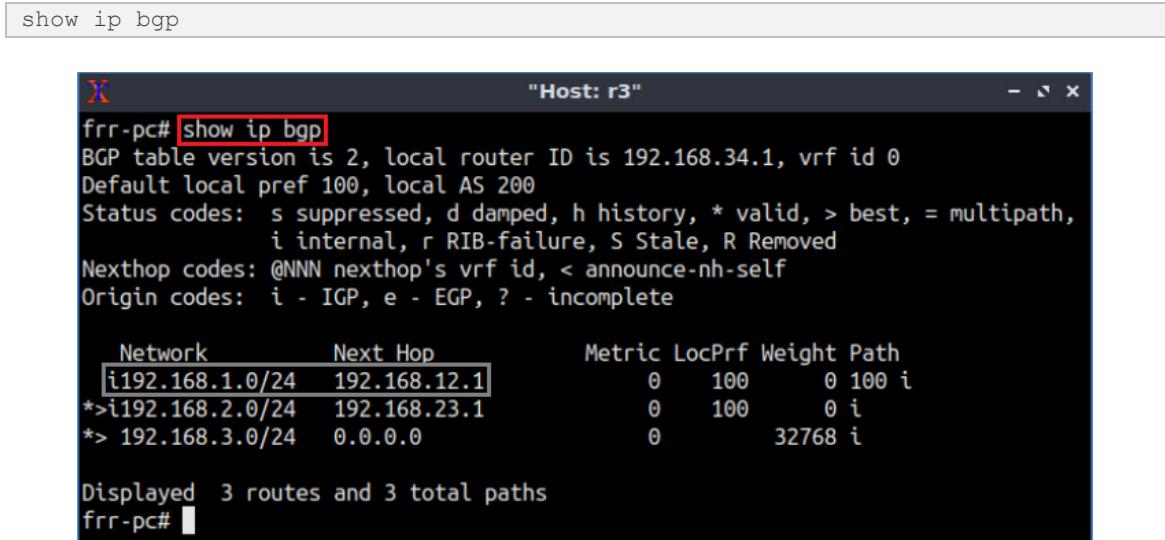
```
neighbor 192.168.23.2 remote-as 200
```



```
"Host: r2"
frr-pc(config-router)# neighbor 192.168.23.2 remote-as 200
frr-pc(config-router)#[ ]"
```

Figure 38. Assigning BGP neighbor to router r2.

Step 2. Type the following command to verify the BGP table of router r3. Router r3 can't reach the network 192.168.1.0/24 since the next hop address (192.168.12.1) is not known to router r3.



```
show ip bgp

"Host: r3"
frr-pc# show ip bgp
BGP table version is 2, local router ID is 192.168.34.1, vrf id 0
Default local pref 100, local AS 200
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-Failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop           Metric LocPrf Weight Path
  [i]192.168.1.0/24  192.168.12.1        0    100      0 100 i
*->i192.168.2.0/24  192.168.23.1        0    100      0 i
*-> 192.168.3.0/24  0.0.0.0            0          32768 i

Displayed 3 routes and 3 total paths
frr-pc# [ ]"
```

Figure 39. Verifying BGP networks on router r3.

Step 3. Router r2 will configure BGP *next-hop-self* so that the neighbor 192.168.23.2 (router r3) can reach the EBGP routes advertised by router r2, such as 192.168.1.0/24, through router r2. To do so, type the following command on router r2 terminal.

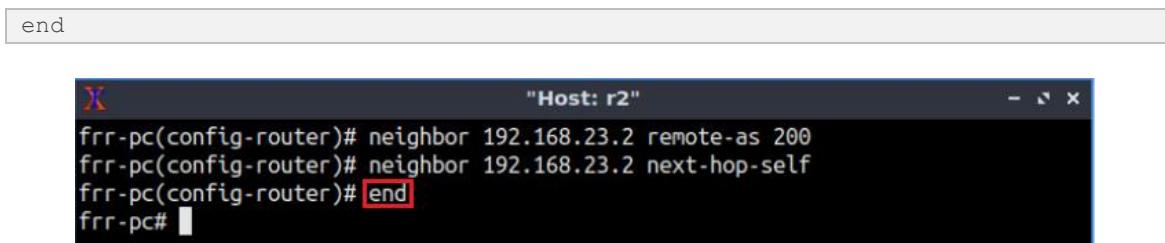


```
neighbor 192.168.23.2 next-hop-self

"Host: r2"
frr-pc(config-router)# neighbor 192.168.23.2 remote-as 200
frr-pc(config-router)# neighbor 192.168.23.2 next-hop-self
frr-pc(config-router)#[ ]"
```

Figure 40. Changing BGP next hop in router r2.

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



```
end

"Host: r2"
frr-pc(config-router)# neighbor 192.168.23.2 remote-as 200
frr-pc(config-router)# neighbor 192.168.23.2 next-hop-self
frr-pc(config-router)# end
frr-pc#[ ]"
```

Figure 41. Exiting from configuration mode.

Step 5. In router r3 terminal, configure IBGP to peer with router r3. All the steps are summarized in the following figure.

```
frr-pc# configure terminal
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.23.1 remote-as 200
frr-pc(config-router)# neighbor 192.168.23.1 next-hop-self
frr-pc(config-router)# end
frr-pc#
```

Figure 42. Configuring IBGP on router r3.

Step 6. Type the following command to verify the routing table of router r1. Router r1 has routes to the networks 192.168.3.0/24 and 192.168.4.0/24.

```
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-eth0, 01:43:43
B>* 192.168.2.0/24 [20/0] via 192.168.12.2, r1-eth1, 01:37:41
B>* 192.168.3.0/24 [20/0] via 192.168.12.2, r1-eth1, 00:50:48
B>* 192.168.4.0/24 [20/0] via 192.168.12.2, r1-eth1, 00:01:24
C>* 192.168.12.0/30 is directly connected, r1-eth1, 01:43:43
frr-pc#
```

Figure 43. Displaying the routing table of router r1.

Step 7. In host h1 terminal, perform a connectivity between host h1 and host h4 by issuing the command shown below. To stop the test, press Ctrl+c. The result will show a successful connectivity test.

```
ping 192.168.4.10
```

```
root@frr-pc:~# ping 192.168.4.10
PING 192.168.4.10 (192.168.4.10) 56(84) bytes of data.
64 bytes from 192.168.4.10: icmp_seq=1 ttl=60 time=0.908 ms
64 bytes from 192.168.4.10: icmp_seq=2 ttl=60 time=0.120 ms
64 bytes from 192.168.4.10: icmp_seq=3 ttl=60 time=0.111 ms
64 bytes from 192.168.4.10: icmp_seq=4 ttl=60 time=0.114 ms
^C
--- 192.168.4.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 56ms
rtt min/avg/max/mdev = 0.111/0.313/0.908/0.343 ms
root@frr-pc:~#
```

Figure 44. Connectivity test using ping command.

4 Perform IP spoofing and DoS attack

In this section, host h1 will spoof the IP address of host h4 and perform a DoS attack on it. Host h1 will send network traffic to hosts h2 and h3 with the source IP address of host h4. Thus, when hosts h2 and h3 receive the network traffic, they will reply to the source, i.e., host h4.

Step 1. Type the following command on h1 terminal. Host h1 is compromised and it will spoof the IP address of host h4 to perform a DoS against it. To do so, h1 sets an interface to the IP address of h4 first.

```
ifconfig lo 192.168.4.10
```

```
"Host: h1"
root@frrr-pc:~# ifconfig lo 192.168.4.10
root@frrr-pc:~#
```

Figure 45. Assigning an IP address to the loopback interface.

Step 2. Type the following command on host h4 terminal. The command `tcpdump` allows you to capture the network traffic. The `-i` option allows you to specify the interface to be monitored (*h4-eth0*).

```
tcpdump -i h4-eth0
```

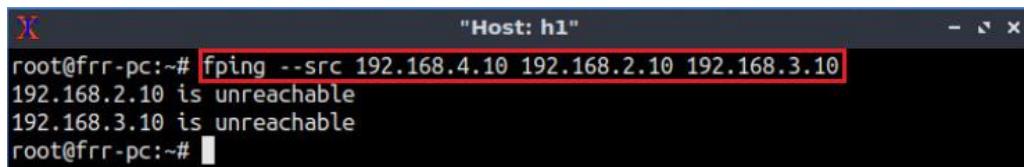
```
"Host: h4"
root@frrr-pc:~# tcpdump -i h4-eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on h4-eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
```

Figure 46. Capturing packets on interface h4-eth0.

Consider Figure 46. Currently, there is no network traffic directed on interface *eth0* of host h4. After launching the DoS attack from host h1, you will notice the network traffic redirected to host h4 using the `tcpdump` command.

Step 3. In host h1 terminal, issue the command shown below. The command used is `fping`. This command allows host h1 to ping multiple hosts. The `--src` option is followed by the source IP address. In this case, host h4 is configured with the source IP address (192.168.4.10). Then, the destinations IP addresses are specified, i.e. host h2 (192.168.2.10) and host h3 (192.168.3.10).

```
fping --src 192.168.4.10 192.168.2.10 192.168.3.10
```

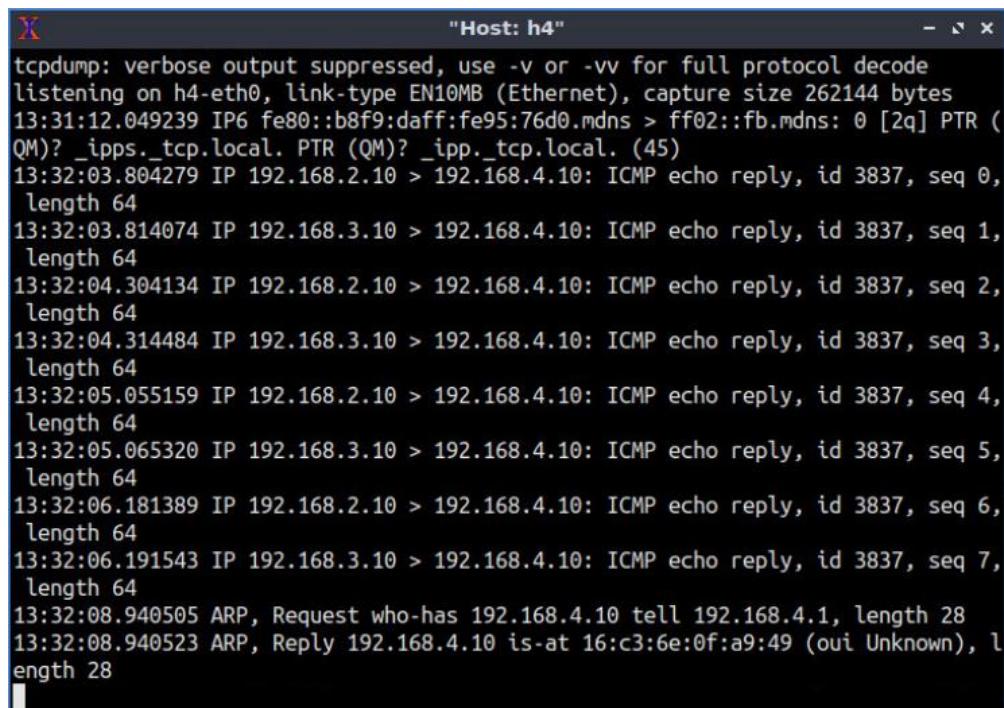


```
"Host: h1"
root@frr-pc:~# fping --src 192.168.4.10 192.168.2.10 192.168.3.10
192.168.2.10 is unreachable
192.168.3.10 is unreachable
root@frr-pc:~#
```

Figure 47. Pinging hosts h2 and h3 from host h1 via the source IP address 192.168.4.10.

Consider Figure 47. The two hosts h2 and h3 are unreachable since they will not reply to host h1. Instead, they will reply to the source IP address 192.168.4.10 which is host h4. This is how host h1 performs a DoS attack using different hosts.

Step 4. In host h4 terminal, observe the network traffic redirected from host h2 (192.168.2.10) and host h3 (192.168.3.10) towards host h4 (192.168.4.10).



```
"Host: h4"
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on h4-eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
13:31:12.049239 IP6 fe80::b8f9:daff:fe95:76d0.mdns > ff02::fb.mdns: 0 [2q] PTR (QM)? _ipps._tcp.local. PTR (QM)? _ipp._tcp.local. (45)
13:32:03.804279 IP 192.168.2.10 > 192.168.4.10: ICMP echo reply, id 3837, seq 0, length 64
13:32:03.814074 IP 192.168.3.10 > 192.168.4.10: ICMP echo reply, id 3837, seq 1, length 64
13:32:04.304134 IP 192.168.2.10 > 192.168.4.10: ICMP echo reply, id 3837, seq 2, length 64
13:32:04.314484 IP 192.168.3.10 > 192.168.4.10: ICMP echo reply, id 3837, seq 3, length 64
13:32:05.055159 IP 192.168.2.10 > 192.168.4.10: ICMP echo reply, id 3837, seq 4, length 64
13:32:05.065320 IP 192.168.3.10 > 192.168.4.10: ICMP echo reply, id 3837, seq 5, length 64
13:32:06.181389 IP 192.168.2.10 > 192.168.4.10: ICMP echo reply, id 3837, seq 6, length 64
13:32:06.191543 IP 192.168.3.10 > 192.168.4.10: ICMP echo reply, id 3837, seq 7, length 64
13:32:08.940505 ARP, Request who-has 192.168.4.10 tell 192.168.4.1, length 28
13:32:08.940523 ARP, Reply 192.168.4.10 is-at 16:c3:6e:0f:a9:49 (oui Unknown), length 28
```

Figure 48. Monitoring network traffic on host h4.

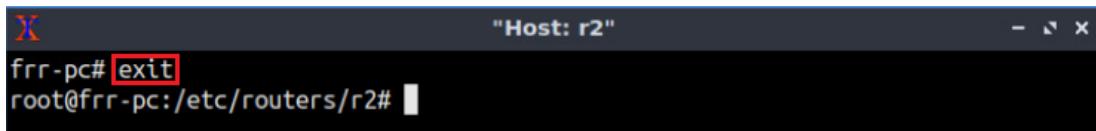
To interrupt capturing the network traffic on interface *eth0* of host h4 press **[Ctrl+c]**.

5 Mitigate DDoS attack by using IP source filtering

In this section, you will configure the ISP (router r2) to filter network traffic of Campus-1 based on their source IP addresses. Thus, mitigating IP spoofing and its possible attacks, such as DoS. To filter network traffic based on the source IP address, iptables utility will be used as FRR doesn't support this feature. Iptables is a command line program used to configure packet filtering on Linux operating systems.

Step 1. In router r2 terminal, type the following command to exit the vtysh session:

```
exit
```

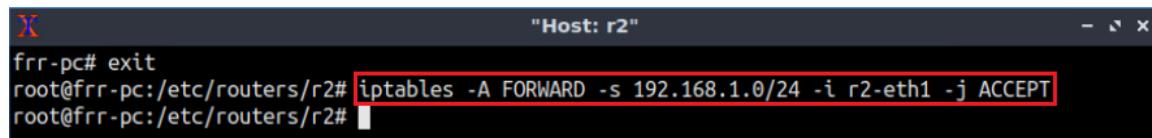


```
frr-pc# exit
```

Figure 49. Exiting the vtysh session.

Step 2. Type the following command on router r2 terminal to add a filtering rule. The option `-A FORWARD` specifies that the rule added corresponds to incoming connections that are not being delivered locally. The option `-s` is used to specify the source IP address of the traffic. The option `-i` is used to specify the input interface receiving the traffic (`r2-eth1`). The option `-j` is to specify what to do if the packet matches. Briefly, in this command we are inserting a rule to accept all incoming packets on interface `r2-eth1` (facing Campus-1) having a source IP address that belongs to the subnet `192.168.1.0/24`.

```
iptables -A FORWARD -s 192.168.1.0/24 -i r2-eth1 -j ACCEPT
```



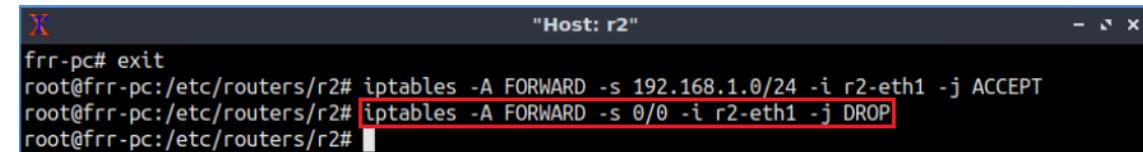
```
frr-pc# exit
```

```
root@frr-pc:/etc/routers/r2# iptables -A FORWARD -s 192.168.1.0/24 -i r2-eth1 -j ACCEPT
```

Figure 50. Adding a filtering rule on router r2.

Step 3. Similarly, to add another filtering rule on router r2, type the following command. The `-s 0/0` option matches all IP addresses. Briefly, we are dropping all incoming packets on interface `r2-eth1`.

```
iptables -A FORWARD -s 0/0 -i r2-eth1 -j DROP
```



```
frr-pc# exit
```

```
root@frr-pc:/etc/routers/r2# iptables -A FORWARD -s 192.168.1.0/24 -i r2-eth1 -j ACCEPT
```

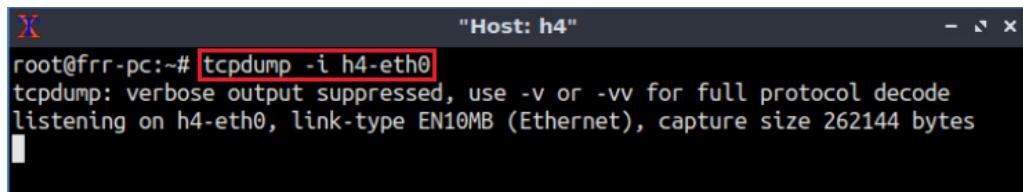
```
root@frr-pc:/etc/routers/r2# iptables -A FORWARD -s 0/0 -i r2-eth1 -j DROP
```

Figure 51. Adding a filtering rule on router r2.

After adding two filters on router r2, all incoming packets on interface `r2-eth1` will be permitted if their IP address belongs to the subnet `192.168.1.0/24`. Otherwise, the packets will be dropped and not forwarded to their destination.

Step 4. We will launch another DoS attack from host h1 on host h4 and validate that the attack is mitigated. On host h4 terminal, type the following command to capture the network traffic on interface `h4-eth0`.

```
tcpdump -i h4-eth0
```

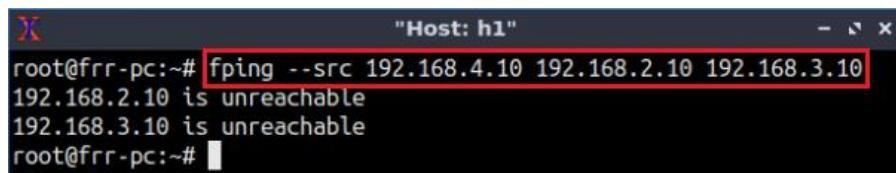


```
root@frr-pc:~# tcpdump -i h4-eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on h4-eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
```

Figure 52. Capturing packets on interface h4-eth0.

Step 5. On host h1 terminal, use the command `fping` to ping hosts h2 (192.168.2.10) and h3 (192.168.3.10) from the source IP address 192.168.4.10.

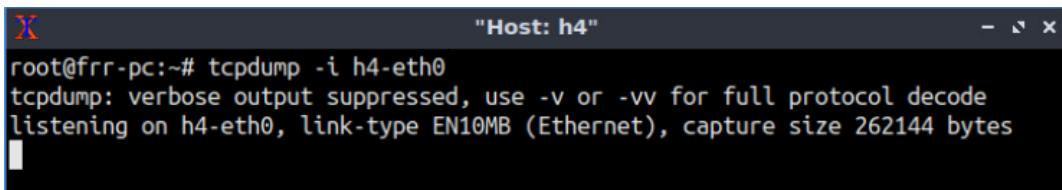
```
fping --src 192.168.4.10 192.168.2.10 192.168.3.10
```



```
root@frr-pc:~# fping --src 192.168.4.10 192.168.2.10 192.168.3.10
192.168.2.10 is unreachable
192.168.3.10 is unreachable
root@frr-pc:~#
```

Figure 53. Pinging hosts h2 and h3 from host h1 via the source IP address 192.168.4.10.

Step 6. In host h4 terminal, observe that even after the DoS attack was performed from host h1, host h4 did not receive any packet. Thus, the ISP (router r2) was able to filter the network traffic based on the source IP address and mitigate IP spoofing attacks.



```
root@frr-pc:~# tcpdump -i h4-eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on h4-eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
```

Figure 54. Monitoring network traffic on host h4.

To interrupt capturing the network traffic on interface *eth0* of host h4 press `Ctrl+c`.

This concludes Lab 12. Stop the emulation and then exit out of MiniEdit.

References

1. A. Tanenbaum, D. Wetherall, "Computer networks", 5th Edition, Pearson, 2012.
2. MANRS, "Anti-Spoofing". [Online]. Available: <https://www.manrs.org/isps/guide/antispoofing/>
3. C. Douligeris, A. Mitrokotsa, "DDoS attacks and defense mechanisms: classification and state-of-the-art", [Online]. Available: <https://reader.elsevier.com/reader/sd/pii/S1389128603004250?token=825EB28028CD30A735627A835E87DF874474EF4072D273D6C9F0B0B58712597778BFDF230132C2FA8A8E082D370CAE51>
4. Ciscopress, "BGP Fundamentals". [Online]. Available: <https://www.ciscopress.com/articles/article.asp?p=2756480&seqNum=6>
5. Netfilter, "The netfilter.org project". [Online]. Available: <https://netfilter.org/>



BORDER GATEWAY PROTOCOL

Lab 13: BGP Hijacking

Document Version: **03-12-2020**



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	4
1.1 BGP overview	4
1.2 BGP hijacking.....	4
1.3 BGP hijacking mitigation techniques	5
1.3.1 Using RPKI to validate route origins	5
1.3.2 BGP prefix filters	6
2 Lab topology.....	7
2.1 Lab settings.....	8
2.2 Open the topology	9
2.3 Load zebra daemon and verify configuration	12
3 Configure BGP on routers	16
4 Perform BGP hijacking	21
5 Mitigate BGP hijacking by using IP prefix filtering.....	24
References	27

Overview

This lab presents Border Gateway Protocol (BGP) hijacking attack that occurs on the Internet between different Autonomous Systems (ASes). In this lab, a compromised host will hijack the Internet Protocol (IP) address of a victim and advertise this address to its BGP routers. Thus, the network traffic destined to the victim will be rerouted to the attacker. The goal of this lab is to configure the Internet Service Provider (ISP) to mitigate BGP hijacking attacks by applying IP prefix filters on the network traffic of its customers.

Objectives

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

1. Configure BGP as the main protocol between ASes.
2. Understand and perform BGP hijacking.
3. Understand BGP hijacking mitigation techniques.
4. Apply IP prefix filters to counter BGP hijacking.

Lab settings

The information in Table 1 provides the credentials to access Client1 machine.

Table 1. Credentials to access Client1 machine.

Device	Account	Password
Client1	admin	password

Lab roadmap

This lab is organized as follows:

1. Section 1: Introduction.
2. Section 2: Lab topology.
3. Section 3: Configure BGP on routers.
4. Section 4: BGP hijacking.
5. Section 5: BGP hijacking mitigation using IP prefix filtering.

1 Introduction

1.1 BGP overview

BGP is an exterior gateway protocol designed to exchange routing and reachability information among ASes on the Internet. BGP is relevant to network administrators of large organizations which connect to one or more ISPs, as well as to ISPs who connect to other network providers. In terms of BGP, an AS is referred to as a routing domain, where all networked systems operate common routing protocols and are under the control of a single administration¹.

BGP is a form of distance vector protocol. It requires each router to maintain a table, which stores the distance and the output interface (i.e., vector) to remote networks. BGP makes routing decisions based on paths, network policies, or rule set configured by a network administrator and is involved in making core routing decisions¹.

Two routers that establish a BGP connection are referred to as BGP peers or neighbors. BGP sessions run over Transmission Control Protocol (TCP). If a BGP session is established between two neighbors in different ASes, the session is referred to as an External BGP (EBGP) session. If the session is established between two neighbors in the same AS, the session is referred to as Internal BGP (IBGP)¹. Figure 1 shows a network running BGP protocol. Routers that exchange information within the same AS use IBGP, while routers that exchange information between different ASes use EBGP.

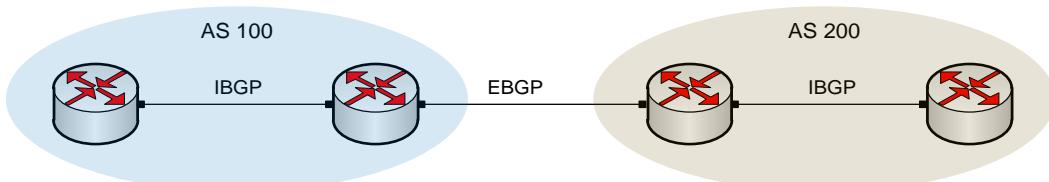


Figure 1. Routers that exchange information within the same AS use IBGP, while routers that exchange information between different ASes use EBGP.

1.2 BGP hijacking

BGP exchanges routing and reachability information among ASes. By default, when routers that have established BGP peering relationship trust each other. Consequently, any IP prefix announced by a router is accepted by its neighbors. However, the Internet is not always ideal, unauthorized network can originate IP prefix owned by other networks to divert traffic for those prefixes towards the unauthorized network². This process is known as BGP hijacking.

Consider Figure 2. The IP address 192.168.3.10 matches the networks 192.168.3.0/25 advertised by router r1, and 192.168.3.0/24 advertised by router r3. To ping the network 192.168.3.10, router r2 prefers the route advertised by router r1 (hijacking router) over

router r3 (legitimate router). This decision is made since router r1 advertises a more specific announcement (/25) than router r3 (/24).

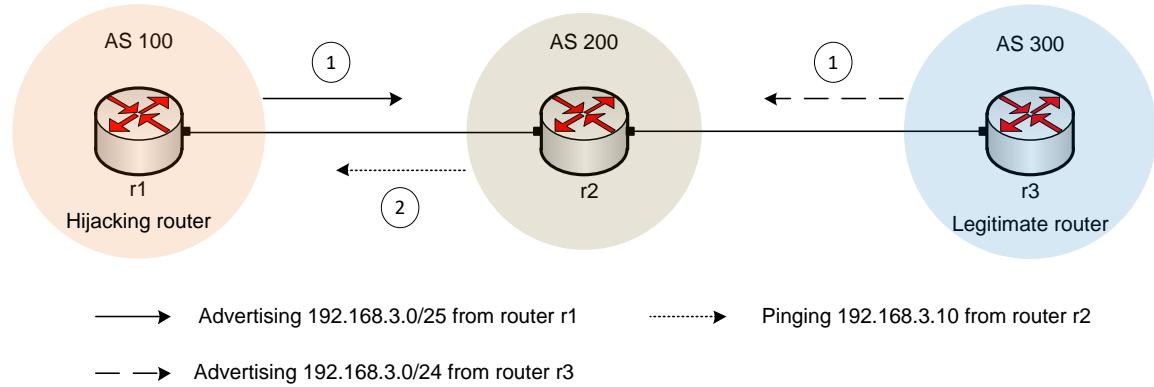


Figure 2. BGP hijacking using specific IP prefix advertisement.

Consider Figure 3. Router r1 (hijacking router) and router r4 (legitimate router) advertise the same network (192.168.4.0/24). To ping the IP address 192.168.4.10, router r2 prefers the route advertised by router r1 (hijacking router) over router r4 (legitimate router). This decision is made since the path to router r1 is shorter than that to router r4.

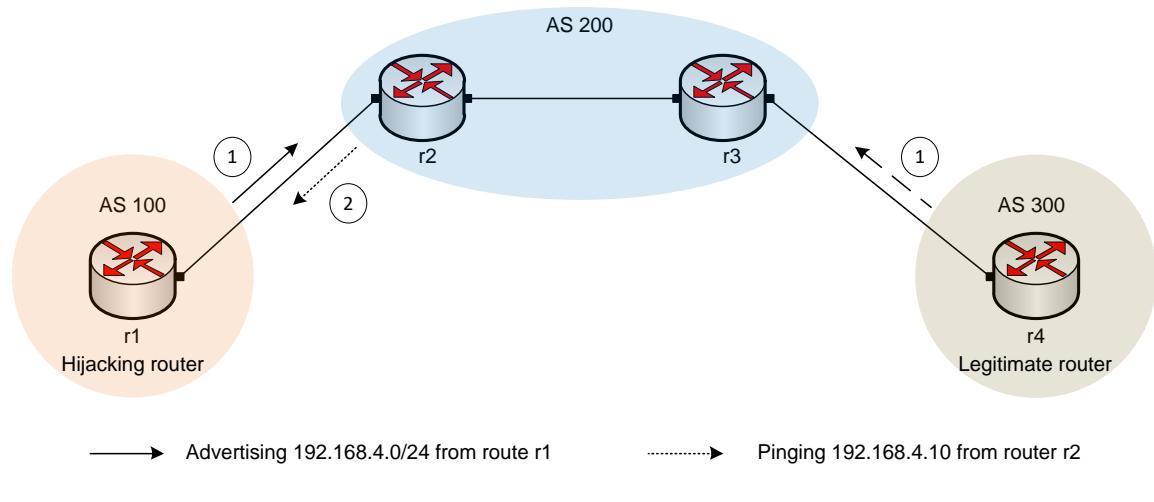


Figure 3. BGP hijacking using shorter path advertisement.

1.3 BGP hijacking mitigation techniques

Mutually Agreed Norms for Routing Security (MANRS) is a global initiative, supported by the Internet Society, that provides crucial fixes to reduce the most common routing threats. MANRS has many recommendations to prevent propagation of incorrect routing information, such as Resource Public Key Infrastructure (RPKI)³.

1.3.1 Using RPKI to validate route origins

Since any route can be originated and announced by any random network, there needs to be a method to manage BGP advertisements. RPKI is a cryptographic method to support improved security of Internet routing. It enables an entity to verifiably assert that it is the legitimate holder of a set of IP addresses or a set of AS numbers⁴.

Consider Figure 4. When router r2 receives route advertisements (1), it contacts the RPKI-enabled server to validate route advertisements. This server in turns sends if the received routes are valid or not based on stored cryptographic information about each entity (2). After receiving the response from the RPKI-enabled server, router r2 will ping the network 192.168.3.0/24 via the legitimate router (router r3).

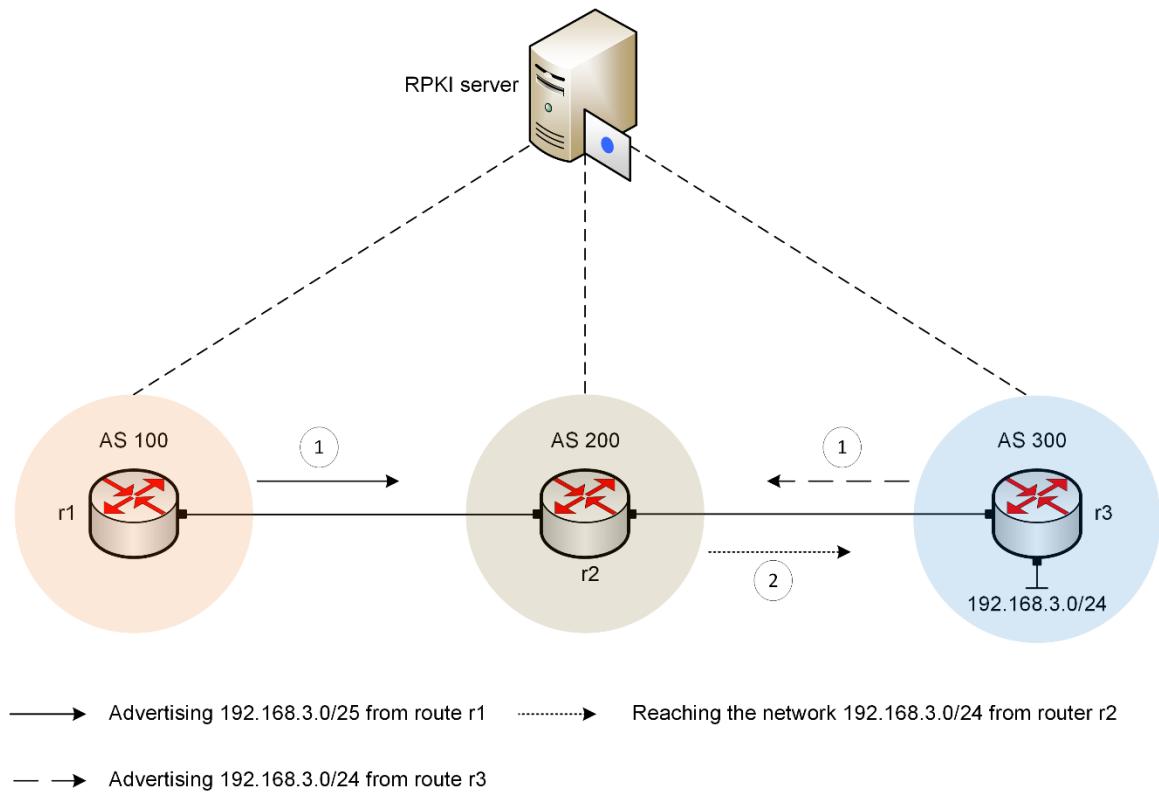


Figure 4. Using RPKI to validate IP prefix advertisements

1.3.2 BGP prefix filters

A router can limit the number of BGP route advertisements by configuring IP prefix filters. Prefix filtering can be applied to inbound and outbound advertisements.

Consider Figure 5. The network 192.168.1.0/24 belongs to its Customer in AS 100. The ISP applies BGP prefix filters to allow only the advertisement of this network. Thus, if router r1 tries to hijack other IP prefixes, the ISP will prevent such action.

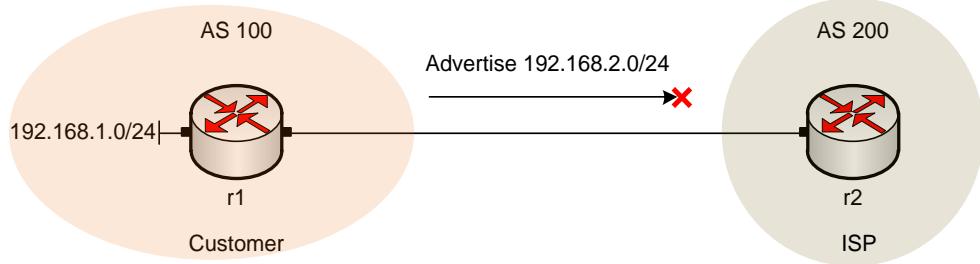


Figure 5. Applying IP prefix filters to prevent hijacked IP prefixes.

Although using RPKI is highly recommended to validate advertised IP prefixes, in this lab, we will apply BGP prefix filters as they are more feasible and easier to be implemented.

2 Lab topology

Consider Figure 6. The topology consists of three ASes. The ISP, consisting of routers r3 and r4, provides Internet service to the Campus-1 (router r1) and Campus-2 (router r2) networks. The Autonomous System Numbers (ASNs) assigned to Campus-1, ISP, and Campus-2 are 100, 200, and 300, respectively. The ISP communicates with the Campus networks via EBGP routing protocol, and the routers within the ISP communicate using IBGP. Host h1 in Campus-1 hijacks the network address assigned to Campus-2. Thus, all the traffic destined to Campus-2 and passing through router r3 will be rerouted to router r1.

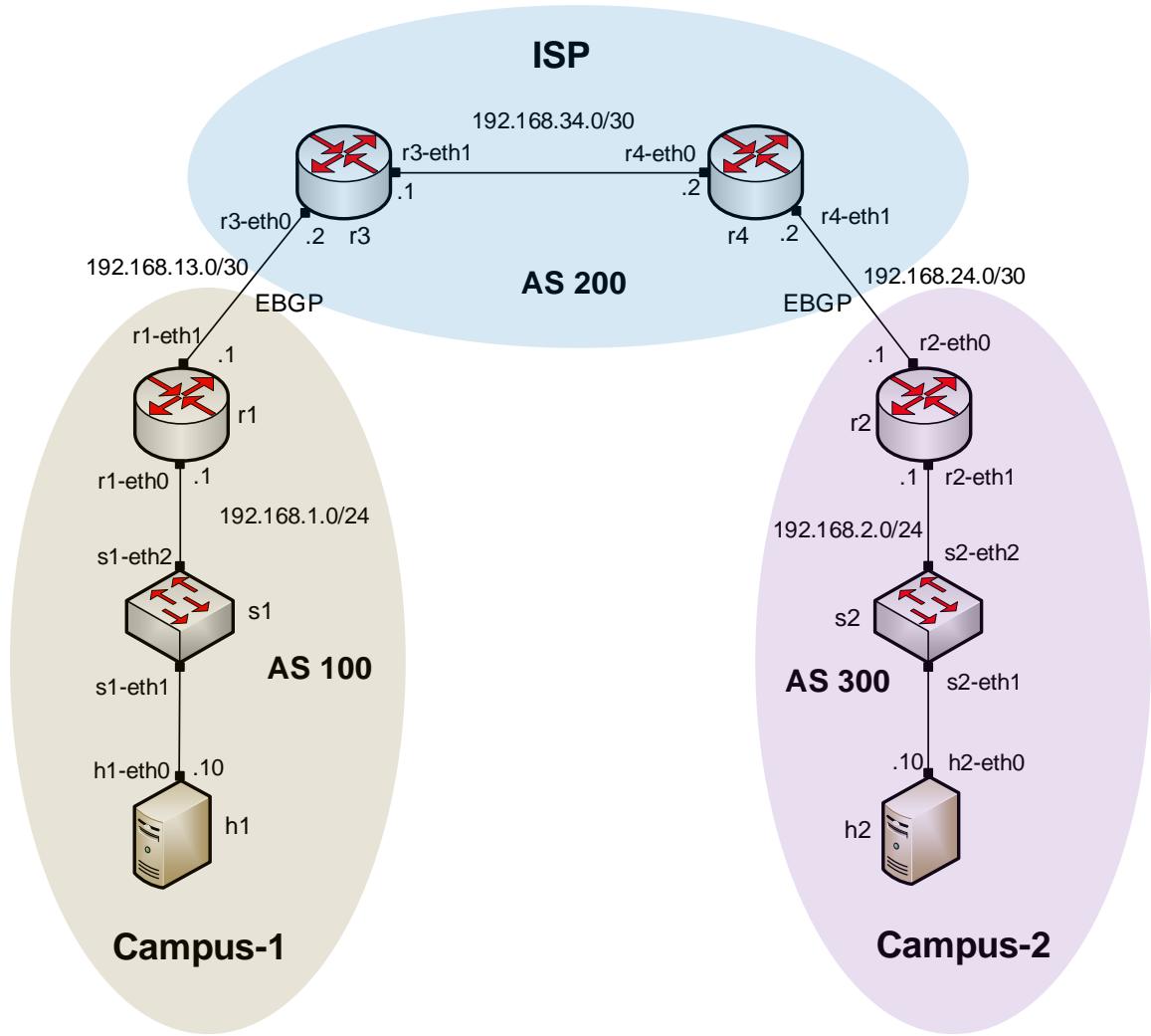


Figure 6. Lab topology.

2.1 Lab settings

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

Table 2. Topology information.

Device	Interface	IPV4 Address	Subnet	Default gateway
r1 (Campus-1)	r1-eth0	192.168.1.1	/24	N/A
	r1-eth1	192.168.13.1	/30	N/A
r2 (Campus-2)	r2-eth0	192.168.2.1	/24	N/A
	r2-eth1	192.168.24.1	/30	N/A
r3 (ISP)	r3-eth0	192.168.13.2	/30	N/A
	r3-eth1	192.168.34.1	/30	N/A

r4 (ISP)	r4-eth0	192.168.34.2	/30	N/A
	r4-eth1	192.168.24.2	/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

Step 1. Start by launching Miniedit by clicking on Desktop's shortcut. When prompted for a password, type `password`.



Figure 7. Miniedit shortcut.

Step 2. On Miniedit's menu bar, click on *File* then *open* to load the lab's topology. Open the *Lab13.mn* topology file stored in the default directory, */home/frr/BGP_Labs/lab13* and click on *Open*.

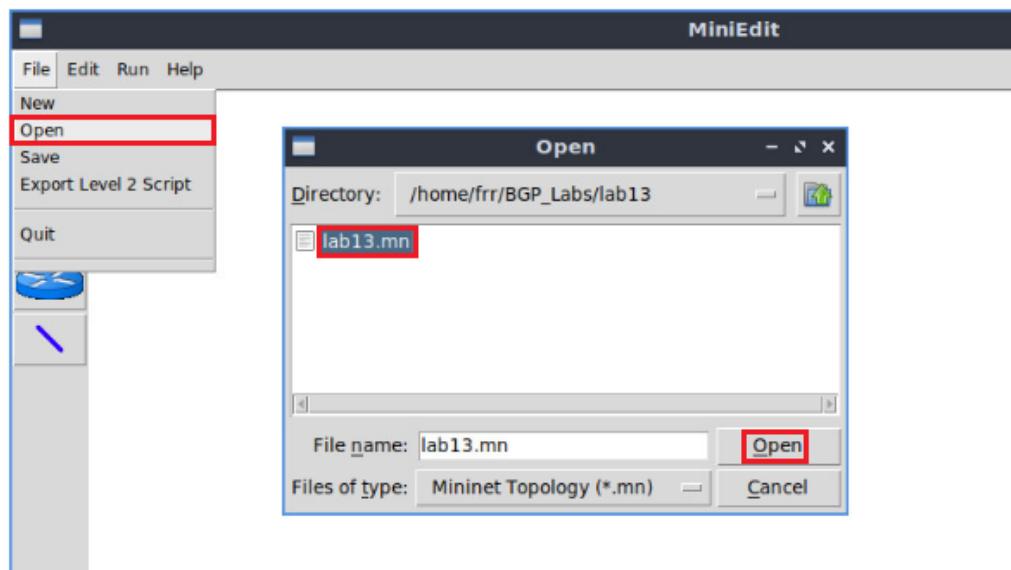


Figure 8. MiniEdit's open dialog.

At this point the topology is loaded with all the required network components. You will execute a script that will load the configuration of the routers.

Step 3. Open the Linux terminal.

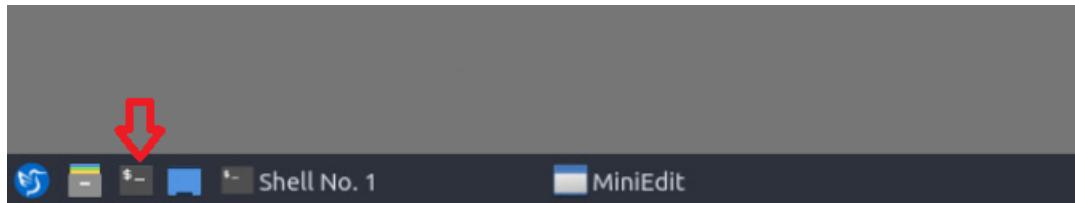


Figure 9. Opening Linux terminal.

Step 4. Click on the Linux's terminal and navigate into *BGP_Labs/lab13* 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 BGP_Labs/lab13
```

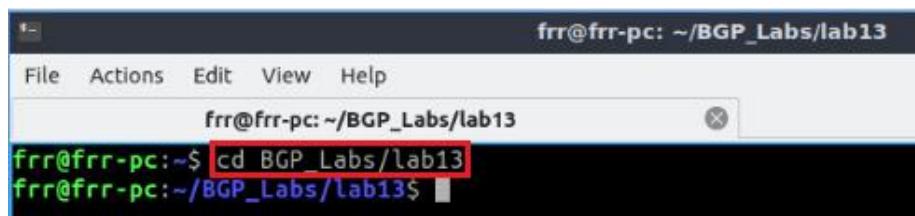


Figure 10. Entering the *BGP_Labs/lab13* directory.

Step 5. To execute the shell script, type the following command. The argument of the program corresponds to the configuration zip file that will be loaded in all the routers in the topology.

```
./config_loader.sh lab13_conf.zip
```

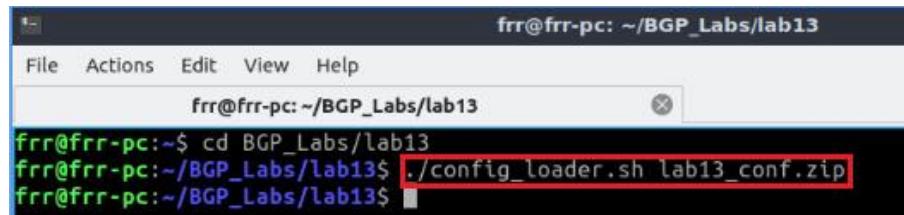
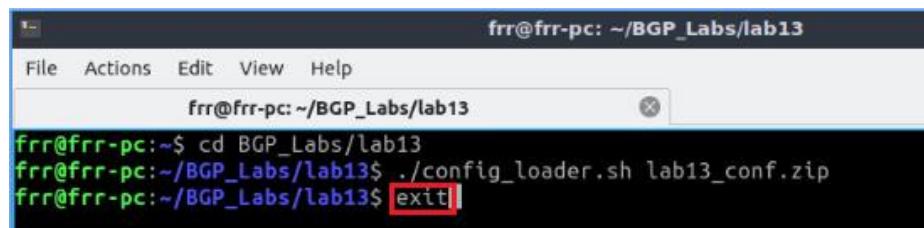


Figure 11. Executing the shell script to load the configuration.

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

```
exit
```



```
frr@frr-pc: ~/BGP_Labs/lab13
File Actions Edit View Help
frr@frr-pc:~/BGP_Labs/lab13
frr@frr-pc:~/BGP_Labs/lab13$ ./config_loader.sh lab13_conf.zip
frr@frr-pc:~/BGP_Labs/lab13$ exit
```

A screenshot of a terminal window titled "frr@frr-pc: ~/BGP_Labs/lab13". The window shows a command-line interface with several commands being entered and executed. The last command entered is "exit", which is highlighted with a red box.

Figure 12. Using `exit` to exit the terminal.

Step 7. At this point hosts h1 and h2 interfaces are configured. To proceed with the emulation, click on the *Run* button located in lower left-hand side.



Figure 13. Starting the emulation.

Step 8. In Mininet's terminal, i.e., the one launched when MiniEdit was started.

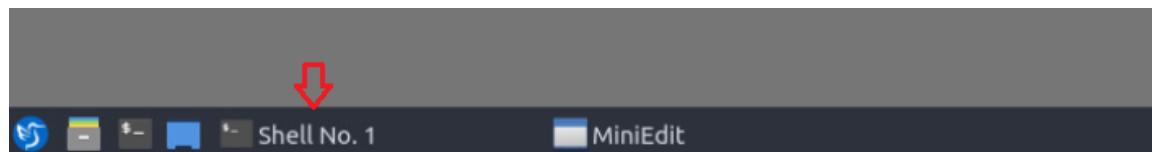
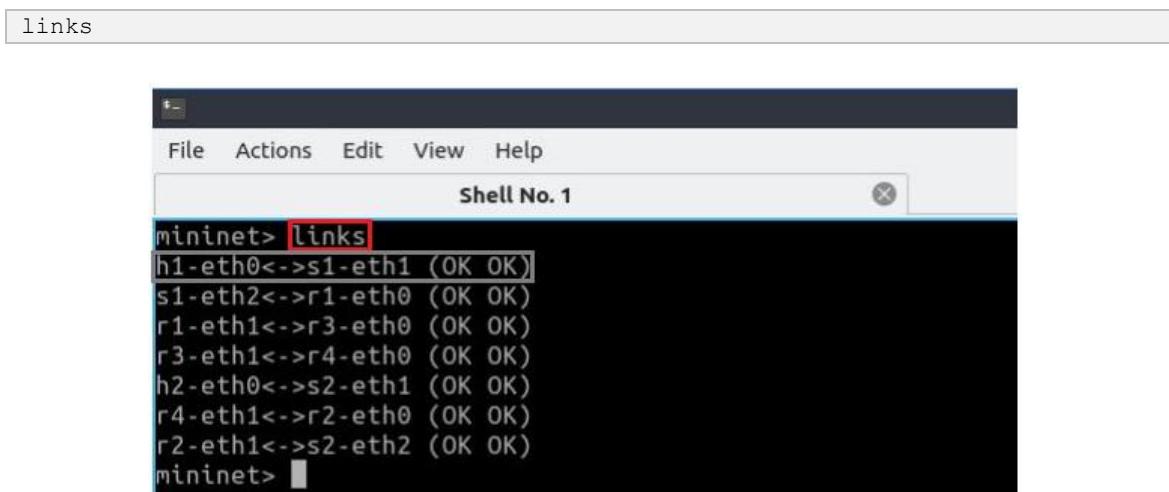


Figure 14. Opening Mininet's terminal.

Step 9. 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)
r1-eth1<->r3-eth0 (OK OK)
r3-eth1<->r4-eth0 (OK OK)
h2-eth0<->s2-eth1 (OK OK)
r4-eth1<->r2-eth0 (OK OK)
r2-eth1<->s2-eth2 (OK OK)
mininet>
```

A screenshot of a terminal window titled "Shell No. 1". The window shows the output of the "links" command. The output lists various network connections between interfaces like h1-eth0, s1-eth1, r1-eth0, etc. The first connection "h1-eth0<->s1-eth1" is highlighted with a gray box.

Figure 15. Displaying network interfaces.

In Figure 15, the link displayed within the gray box indicates that interface *eth0* of host h1 connects to interface *eth1* of switch s1 (i.e., *h1-eth0<->s1-eth1*).

2.3 Load zebra daemon and verify configuration

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

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.

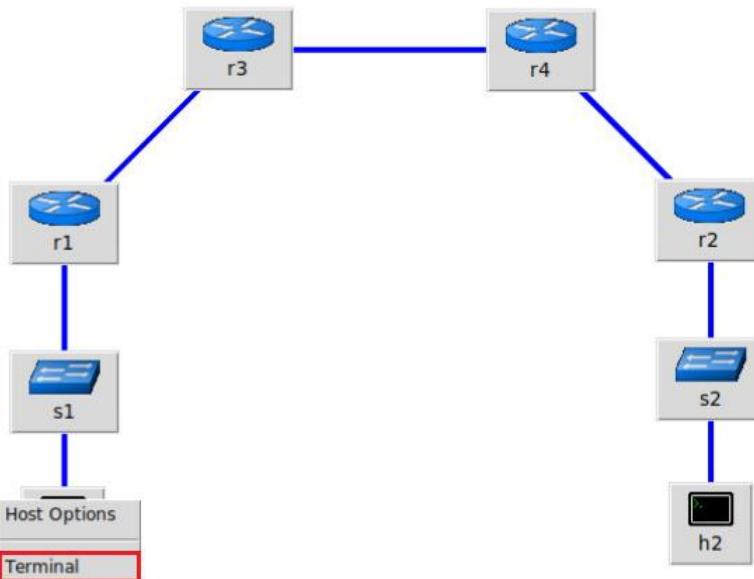


Figure 16. Opening a terminal on host h1.

Step 2. On 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.

```
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 17. Output of `ifconfig` command.

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

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

```

route
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 18. Output of `route` command.

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

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

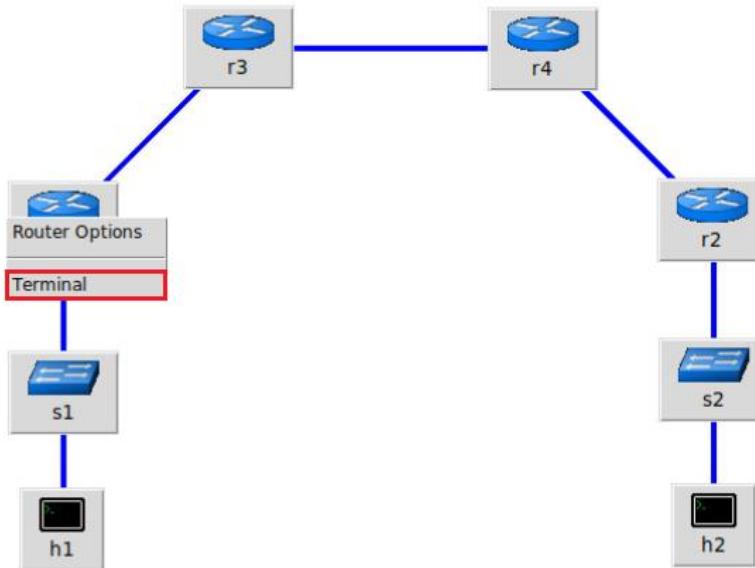


Figure 19. Opening a terminal on router r1.

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

```
zebra
```

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

Figure 20. Starting zebra daemon.

Step 7. After initializing zebra, vtysh is started in order to provide all the CLI commands defined by the daemons in a single shell. 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 21. 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 attached to router r2 (192.168.2.0/24) and router r4 (192.168.24.0/30, 192.168.34.0/30) as there is no routing protocol configured yet.

```
show ip route
```

```

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, 00:00:12
C>* 192.168.13.0/30 is directly connected, r1-eth1, 00:00:12
frr-pc# 
```

Figure 22. Displaying routing table of router r1.

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.

```

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-eth1, 00:00:06
C>* 192.168.24.0/30 is directly connected, r2-eth0, 00:00:06
frr-pc# 
```

Figure 23. Displaying routing table of router r2.

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

```

root@frr-pc:/etc/routers/r3# zebra
root@frr-pc:/etc/routers/r3# 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>* 192.168.13.0/30 is directly connected, r3-eth0, 00:00:06
C>* 192.168.34.0/30 is directly connected, r3-eth1, 00:00:06
frr-pc# 

```

Figure 24. Displaying routing table of router r3.

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

```

root@frr-pc:/etc/routers/r4# zebra
root@frr-pc:/etc/routers/r4# 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>* 192.168.24.0/30 is directly connected, r4-eth1, 00:00:06
C>* 192.168.34.0/30 is directly connected, r4-eth0, 00:00:06
frr-pc# 

```

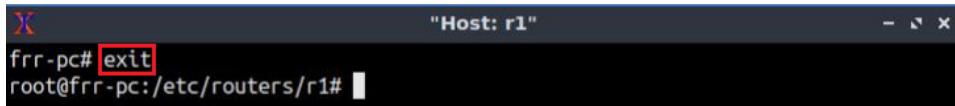
Figure 25. Displaying routing table of router r4.

3 Configure BGP on routers

In this section, you will configure EBGP on the routers that are hosted in different ASes. You will assign BGP neighbors to allow the routers to exchange BGP routes. Furthermore, routers r1 and r2 will advertise their Local Area Networks (LANs) via BGP. Therefore, router r3 and router r4 will receive route information about LAN 192.168.1.0/24 and 192.168.2.0/24, respectively.

Step 1. To configure BGP routing protocol, you need to enable the BGP daemon first. In router r1, type the following command to exit the vtysh session:

```
exit
```

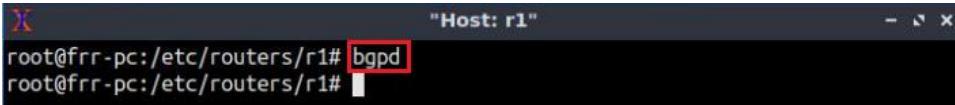


```
"Host: r1"
frr-pc# exit
root@frr-pc:/etc/routers/r1#
```

Figure 26. Exiting the vtysh session.

Step 2. Type the following command on router r1 terminal to enable and to start BGP routing protocol.

```
bgpd
```

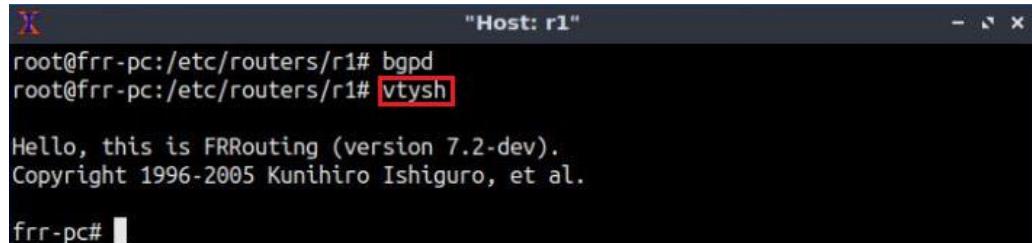


```
"Host: r1"
root@frr-pc:/etc/routers/r1# bgpd
root@frr-pc:/etc/routers/r1#
```

Figure 27. Starting BGP daemon.

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

```
vtysh
```



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

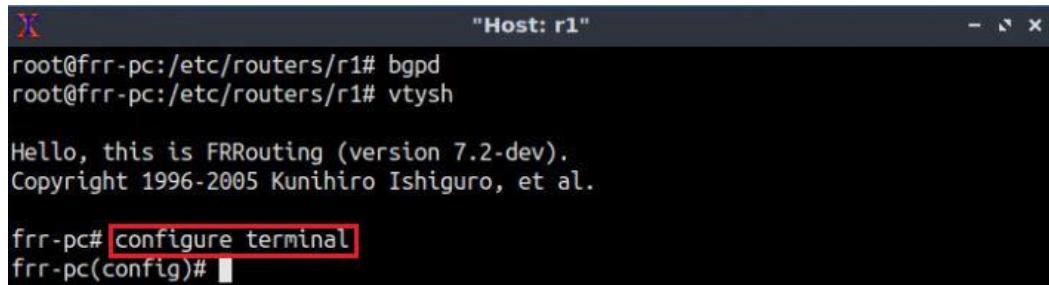
Hello, this is FRRouting (version 7.2-dev).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr-pc#
```

Figure 28. Starting vtysh on router r1.

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

```
configure terminal
```



```
"Host: r1"
root@frr-pc:/etc/routers/r1# bgpd
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 29. Enabling configuration mode on router r1.

Step 5. The ASN assigned for router r1 is 100. In order to configure BGP, type the following command:

```
router bgp 100
```



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

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

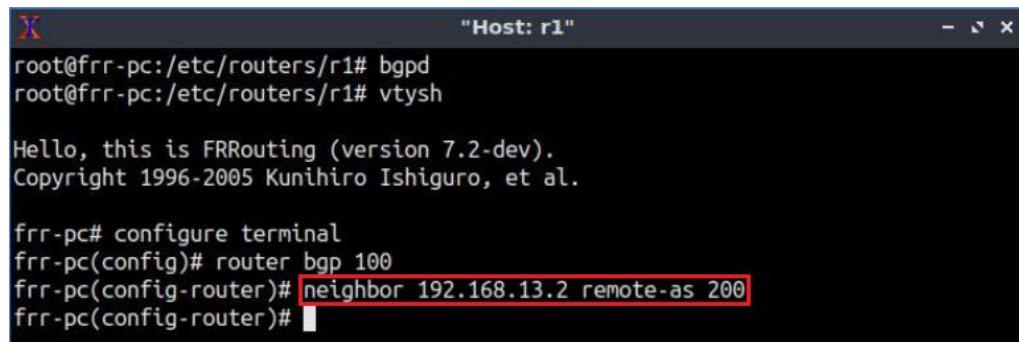
frr-pc# configure terminal
frr-pc(config)# router bgp 100
frr-pc(config-router)#

```

Figure 30. Configuring BGP on router r1.

Step 6. To configure a BGP neighbor to router r1 (AS 100), type the command shown below. This command specifies the neighbor IP address (192.168.13.2) and ASN of the remote BGP peer (AS 200).

```
neighbor 192.168.13.2 remote-as 200
```



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

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

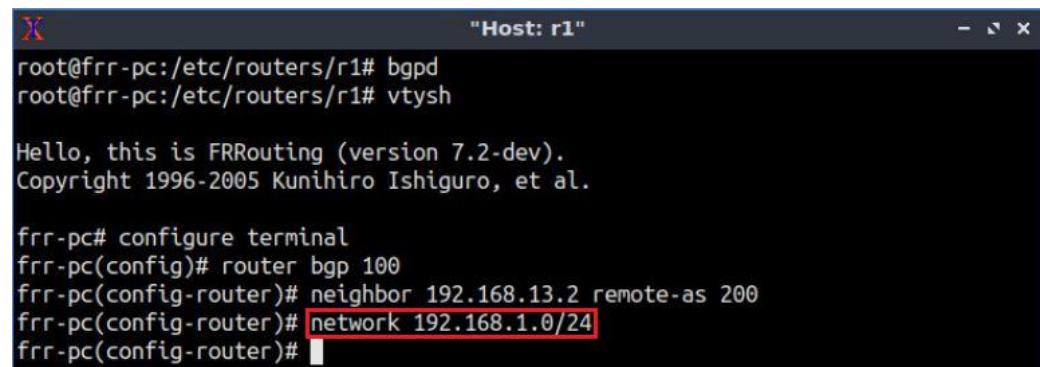
frr-pc# configure terminal
frr-pc(config)# router bgp 100
frr-pc(config-router)# neighbor 192.168.13.2 remote-as 200
frr-pc(config-router)#

```

Figure 31. Assigning BGP neighbor to router r1.

Step 7. In this step, router r1 will advertise the LAN 192.168.1.0/24 to router r3 through EBGP. To do so, issue the following command:

```
network 192.168.1.0/24
```



```
"Host: r1"
root@frr-pc:/etc/routers/r1# bgpd
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 bgp 100
frr-pc(config-router)# neighbor 192.168.13.2 remote-as 200
frr-pc(config-router)# network 192.168.1.0/24
frr-pc(config-router)#

```

Figure 32. Advertising a network on router r1.

Step 8. Type the following command to exit from the configuration mode.

```
end
```

```

root@frr-pc:/etc/routers/r1# bgpd
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 bgp 100
frr-pc(config-router)# neighbor 192.168.13.2 remote-as 200
frr-pc(config-router)# network 192.168.1.0/24
frr-pc(config-router)# end
frr-pc#

```

Figure 33. Exiting from configuration mode.

Step 9. Type the following command to verify BGP networks. You will observe the LAN network of router r1.

```

show ip bgp

frr-pc# show ip bgp
BGP table version is 1, local router ID is 192.168.13.1, vrf id 0
Default local pref 100, local AS 100
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexhop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop           Metric LocPrf Weight Path
*-> 192.168.1.0/24    0.0.0.0                  0        32768 i

Displayed 1 routes and 1 total paths
frr-pc#

```

Figure 34. Verifying BGP networks on router r1.

Step 10. Follow from step 1 to step 8 but with different metrics in order to configure BGP in router r2. All these steps are summarized in the following figure.

```

frr-pc# exit
root@frr-pc:/etc/routers/r2# bgpd
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 bgp 300
frr-pc(config-router)# neighbor 192.168.24.2 remote-as 200
frr-pc(config-router)# network 192.168.2.0/24
frr-pc(config-router)# end
frr-pc#

```

Figure 35. Configuring BGP on router r2.

Step 11. Follow from step 1 to step 8 but with different metrics in order to configure BGP in router r3. Additionally, router r3 will configure BGP *next-hop-self* so that the neighbor

192.168.34.2 (router r4) can reach the EBGP routes advertised by router r3, such as 192.168.1.0/24, through router r3. All these steps are summarized in the following figure.

The terminal window shows the following session:

```
frr-pc# exit  
root@frr-pc:/etc/routers/r3# bgpd  
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 bgp 200  
frr-pc(config-router)# neighbor 192.168.13.1 remote-as 100  
frr-pc(config-router)# neighbor 192.168.34.2 remote-as 200  
frr-pc(config-router)# neighbor 192.168.34.2 next-hop-self  
frr-pc(config-router)# end  
frr-pc#
```

Figure 36. Configuring BGP on router r3.

Step 12. Follow from step 1 to step 8 but with different metrics in order to configure BGP in router r4. Additionally, router r4 will configure BGP *next-hop-self* so that the neighbor 192.168.34.1 (router r3) can reach the EBGP routes advertised by router r4, such as 192.168.2.0/24, through router r4. All these steps are summarized in the following figure.

The terminal window shows the following session:

```
frr-pc# exit  
root@frr-pc:/etc/routers/r4# bgpd  
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 bgp 200  
frr-pc(config-router)# neighbor 192.168.24.1 remote-as 300  
frr-pc(config-router)# neighbor 192.168.34.1 remote-as 200  
frr-pc(config-router)# neighbor 192.168.34.1 next-hop-self  
frr-pc(config-router)# end  
frr-pc#
```

Figure 37. Configuring BGP on router r4.

Step 13. Type the following command to verify the routing table of router r1. The LAN of router r2 network (192.168.2.0/24) is advertised to router r1 through EBGP.

```
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, 01:30:22
B>* 192.168.2.0/24 [20/0] via 192.168.13.2, r1-eth1, 00:07:04
C>* 192.168.13.0/30 is directly connected, r1-eth1, 01:30:22
frr-pc#
```

Figure 38. Verifying the routing table of router r1.

4 Perform BGP hijacking

In this section, you will configure router r1 to hijack the network 192.168.2.0/24 corresponding to Campus-2 by advertising it to its BGP neighbors. Thus, router r3 will have a route to the network 192.168.2.0/24 through router r1 and will use this route to send all network traffic destined to Campus-2.

Step 1. On router r3 terminal, type the following command to verify BGP networks.

```
show ip bgp
```

```
"Host: r3"
frr-pc# show ip bgp
BGP table version is 2, local router ID is 192.168.34.1, vrf id 0
Default local pref 100, local AS 200
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexhop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*-> 192.168.1.0/24    192.168.13.1        0          0 100 i
*>i192.168.2.0/24    192.168.34.2        0         100        0 300 i

Displayed 2 routes and 2 total paths
frr-pc#
```

Figure 39. Verifying BGP networks in router r3.

Consider Figure 39. Router r3 can reach the networks 192.168.1.0/24 and 192.168.2.0/24 through the next hops 192.168.13.1 and 192.168.34.2, respectively.

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

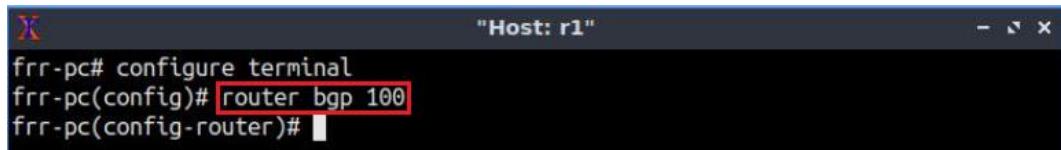
```
configure terminal
```

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

Figure 40. Enabling configuration mode in router r1.

Step 3. You will advertise the network 192.168.2.0/24 of Campus-2 from router r1 to all BGP neighbors. Type the following command to enter BGP configuration mode.

```
router bgp 100
```

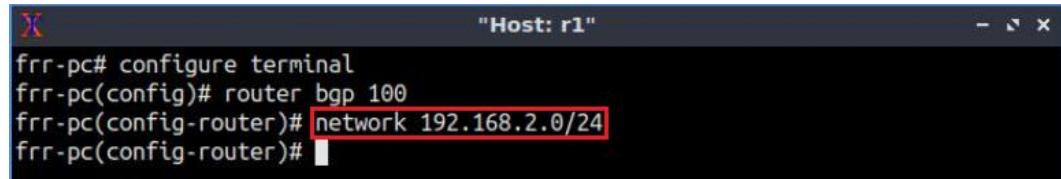


```
frr-pc# configure terminal
frr-pc(config)# router bgp 100
frr-pc(config-router)#
```

Figure 41. Configuring BGP on router r1.

Step 4. In this step, router r1 will hijack the network 192.168.2.0/24 of Campus-2 by advertising this network to all its BGP neighbors.

```
network 192.168.2.0/24
```

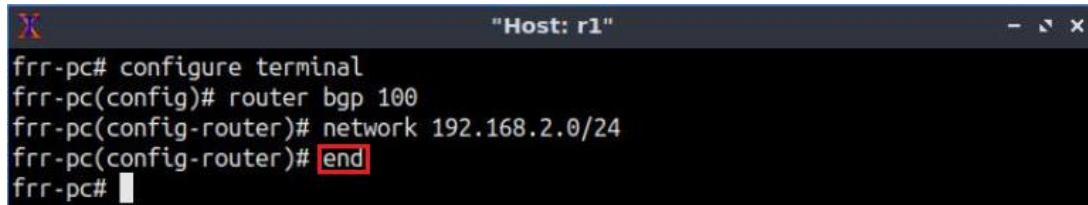


```
frr-pc# configure terminal
frr-pc(config)# router bgp 100
frr-pc(config-router)# network 192.168.2.0/24
frr-pc(config-router)#
```

Figure 42. Hijacking a network on router r1.

Step 5. Type the following command to exit from the configuration mode.

```
end
```



```
frr-pc# configure terminal
frr-pc(config)# router bgp 100
frr-pc(config-router)# network 192.168.2.0/24
frr-pc(config-router)# end
frr-pc#
```

Figure 43. Exiting from configuration mode.

Step 6. On router r3 terminal, type the following command to verify BGP networks.

```
show ip bgp
```

```
X "Host: r3"
frr-pc# show ip bgp
BGP table version is 3, local router ID is 192.168.34.1, vrf id 0
Default local pref 100, local AS 200
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*-> 192.168.1.0/24  192.168.13.1        0        0 100 i
*-> 192.168.2.0/24  192.168.13.1        0        0 100 i
* i                  192.168.34.2        0     100        0 300 i

Displayed 2 routes and 3 total paths
frr-pc#
```

Figure 44. Verifying BGP networks in router r3.

Consider Figure 44. Router r3 can reach the network 192.168.2.0/24 through the next hops 192.168.13.1 (router r1) and 192.168.34.2 (router r4). However, router r3 prefers to use the next hop 192.168.13.1 over 192.168.34.2 in its route to 192.168.2.0/24. This can be inferred from the characters **>*** next to the network address 192.168.2.0/24, which means that the corresponding next hop (192.168.13.1) is the best route to reach the network.

Step 7. On router r1 terminal,type the following command to exit from the configuration mode.

exit

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

Figure 45. Exiting the vtysh session.

Step 8. Type the following command on router r1 terminal. The command `tcpdump` allows you to capture the network traffic. The `-i` option allows you to specify the interface to be monitored (`r1-eth1`).

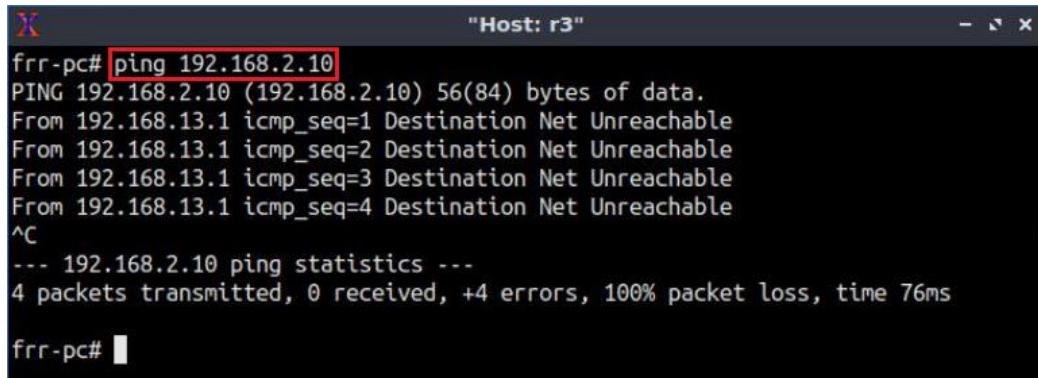
```
tcpdump -i r1-eth1
```

```
X "Host: r1"
root@frr-pc:/etc/routers/r1# tcpdump -i r1-eth1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on r1-eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
```

Figure 46. Capturing packets on interface *r1-eth1*.

Step 9. On router r3 terminal, type the following command to ping the IP address 192.168.2.10 corresponding to host h2.

```
ping 192.162.2.10
```

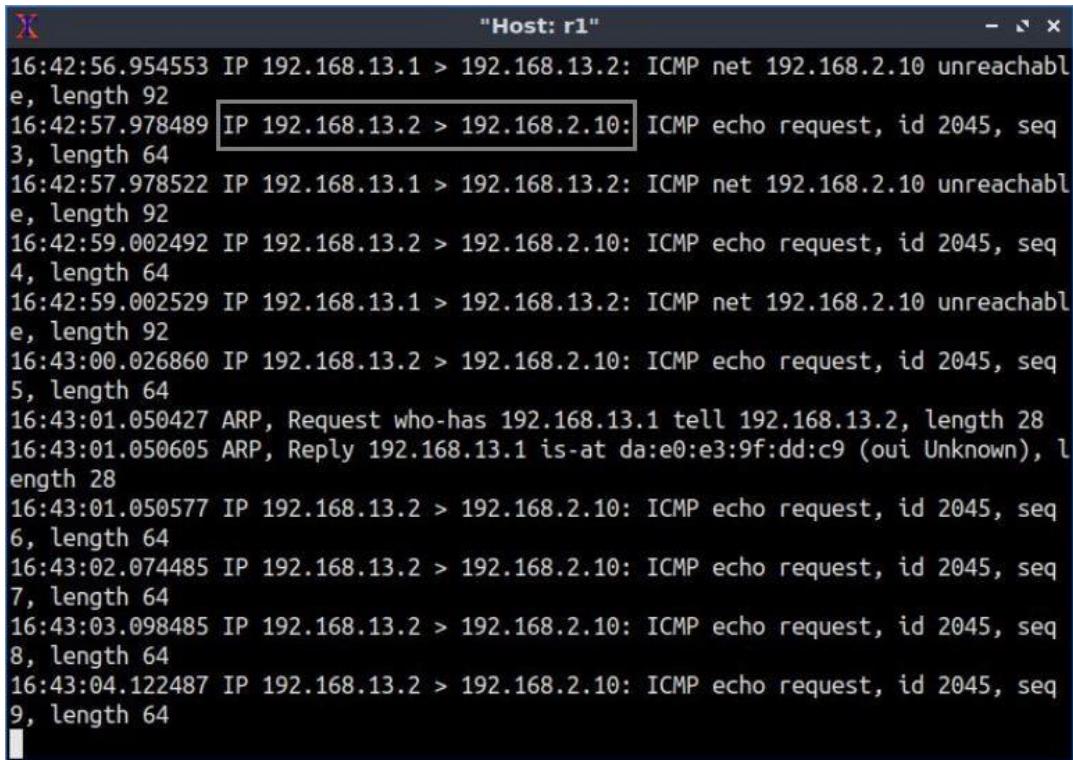


```
frr-pc# ping 192.168.2.10
PING 192.168.2.10 (192.168.2.10) 56(84) bytes of data.
From 192.168.13.1 icmp_seq=1 Destination Net Unreachable
From 192.168.13.1 icmp_seq=2 Destination Net Unreachable
From 192.168.13.1 icmp_seq=3 Destination Net Unreachable
From 192.168.13.1 icmp_seq=4 Destination Net Unreachable
^C
--- 192.168.2.10 ping statistics ---
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 76ms
frr-pc#
```

Figure 47. Pinging host h2 from router r3.

Consider Figure 47. Router r3 is pinging a host within the network 192.168.2.0/24. If the hijacking was successful, then the network traffic must be redirected to router r1, rather than router r4 (and eventually host h2).

Step 10. On router r1, notice how the network traffic that is sent from the IP addresses 192.168.13.2 (router r3) to 192.168.2.10 (host h2) is captured on router r1.



```
16:42:56.954553 IP 192.168.13.1 > 192.168.13.2: ICMP net 192.168.2.10 unreachable, length 92
16:42:57.978489 IP 192.168.13.2 > 192.168.2.10: ICMP echo request, id 2045, seq 3, length 64
16:42:57.978522 IP 192.168.13.1 > 192.168.13.2: ICMP net 192.168.2.10 unreachable, length 92
16:42:59.002492 IP 192.168.13.2 > 192.168.2.10: ICMP echo request, id 2045, seq 4, length 64
16:42:59.002529 IP 192.168.13.1 > 192.168.13.2: ICMP net 192.168.2.10 unreachable, length 92
16:43:00.026860 IP 192.168.13.2 > 192.168.2.10: ICMP echo request, id 2045, seq 5, length 64
16:43:01.050427 ARP, Request who-has 192.168.13.1 tell 192.168.13.2, length 28
16:43:01.050605 ARP, Reply 192.168.13.1 is-at da:e0:e3:9f:dd:c9 (oui Unknown), length 28
16:43:01.050577 IP 192.168.13.2 > 192.168.2.10: ICMP echo request, id 2045, seq 6, length 64
16:43:02.074485 IP 192.168.13.2 > 192.168.2.10: ICMP echo request, id 2045, seq 7, length 64
16:43:03.098485 IP 192.168.13.2 > 192.168.2.10: ICMP echo request, id 2045, seq 8, length 64
16:43:04.122487 IP 192.168.13.2 > 192.168.2.10: ICMP echo request, id 2045, seq 9, length 64
```

Figure 48. Monitoring network traffic on router r1.

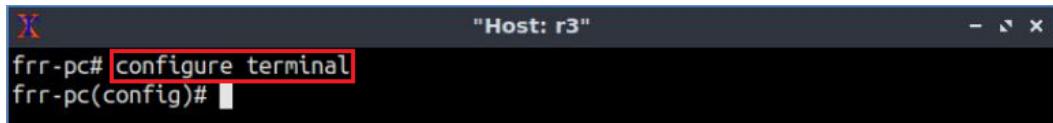
To interrupt capturing the network traffic on interface *eth1* of router r1 press **Ctrl+c**.

5 Mitigate BGP hijacking by using IP prefix filtering

In this section, you will configure IP prefix lists on routers r3 and r4 to restrict route advertisements from Campus-1 and Campus-2, respectively. Thus, mitigating BGP hijacking attacks.

Step 1. To enable router r3 into configuration mode, issue the following command.

```
configure terminal
```

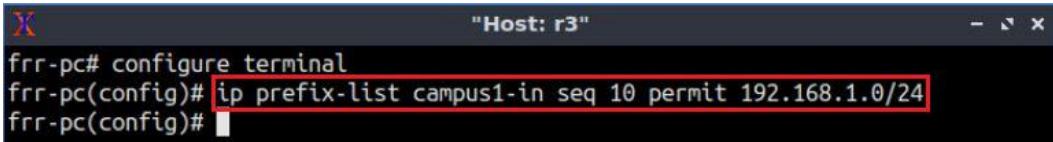


```
frr-pc# configure terminal
frr-pc(config)#
```

Figure 49. Enabling configuration mode on router r1.

Step 2. In this step, you will create an IP prefix list that permits the network 192.168.1.0/24. An IP prefix list must have a name (campus1-in), a permit or deny clause to allow or reject the packets that match the prefix list, and the network IP address to match on (192.168.1.0/24).

```
ip prefix-list campus1-in seq 10 permit 192.168.1.0/24
```

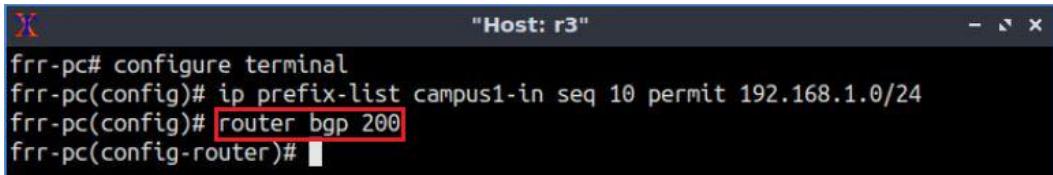


```
frr-pc# configure terminal
frr-pc(config)# ip prefix-list campus1-in seq 10 permit 192.168.1.0/24
frr-pc(config)#
```

Figure 50. Creating an IP prefix list.

Step 3. You will filter the route updates that are advertised by neighbor router r1 to router r3. Type the following command to enter BGP configuration mode:

```
router bgp 200
```

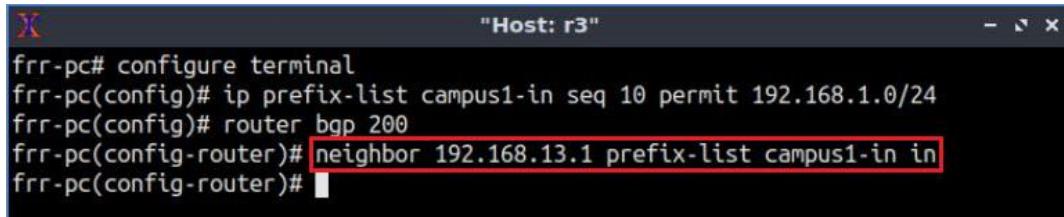


```
frr-pc# configure terminal
frr-pc(config)# ip prefix-list campus1-in seq 10 permit 192.168.1.0/24
frr-pc(config)# router bgp 200
frr-pc(config-router)#
```

Figure 51. Configuring BGP on router r3.

Step 4. Router r3 will apply the prefix list *campus1-in* to its neighbor 192.168.13.1 (router r1). Thus, only BGP advertisements corresponding to 192.168.1.0/24 will be permitted since they match the IP prefix list assigned. The option *in* corresponds to inbound traffic, i.e., the traffic coming to router r3.

```
neighbor 192.168.13.1 prefix-list campus1-in in
```



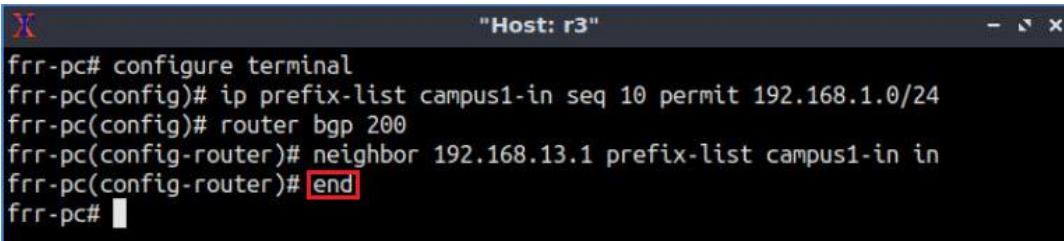
```
"Host: r3"
frr-pc# configure terminal
frr-pc(config)# ip prefix-list campus1-in seq 10 permit 192.168.1.0/24
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.13.1 prefix-list campus1-in in
frr-pc(config-router)#

```

Figure 52. Applying the IP prefix list to router r3 neighbor.

Step 5. Type the following command to exit from the configuration mode.

```
end
```



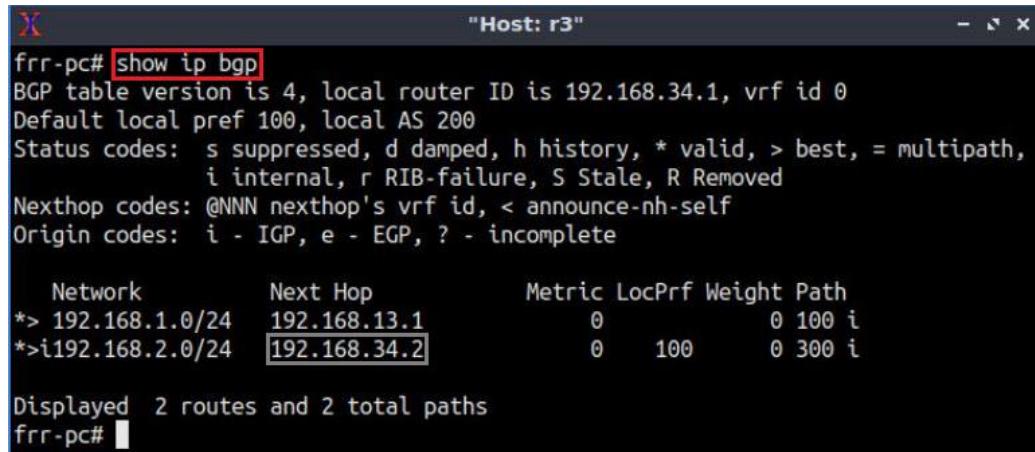
```
"Host: r3"
frr-pc# configure terminal
frr-pc(config)# ip prefix-list campus1-in seq 10 permit 192.168.1.0/24
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.13.1 prefix-list campus1-in in
frr-pc(config-router)# end
frr-pc#

```

Figure 53. Exiting from configuration mode.

Step 6. On router r3 terminal, type the following command to verify BGP networks.

```
show ip bgp
```



```
"Host: r3"
frr-pc# show ip bgp
BGP table version is 4, local router ID is 192.168.34.1, vrf id 0
Default local pref 100, local AS 200
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
              i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop           Metric LocPrf Weight Path
*-> 192.168.1.0/24    192.168.13.1        0          0 100 i
*->i192.168.2.0/24   [192.168.34.2]        0         100        0 300 i

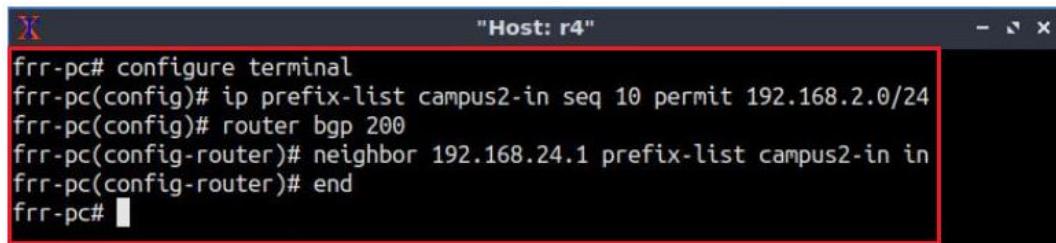
Displayed 2 routes and 2 total paths
frr-pc#

```

Figure 54. Verifying BGP networks in router r3.

Consider Figure 54. Router r3 has a route to the network 192.168.2.0/24 through the next hop 192.168.34.2 (router r4) only. Even though router r1 is still advertising the network 192.168.2.0/24, router r3 mitigates BGP hijacking through the configure IP prefix list filters.

Step 7. Follow from step 1 to step 5 to configure IP prefix list on router r4. Router r4 will configure the IP prefix list campus2-in to only permit advertising the network 192.168.2.0/24 from router r2.



```
frr-pc# configure terminal
frr-pc(config)# ip prefix-list campus2-in seq 10 permit 192.168.2.0/24
frr-pc(config)# router bgp 200
frr-pc(config-router)# neighbor 192.168.24.1 prefix-list campus2-in in
frr-pc(config-router)# end
frr-pc#
```

Figure 55. Configuring IP prefix list on router r4.

This concludes Lab 13. Stop the emulation and then exit out of MiniEdit.

References

1. A. Tanenbaum, D. Wetherall, "Computer networks," 5th Edition, Pearson, 2012.
2. V. Khare, Q. Ju, B. Zhang, "Concurrent Prefix Hijacks: Occurrence and Impacts," 2012. [Online]. Available: <https://dl.acm.org/doi/pdf/10.1145/2398776.2398780>
3. MANRS, "Filtering," [Online]. Available: <https://www.manrs.org/isps/guide/filtering/>
4. M. Lepinski, S. Kent, "An infrastructure to support secure internet routing," [Online]. Available: <https://tools.ietf.org/html/rfc6480>
5. Ciscopress, "CCNP Routing and Switching Portable Command Guide: Configuration of Redistribution," 2015. [Online]. Available: <https://www.ciscopress.com/articles/article.asp?p=2273507&seqNum=11>