UNIVERSITY OF
SOUTH CAROLINA

SOFTWARE DEFINED NETWORKING

Lab 7: Interconnection between legacy networks and SDN networks

Document Version: 05-31-2020

Award 1829698
"CyberTraining CIP: Cyberinfrastructure Expertise on High-throughput Networks for Big Science Data Transfers"

Contents

Overview

This lab is an introduction to integrating Software Defined Networking (SDN) networks with legacy networks. The focus in this lab is to understand SDN-IP, an application that comes with ONOS controller and allows the interconnection between legacy and SDN networks. The SDN-IP application allows the SDN network to understand the exchanged Border Gateway Protocol (BGP) information and communicate with other legacy networks through this protocol.

Objectives

By the end of this lab, the user will:

1. Understand how legacy networks operate.
2. Understand how SDN networks operate.
3. Configure BGP on legacy routers.
4. Integrate SDN and legacy networks through the SDN-IP application.
5. Verify the connectivity between the SDN and legacy networks.

Lab settings

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

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

Device	**Account**	**Password**
Client	admin	password

Lab roadmap

This lab is organized as follows:

1. Section 1: Introduction.
2. Section 2: Lab topology.
3. Section 3: Configuring BGP within legacy networks.
4. Section 4: Starting ONOS controller.
5. Section 5: Integrating SDN and legacy networks.
6. Section 6: Verifying the connectivity between the networks.

1 Introduction

Today's Internet is utterly dependent on routing protocols, such as BGP, so without a clean mechanism to integrate an OpenFlow/SDN and legacy/IP networks, the use of OpenFlow will remain restricted to isolated data center deployments. In this lab, you will understand how to integrate SDN and legacy networks, and how the SDN network translates the BGP information into OpenFlow entries[1].

1.1 Legacy networks

The Internet can be viewed as a collection of networks or Autonomous Systems (ASes) that are interconnected. An AS refers to a group of connected networks under the control of a single administrative entity or domain. Traditional networks depend on routing protocols to interconnect and share routing information. Such protocols are also referred to as control protocols and they are run by each device in the network[1].

BGP is the standard 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[1].

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) session[1].

Figure 1 shows two legacy networks, each in an AS. Each router runs its internal local algorithm (routing protocol) to communicate with other peers. The routing protocol used between ASes is BGP.

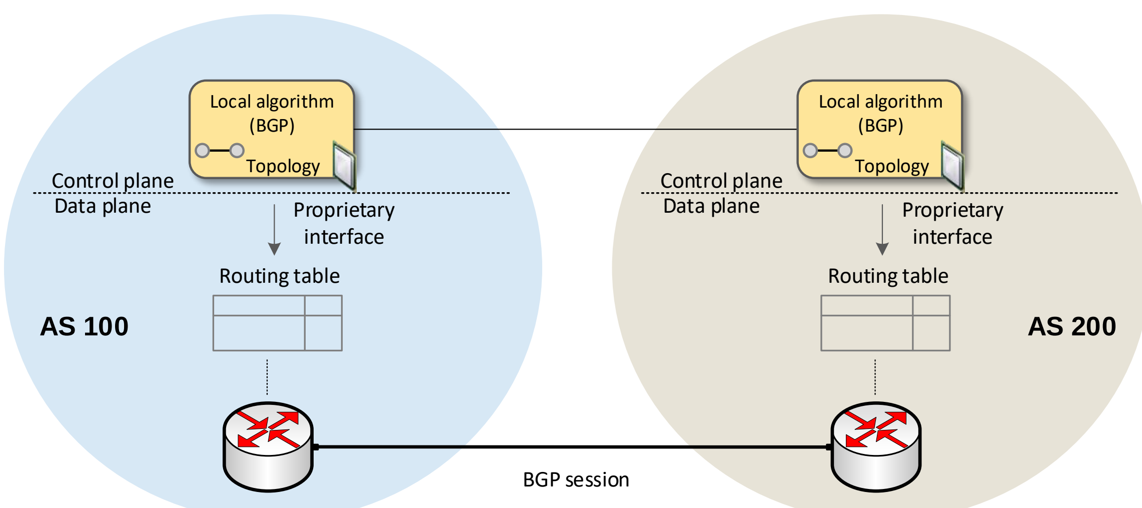

Figure 1. Legacy networks use BGP to share routing information between ASes.

1.2 SDN network

Routing protocols were essential to respond to rapidly changing network conditions. However, these conditions no longer exist in modern data centers. Thus, the behavior of the routing protocols wreaks temporary havoc inside the data center and hinders the ability to process large data traffic[1].

SDN is a new paradigm that solves the aforementioned problem by creating a centralized approach, rather than a distributed one. The main concept of SDN is to separate the control plane from the data plane in order to maximize the efficiency of the data plane devices. Moving the control software off the device into a centralized server makes it capable of seeing the entire network and making decisions that are optimal given a complete understanding of the situation[1].

Consider Figure 2. The control plane is decoupled from the data plane. The former is moved into a centrally located computer resource and it controls data plane devices by pushing rules into their tables[1].

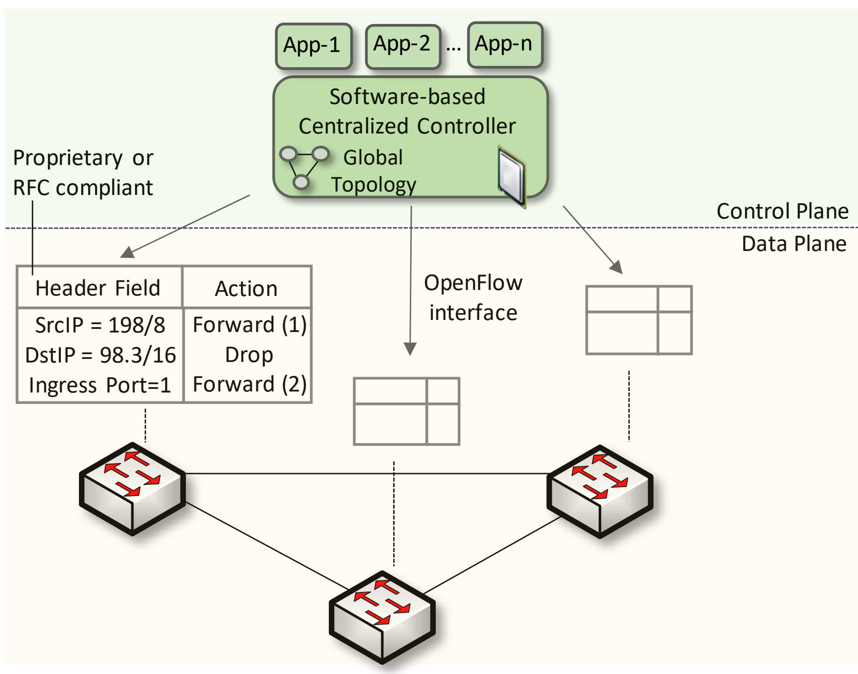

Figure 2. The control plane is embedded in a centralized server and it is decoupled from data plane devices.

1.3 Interworking SDN and legacy networks via SDN-IP application

SDN networks operate in a different manner than legacy networks, which are not going to be entirely replaced in the present time. Thus, one of the obstacles of deploying an SDN network was to integrate it with IP networks[1].

Peering between ASes on the Internet today is universally done with BGP. Therefore, a clear mechanism is needed for an SDN AS to communicate with IP ASes via BGP[1].

SDN-IP is an ONOS application that allows the SDN network to scale and connect to the rest of the Internet[3].

Consider Figure 3. Typically, an SDN network is composed of various OpenFlow switches (switches s1 and s2) connected to the controller (c0). External networks connect through their BGP routers (router r1) to the SDN data plane (switch s1), i.e., the OpenFlow switches, via an EBGP session. To communicate with external IP networks, a BGP speaker (router r2), referred to as IBGP router, must exist within the SDN network and be connected to the data plane. The SDN-IP application runs on top of the ONOS controller and it is connected to the IBGP speaker within the SDN network through an IBGP session[4]. The process that takes place is summarized as follows[4].

1. The network operator expresses the attachment points where both the BGP speakers (router r2) and external routers (router r1) are attached, as well as which BGP speaker is in charge of peering with a certain router. This operation is done using a configuration file that is loaded into ONOS prior to activating the SDN-IP application.
2. Once activated, SDN-IP application parses the configurations and puts in communication the external routers with the BGP speakers. This is translated by ONOS into flows and inserted to the switches. Additionally, the controller opens the port 2000 to communicate with the BGP speakers.
3. Assuming the BGP speakers and the external routers have been configured to peer via BGP, once the flows are installed in the switches, each external router can create EBGP sessions with the BGP speakers.
4. Routes get advertised from the external routers to the BGP speaker via EBGP.
5. Routes propagate among the BGP speakers within the SDN network, as well as to the SDN-IP application via IBGP.
6. The learned routes are advertised by the BGP speakers to other external routers.
7. SDN-IP application translates the learned routes into requests understood by ONOS. The latter translates the requests into OpenFlow entries on the switches.
8. External routers within legacy networks communicate directly through the OpenFlow data plane.

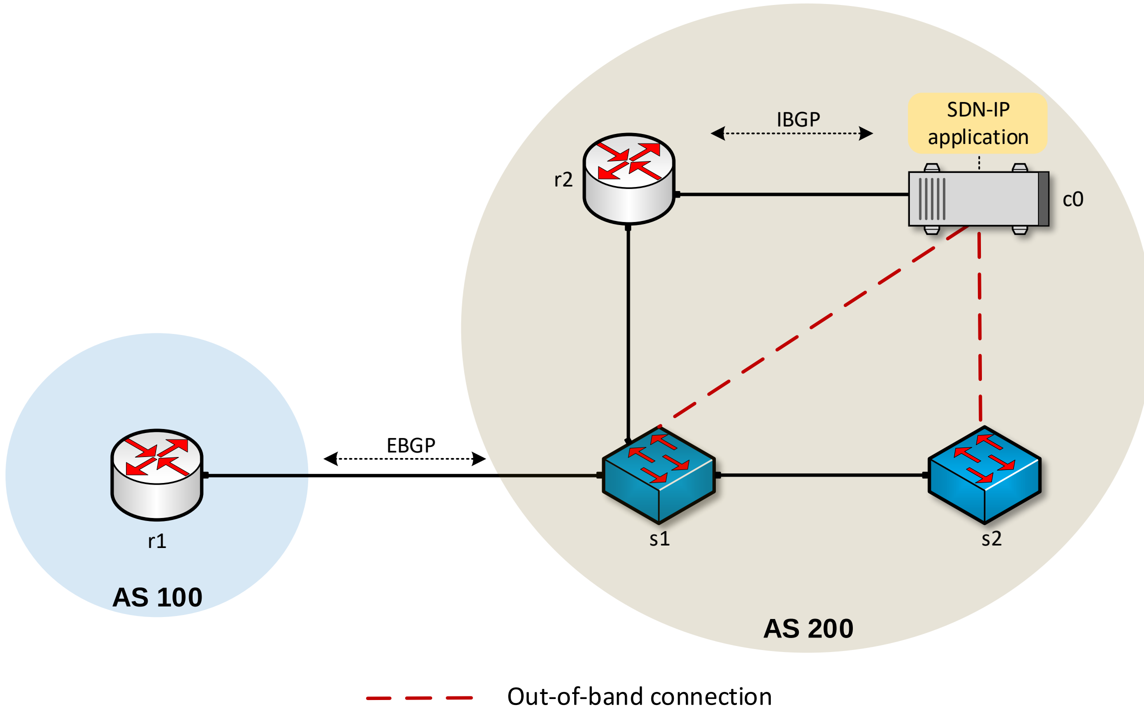

Figure 3. Integrating SDN and IP networks.

2 Lab topology

Consider Figure 4. The topology consists of two IP networks (AS 200 and AS 300) and one SDN network (AS 100). IP networks connect to the SDN network through their BGP routers. Router r1 is a BGP router within the SDN network. It communicates with EBGP routers r2 and r3 via the network addresses 192.168.12.0/30 and 192.168.13.0/30, respectively. Furthermore, router r1 is connected to the controller in order to propagate the BGP advertisements to the SDN-IP application running on top of ONOS controller. Router r1 and the controller are connected via the network 10.0.0.0/24.

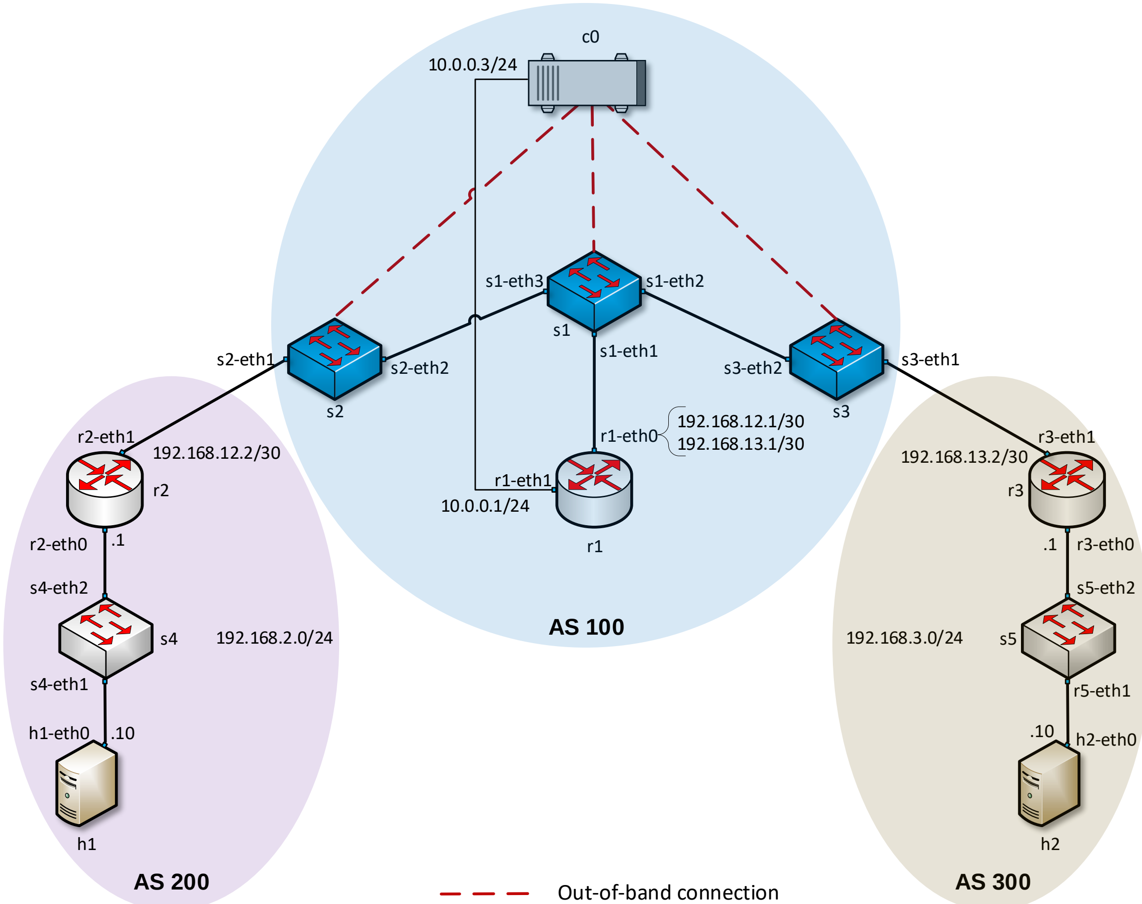

Figure 4. Lab topology.

2.1 Lab settings

The devices are already configured according to Table 2.

Table 2. Topology information.

Device	Interface	IP Address	Subnet	Default gateway
Router r1	r1-eth0	192.168.12.1	/30	N/A
		192.168.13.1	/30	N/A
	r1-eth1	10.0.0.1	/24	N/A
Router r2	r2-eth0	192.168.2.1	/24	N/A
	r2-eth1	192.168.12.2	/30	N/A
Router r3	r3-eth0	192.168.3.1	/24	N/A
	r3-eth1	192.168.13.2	/30	N/A
Host h1	h1-eth0	192.168.2.10	/24	192.168.2.1
Host h2	h2-eth0	192.168.3.10	/24	192.168.3.1

c0	n/a	127.0.0.1	/32	n/a
	n/a	10.0.0.3	/24	n/a

2.2 Loading a topology

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

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

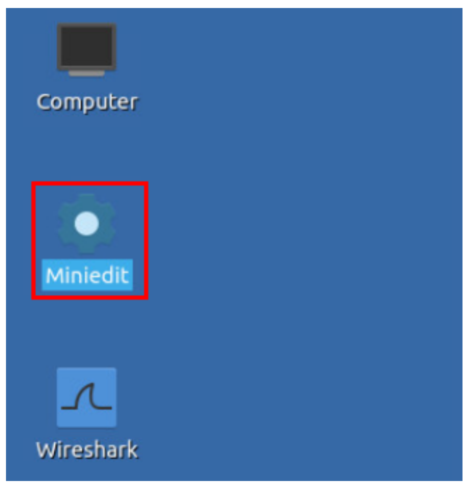

Figure 5. MiniEdit shortcut.

Step 2. On Miniedit's menu bar, click on *File* then *open* to load the lab's topology. Open the *Lab7.mn* topology file stored in the default directory, */home/sdn/SDN_Labs /lab7* and click on *Open*.

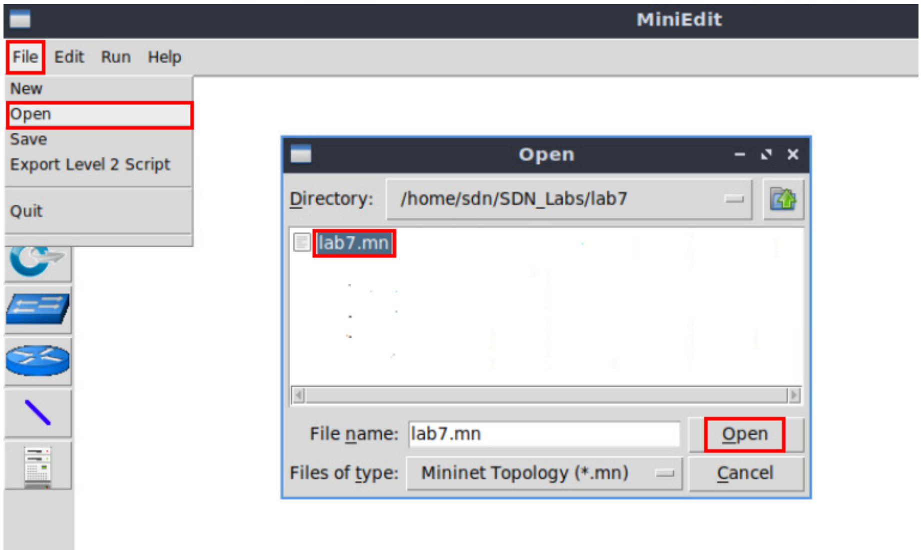

Figure 6. Opening topology.

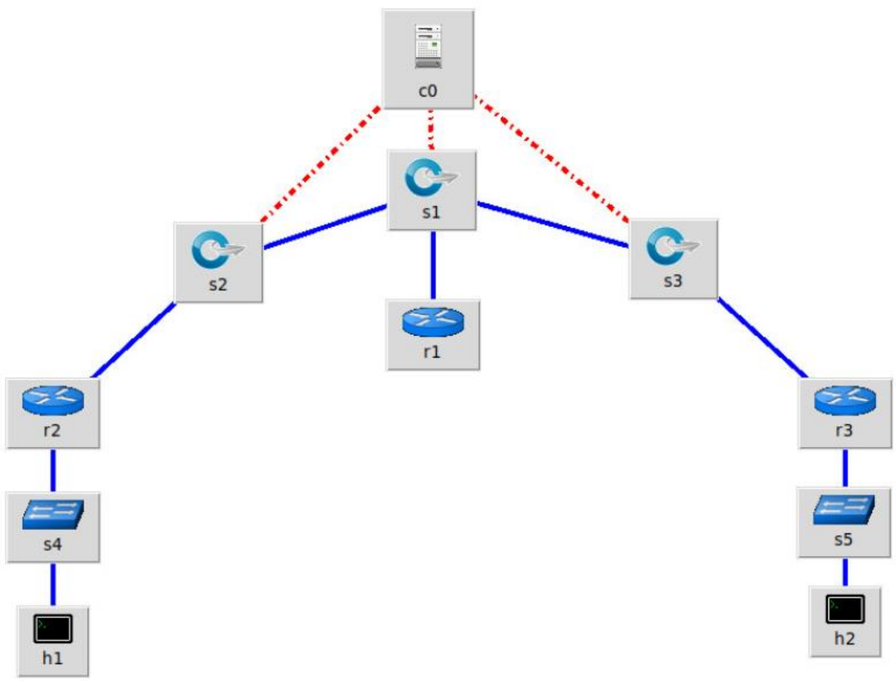

Figure 7. MiniEdit's topology.

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 terminal.

Figure 8. Opening Linux terminal.

Step 2. Click on the Linux terminal and navigate into *SDN_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 SDN_Labs/lab7
```

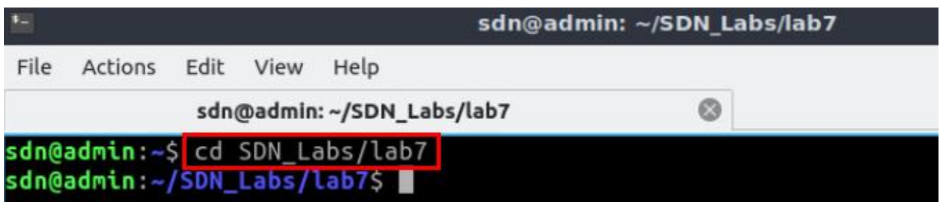

Figure 9. Entering the *SDN_Labs/lab7* directory.

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 lab7_conf.zip
```

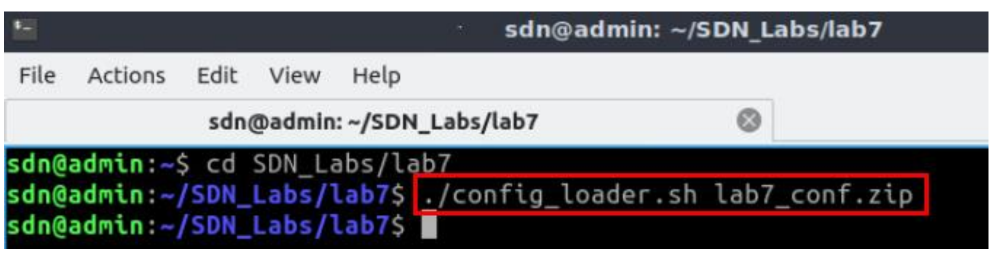

Figure 10. Executing the shell script to load the configuration.

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

```
exit
```

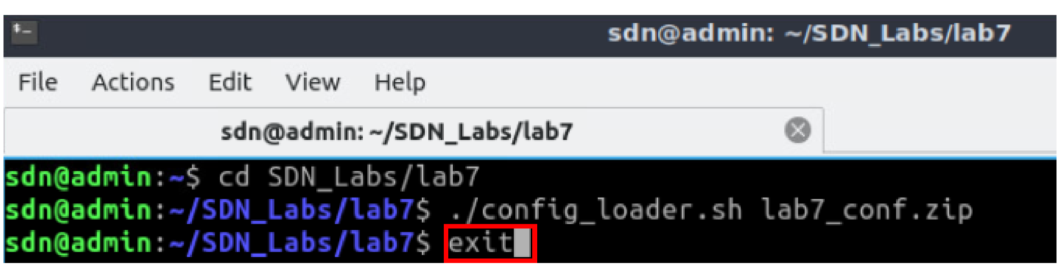

Figure 11. Exiting from the terminal.

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 12. Starting the emulation.

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

```
links
```

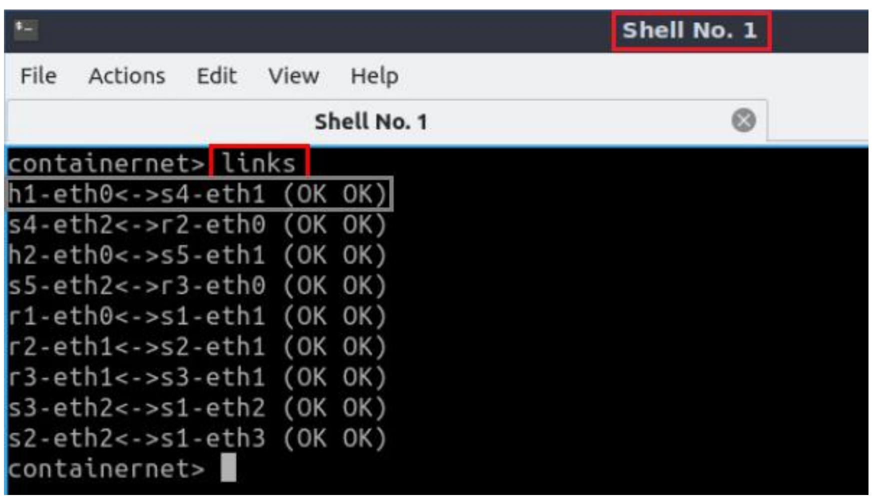

Figure 13. Displaying network interfaces.

In Figure 11, the link displayed within the gray box indicates that interface *eth1* of switch s4 connects to interface *eth0* of host h1 (i.e., *s4-eth1<->h1-eth0*).

2.5 Verify the 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 on that host.

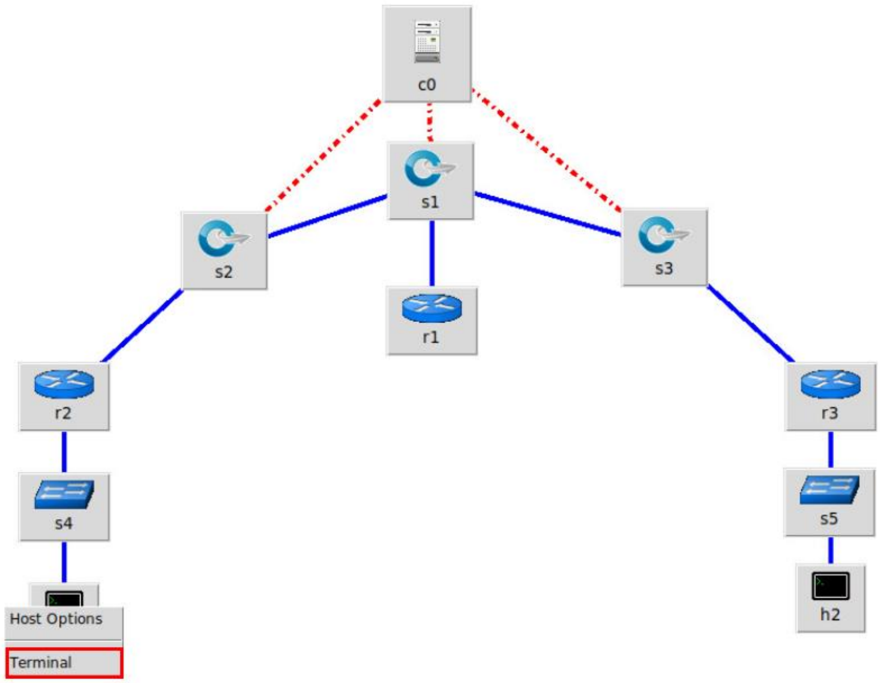

Figure 14. Opening a terminal on host h1.

Step 2. On host h1 terminal, type the command shown below to verify that the IP address was assigned successfully. You will corroborate that host h1 has two interfaces, *h1-eth0* configured with the IP address 192.168.2.10 and the subnet mask 255.255.255.0.

```
ifconfig
```

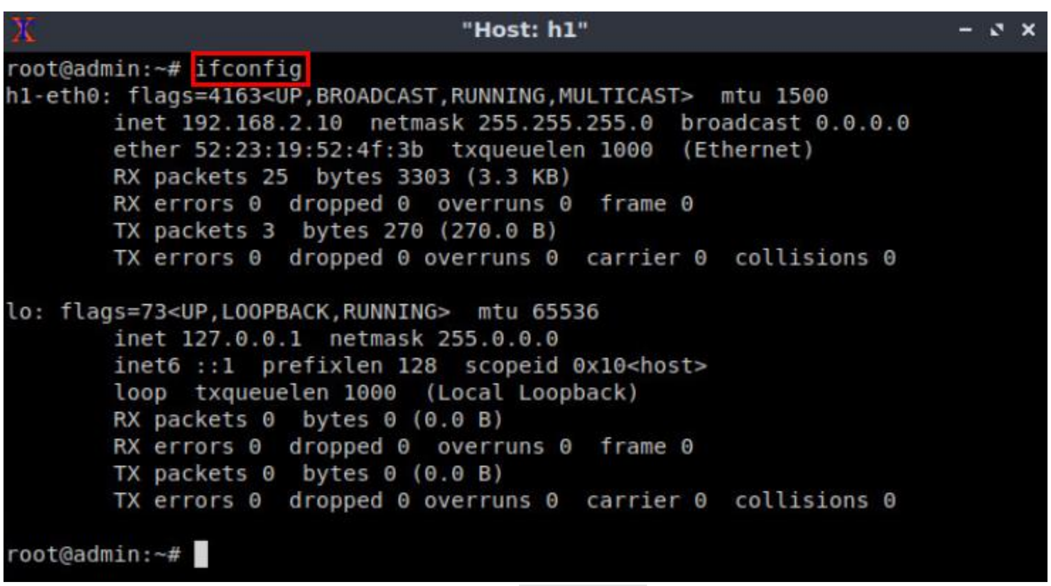

Figure 15. Output of `ifconfig` command.

Step 3. On host h1 terminal, type the command shown below to verify that the default gateway IP address is 192.168.2.1.

```
route
```

Figure 16. Output of `route` command.

Step 4. In order to verify host h2 default route, proceed similarly by repeating from step 1 to step 3 on 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*.

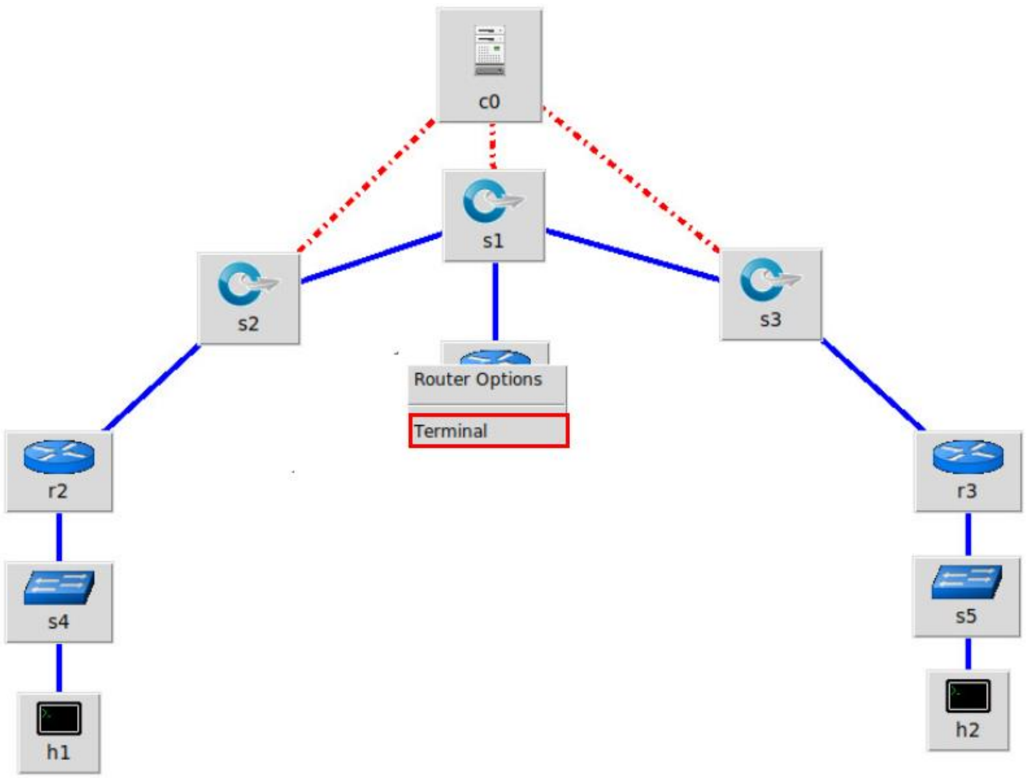

Figure 17. Opening a terminal on router r1.

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
```

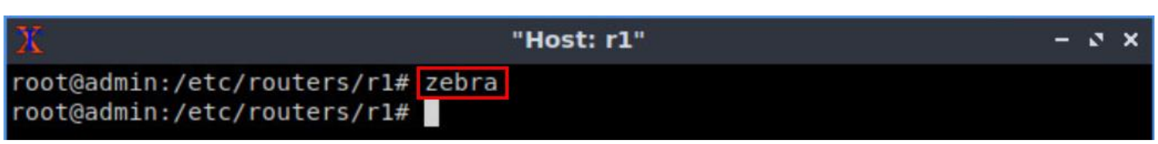

Figure 18. Starting zebra daemon.

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
```

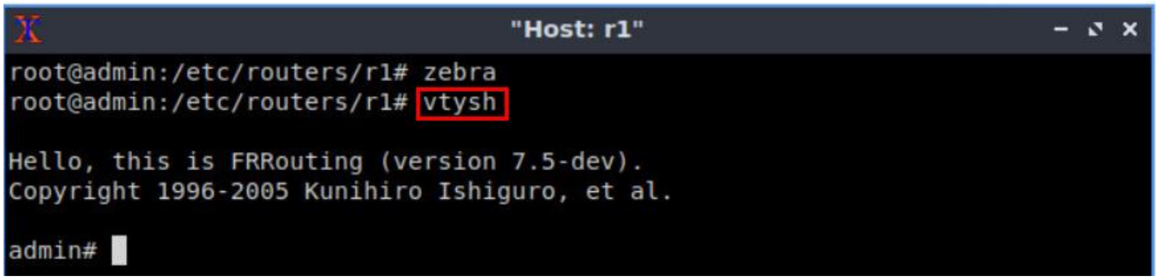

Figure 19. Starting vtysh on router r1.

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 of router 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
```

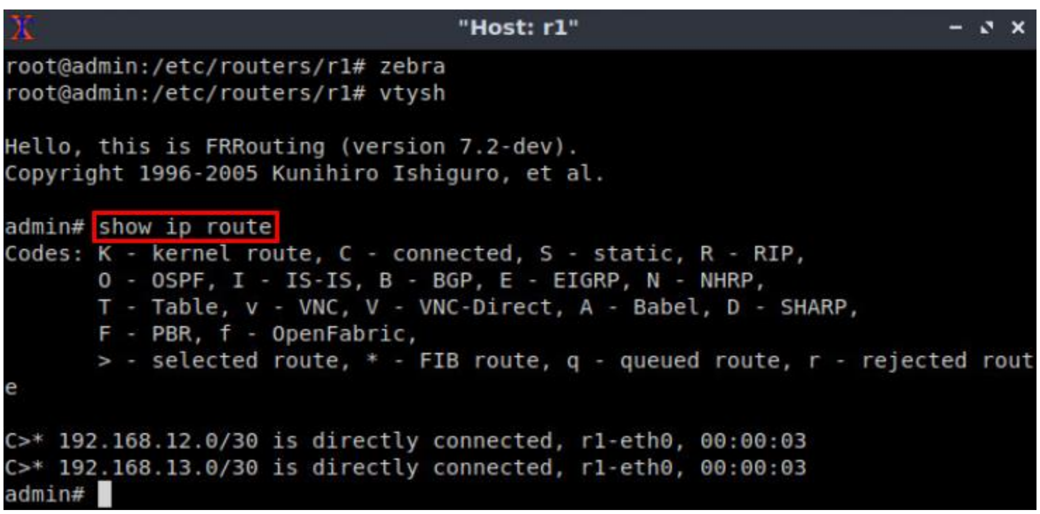

Figure 20.Displaying routing table of router r1.

The output in the figure above shows that the networks 192.168.12.0/24 and 192.168.13.0/30 are directly connected through the interface *r1-eth0*.

Step 9. Hold right-click on router r2 and select *Terminal*.

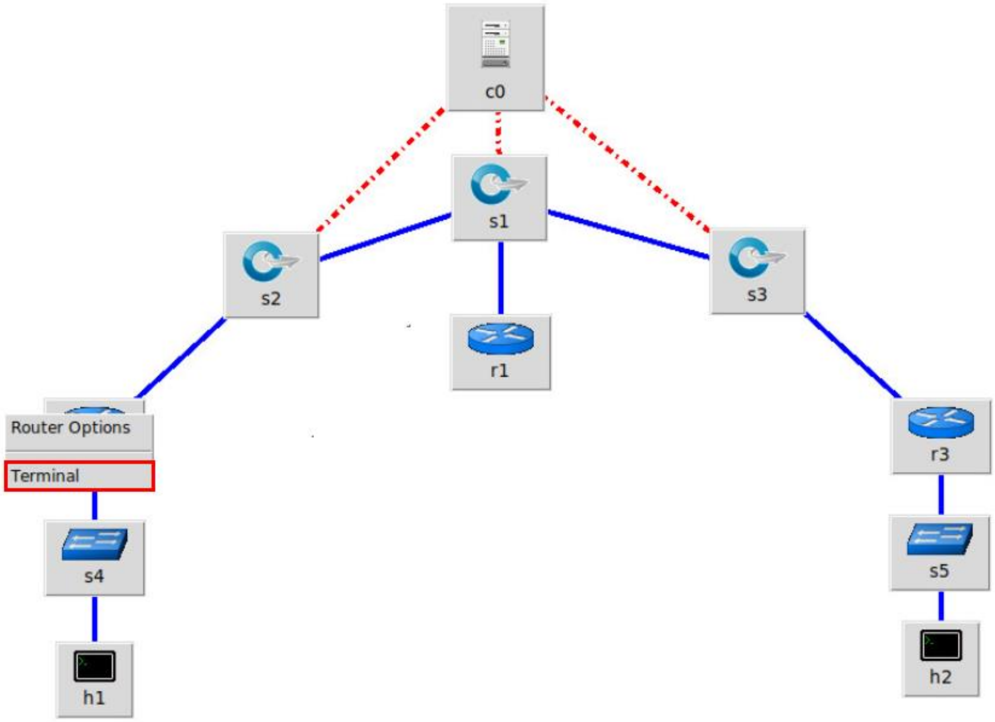

Figure 21. Opening a terminal on router r2.

Step 10. 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.

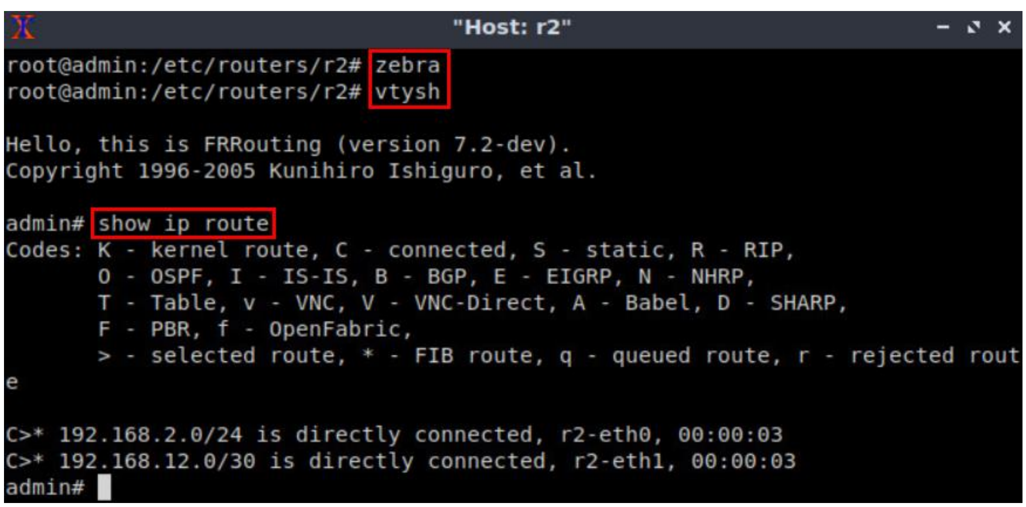

Figure 22. Displaying routing table of router r2.

Step 11. 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.

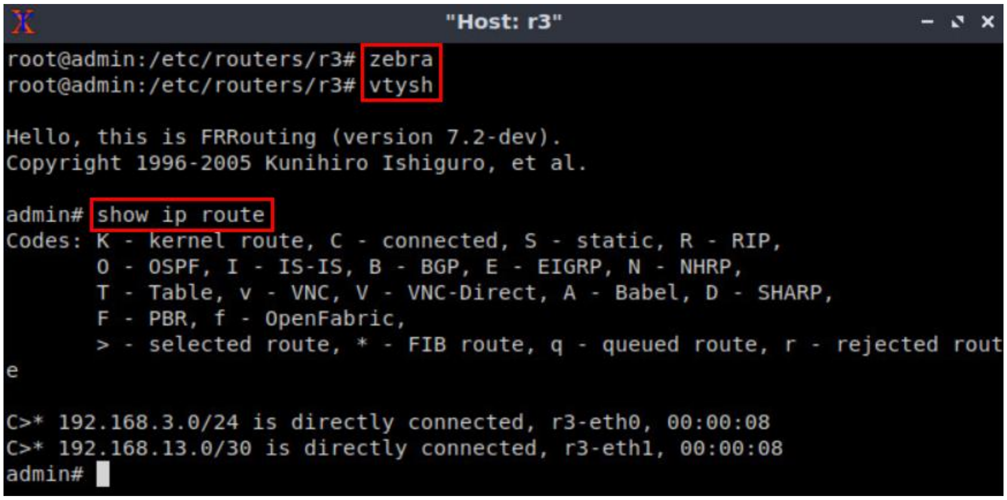

Figure 23. Displaying routing table of router r3.

3 Configuring BGP within legacy networks

In the previous section you used a script to assign the IP addresses to all devices' interfaces. In this section you will configure BGP routing protocol on the legacy networks (routers r2 and r3), the standard protocol used to connect ASes. First, you will initialize the daemon that enables BGP configuration. Then, you need to assign BGP neighbors to allow BGP peering to the remote neighbor. Additionally, you will advertise the local networks so that they are advertised to EBGP neighbors.

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

```
exit
```

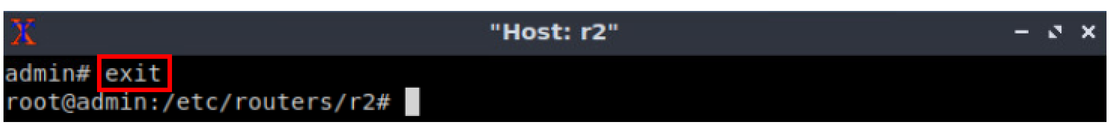

Figure 24. Exiting the vtysh session.

Step 2. Type the following command on router r2 terminal to start BGP routing protocol.

```
bgpd
```

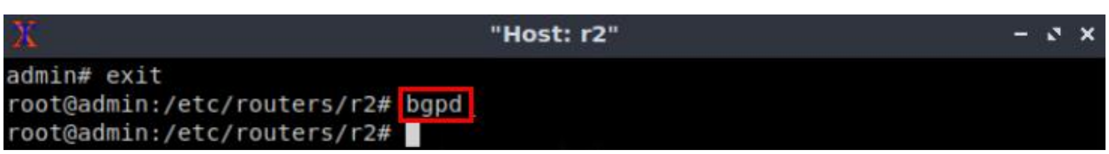

Figure 25. Starting BGP daemon.

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

```
vtysh
```

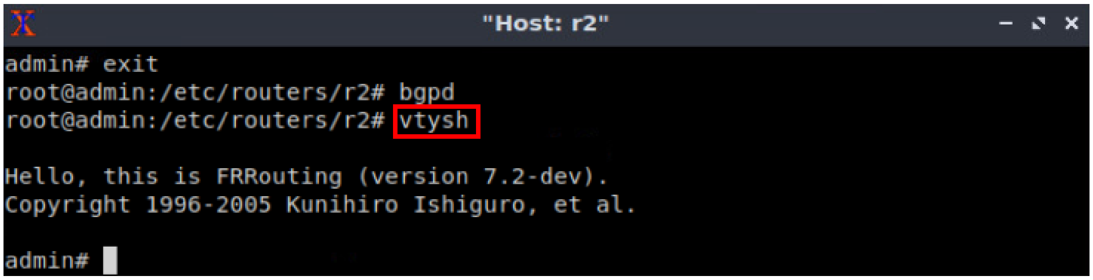

Figure 26. Starting vtysh on router r2.

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

```
configure terminal
```

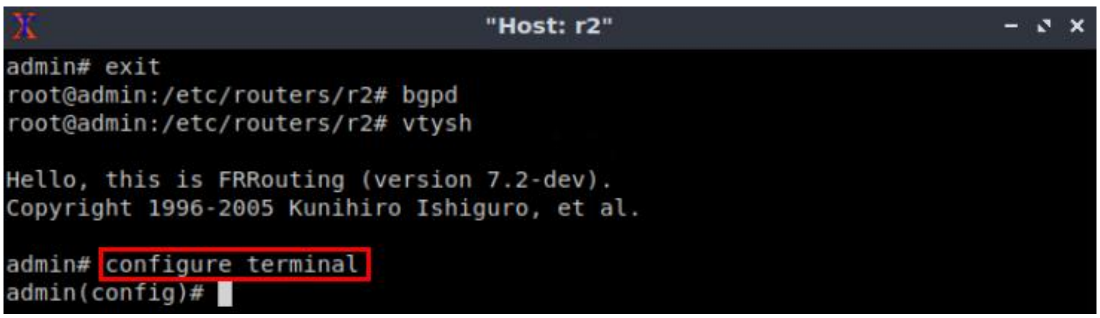

Figure 27. Enabling configuration mode on router r1.

Step 5. The Autonomous System Number (ASN) assigned for router r2 is 200. In order to configure BGP, type the following command.

```
router bgp 200
```

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 (192.168.12.1) and ASN of the remote BGP peer (AS 100).

```
neighbor 192.168.12.1 remote-as 100
```

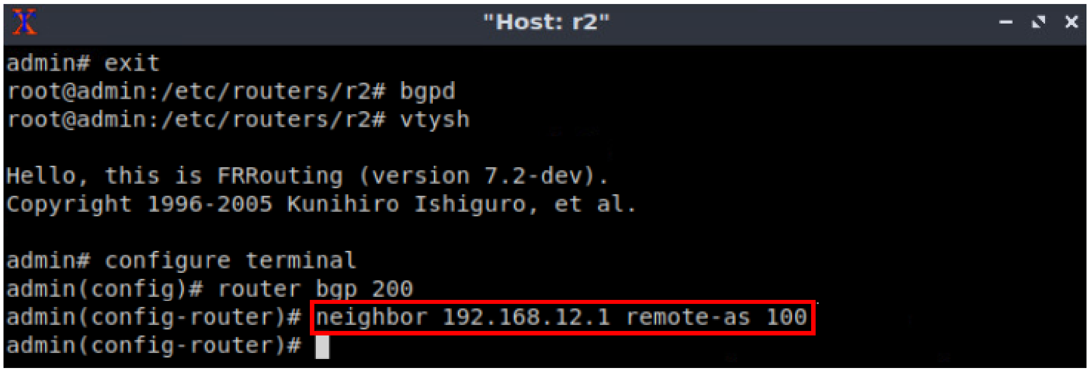

Figure 29. Assigning BGP neighbor to router r1.

Step 7. Issue the following command so that router r2 advertises the network 192.168.2.0/24.

```
network 192.168.2.0/24
```

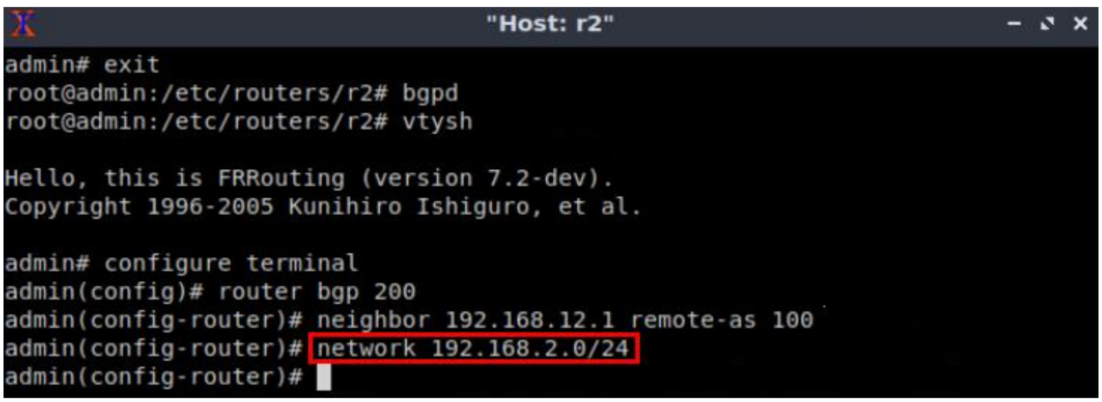

Figure 30. Advertising the network connected to router r2.

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

```
end
```

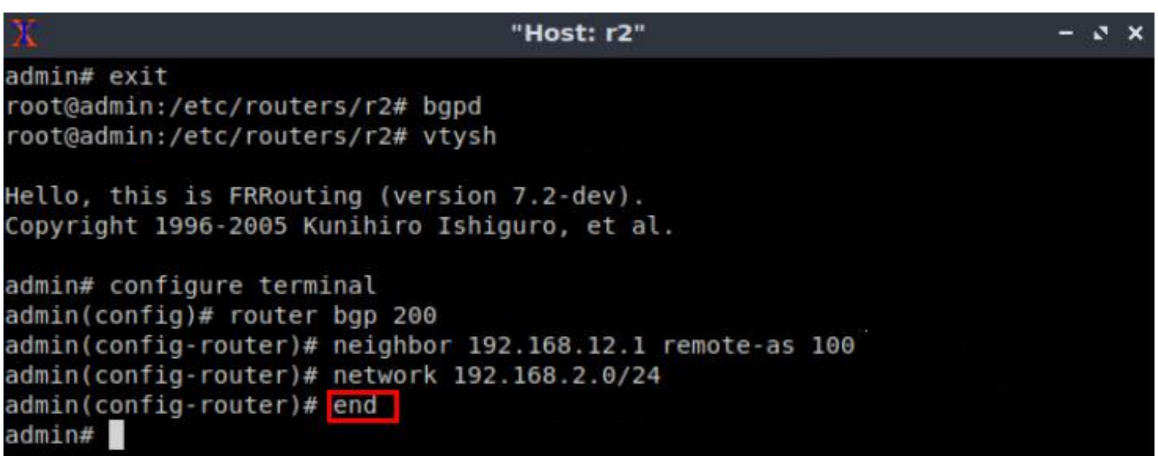

Figure 31. Exiting from configuration mode.

Step 9. 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
```

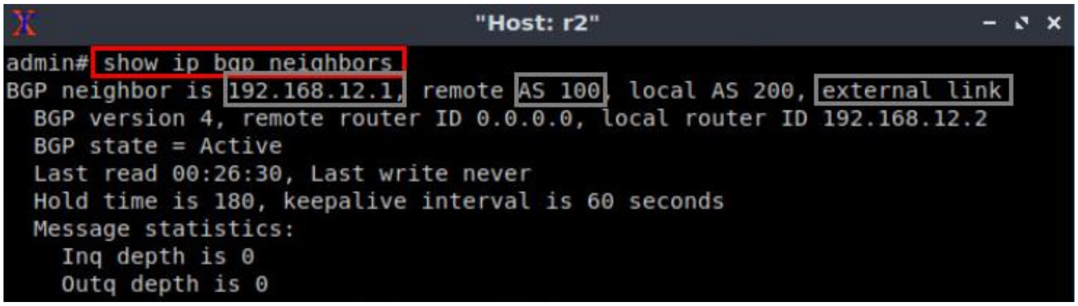

Figure 32. Verifying BGP neighbors on router r2.

Step 10. The configuration of BGP on router r3 is similarly configured as router r2. Router r3 lies within AS 300, it establishes BGP neighbor relationship with router r1 (192.168.13.1) in AS 100, and advertises the network 192.168.3.0/24. The configuration of BGP on router r3 is depicted in the below figure.

```
end
```

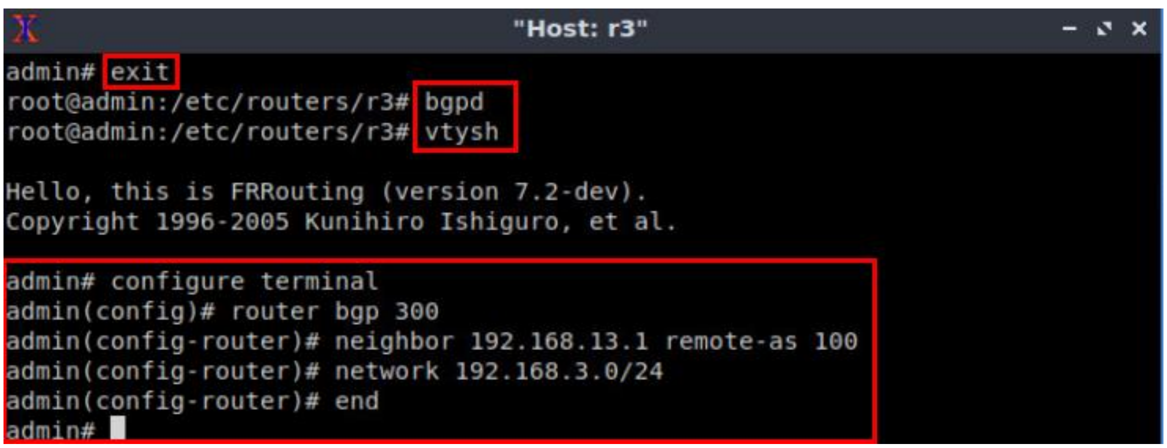

Figure 33. BGP configuration on router r3.

Step 11. To verify BGP neighbors of router r3, type the following command.

```
show ip bgp neighbors
```

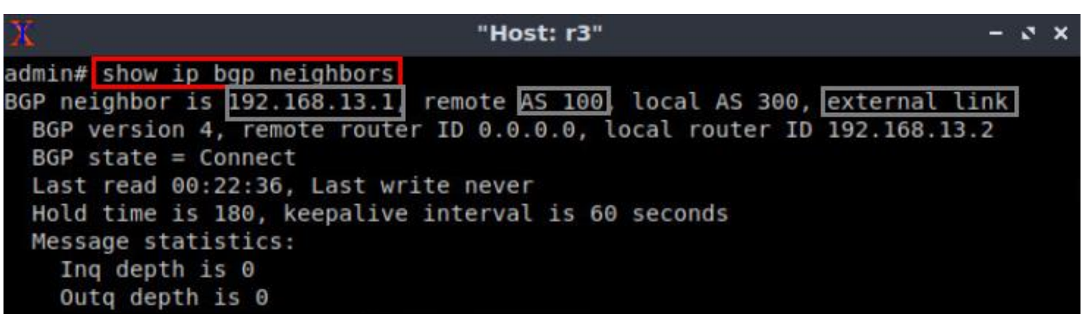

Figure 34. Verifying BGP neighbors on router r3.

4 Starting ONOS controller

In the section, you will start ONOS controller and activate OpenFlow application so that the controller discovers the devices, hosts, and links in the topology.

Step 1. Go to the opened linux terminal.

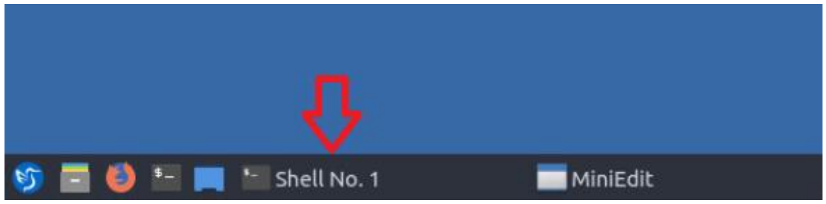

Figure 35. Opening Linux terminal.

Step 2. Click on *File>New Tab* to open an additional tab in Linux terminal. Alternatively, the user may press `Ctrl+Shift+T`.

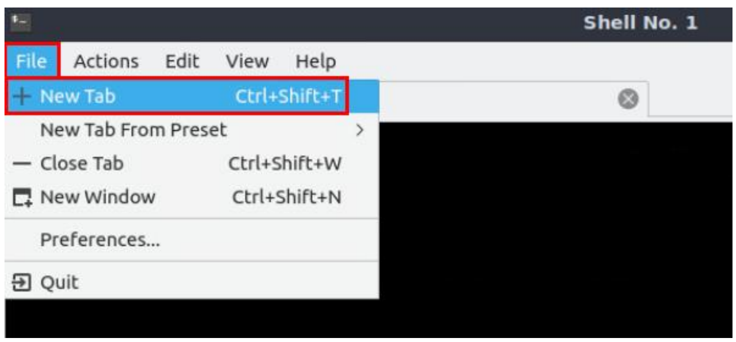

Figure 36. Opening an additional tab.

Step 3. Navigate into *SDN_Labs/lab7* directory by issuing the following command.

```
cd SDN_Labs/lab7
```

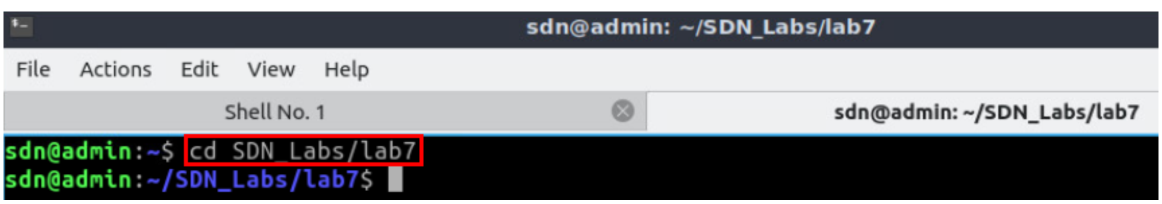

Figure 37. Entering the *SDN_Labs/lab7* directory.

Step 4. Issue the command below to execute programs with the security privileges of the superuser (root). When prompted for a password, type `password`.

```
sudo su
```

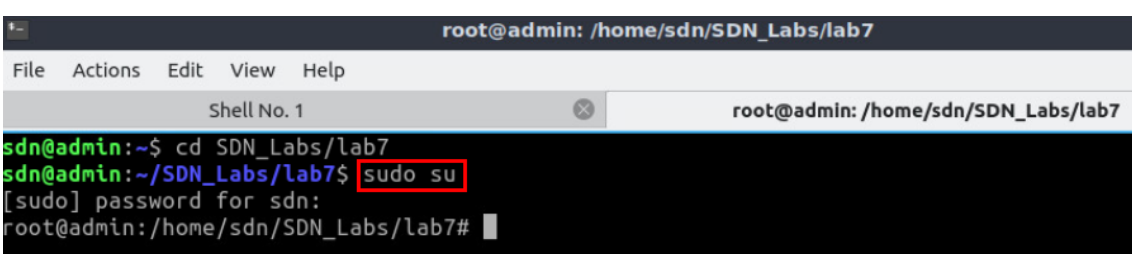

Figure 38. Switching to root mode.

Step 5. A script was written to run ONOS and enter its Command Line Interface (CLI). In order to run the script, issue the following command.

```
./run_onos
```

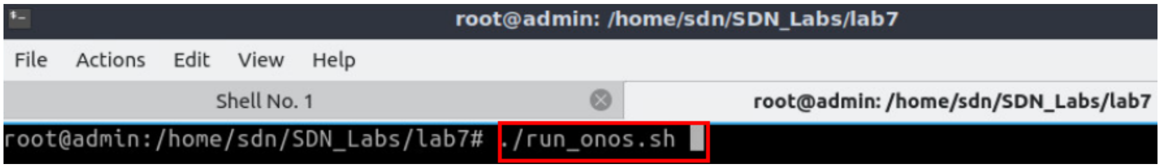

Figure 39. Running ONOS instance.

Once the script finishes executing and ONOS is ready, you will be able to execute commands on ONOS CLI as shown in the figure below. Note that this script may take a couple of minutes.

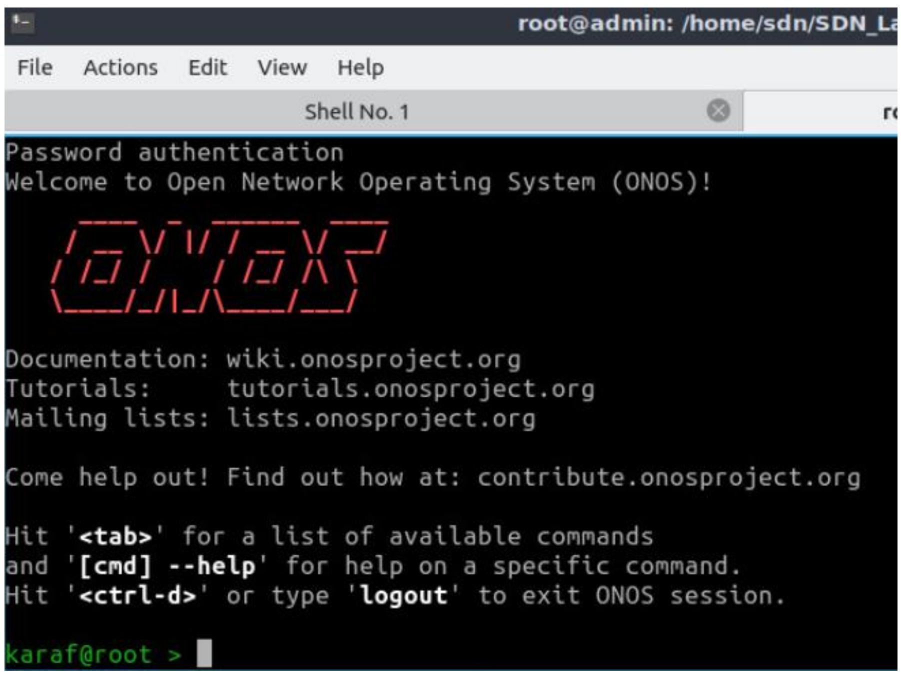

Figure 40. ONOS CLI.

Step 6. In ONOS terminal, issue the following command to activate the OpenFlow application.

```
app activate org.onosproject.openflow
```

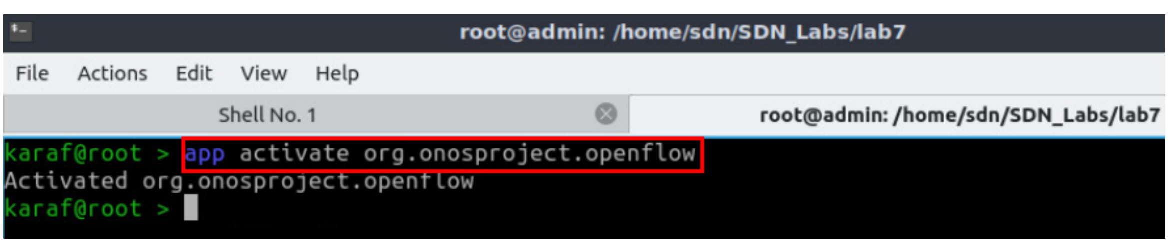

Figure 41. Activating OpenFlow application.

Note that when you activate any ONOS application, you may have to wait few seconds so that the application gives the correct output.

Step 7. To display the list of all currently known devices (OVS switches), type the following command.

```
devices
```

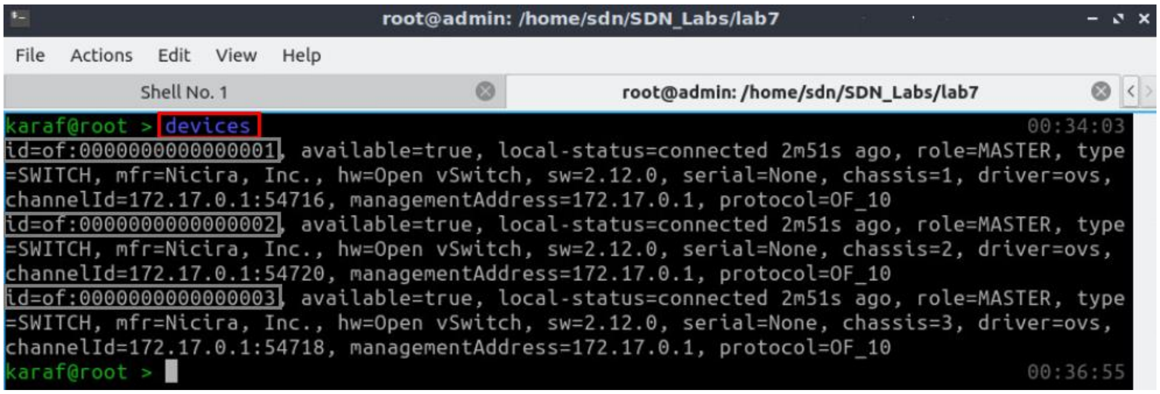

Figure 42. Displaying the current known devices (switches).

Step 8. To display the list of all currently known links, type the following command.

```
links
```

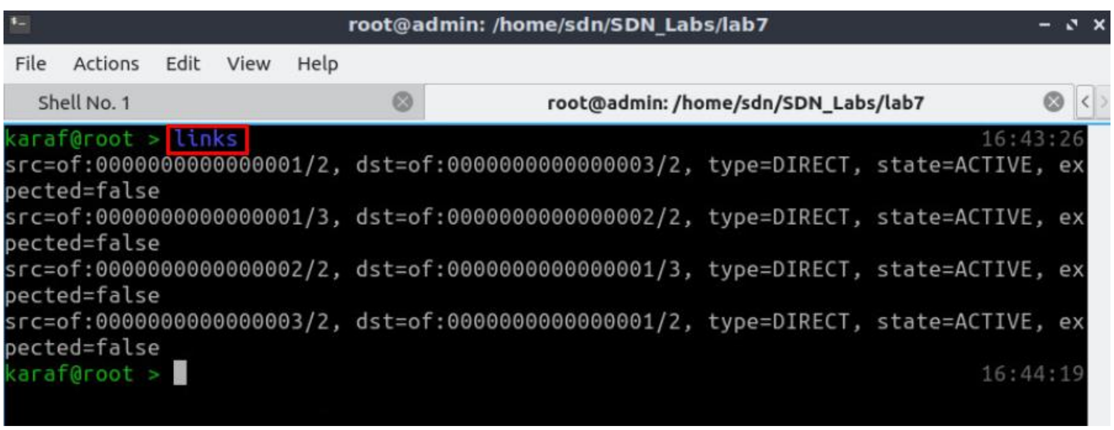

Figure 43. Displaying the current known links.

Step 9. To display the list of all currently known hosts, type the following command.

```
hosts
```

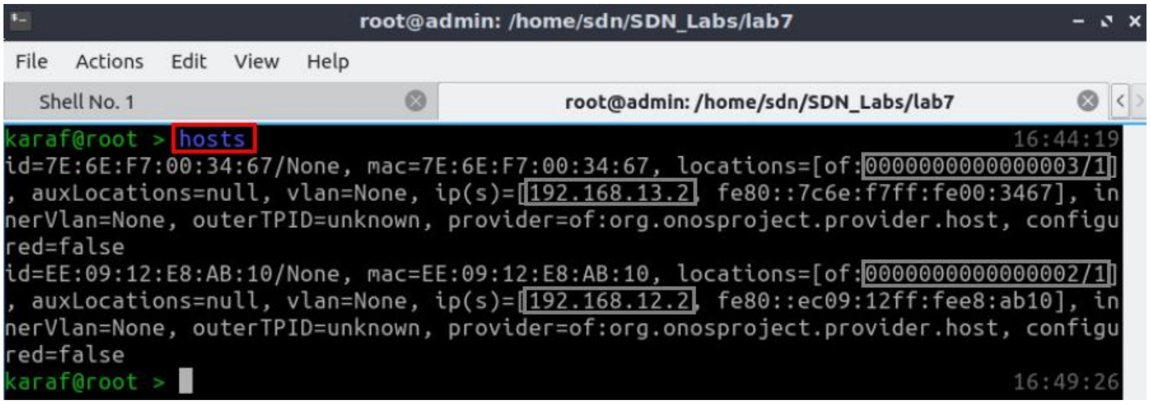

Figure 44. Displaying the current known links.

Consider Figure 44. ONOS recognizes router r2 (192.168.12.2) and router r3 (192.168.13.2) and display the interfaces of the OpenFlow switches they are connected to. Note that you might have to wait until ONOS discovers the two hosts in case they don't appear immediately.

5 Integrating SDN and legacy networks

In the previous sections, you configured the legacy devices, as well as started ONOS and its OpenFlow application to discover the topology. In this section, you will first execute a script that connects the IBGP speaker (router r1) with ONOS controller, thus, the two entities can communicate. Additionally, you will configure BGP on router r1 so that it peers with routers r2 and r3 in the external networks, as well as with ONOS. Furthermore, you will activate ONOS SDN-IP application to interconnect the three ASes.

5.1 Connecting the IBGP speaker (router r1) with ONOS controller

In this section, you will execute a script that creates a peer-to-peer link connecting router r1 with ONOS.

Step 1. Go to Mininet tab in the Linux terminal.

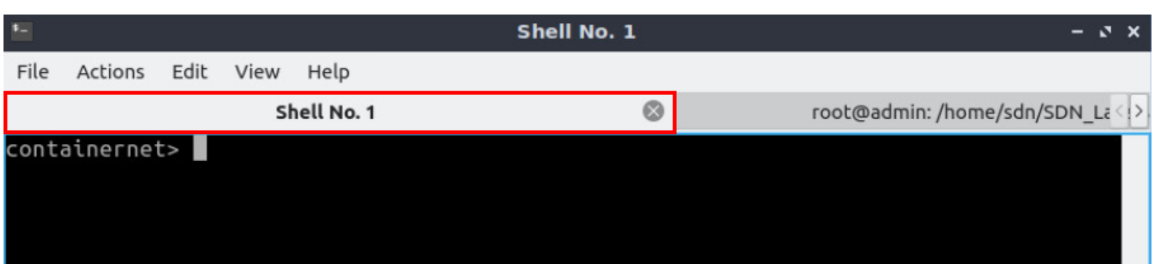

Figure 45. Opening Mininet tab.

Step 2. In order to create a point-to-point network between the IBGP speaker (router r1) and ONOS, a script was written to facilitate the process. In order to execute the script, type the following command.

```
source create_link.sh
```

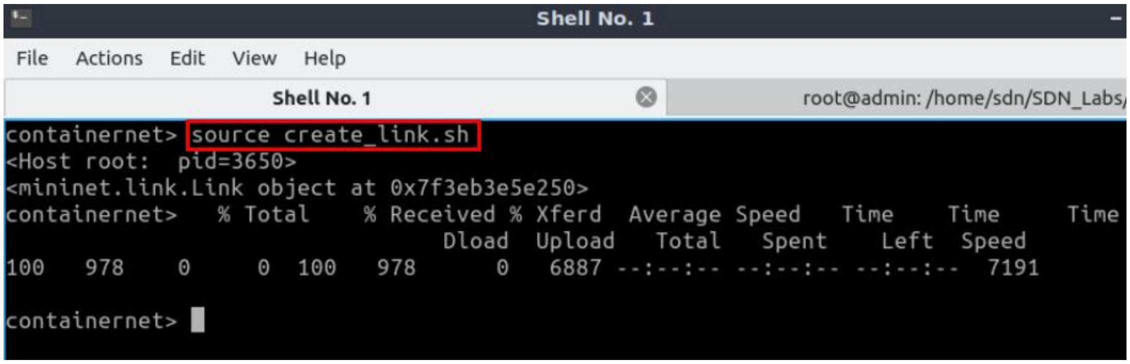

Figure 46. Creating a point-to-point network (link) between the IBGP speaker and ONOS controller.

Consider Figure 46. The script creates a point-to-point network between router r1 and ONOS. The network address of the point-to-point network is 10.0.0.0/24. Router r1 is assigned the IP address 10.0.0.1/24, whereas ONOS controller is assigned 10.0.0.3/24.

Step 3. In router r1 terminal, type the following command to exit the vtysh session.

```
exit
```

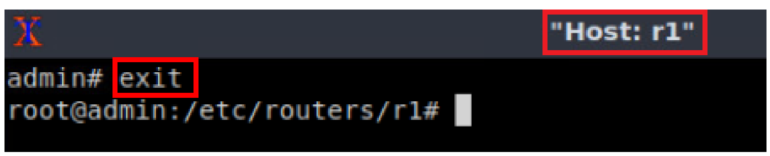

Figure 47. Creating a network between the IBGP speaker and ONOS controller.

Step 4. Now that router r1 is connected to ONOS controller, a new interface must appear in it. In order to verify the connected interface, type the following command.

```
ifconfig
```

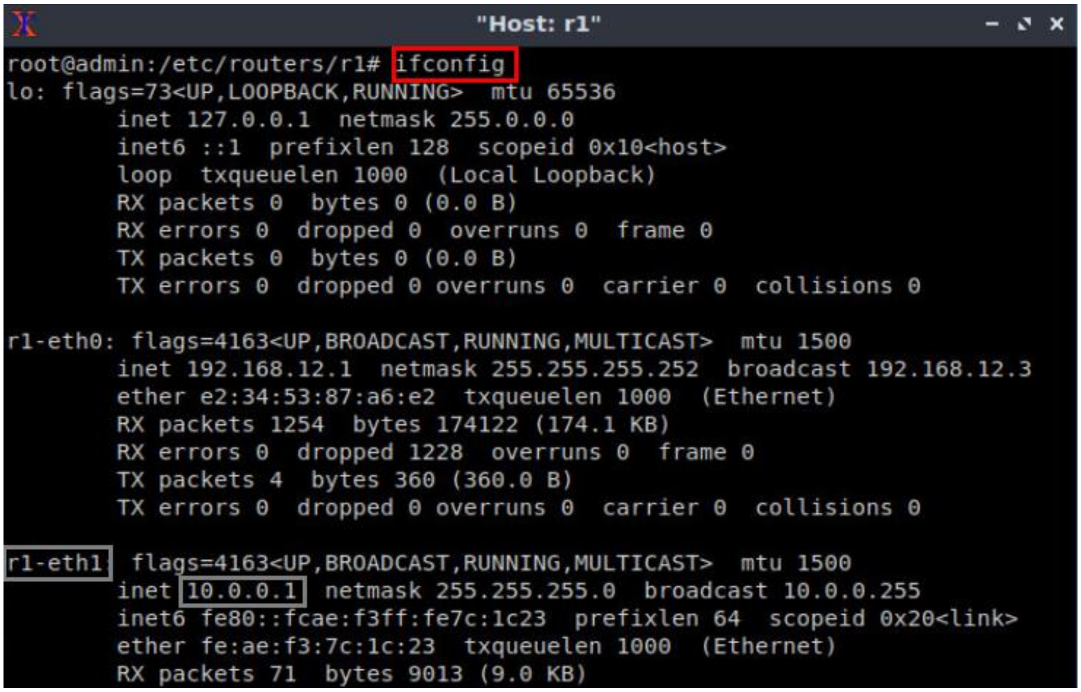

Figure 48. Listing router r1’s interfaces.

Consider Figure 48. Interface *r1-eth1* is added after creating a point-to-point network between router r1 and ONOS controller. Furthermore, the interface is associated with the IP address 10.0.0.1.

5.2 Configuring BGP on router r1

In this section, you will configure BGP on router r1 to peer with routers r2 and r3, as well as with ONOS.

Step 1. Type the following command on router r1 terminal to start BGP routing protocol.

```
bgpd
```

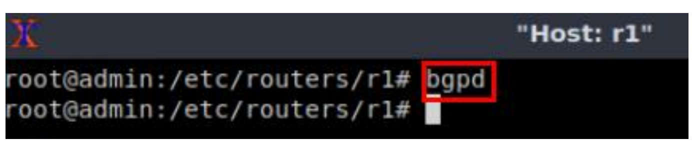

Figure 49. Starting BGP daemon.

Step 2. In order to enter to router r1 terminal, type the following command.

```
vtysh
```

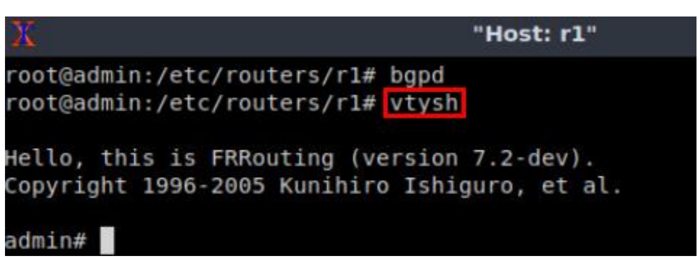

Figure 50. Starting vtysh on router r1.

Step 3. To enable router r1 configuration mode, issue the following command.

```
configure terminal
```

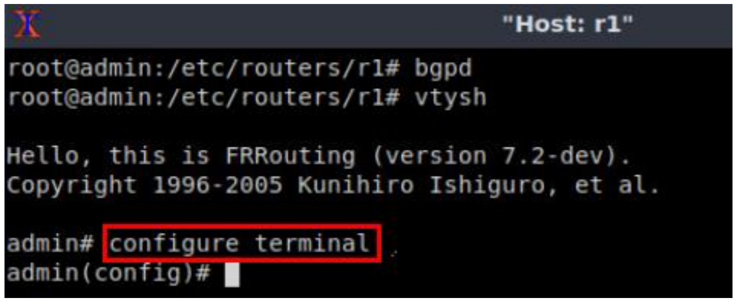

Figure 51. Enabling configuration mode on router r1.

Step 4. The ASN assigned for router r1 is 100. In order to configure BGP, type the following command.

```
router bgp 100
```

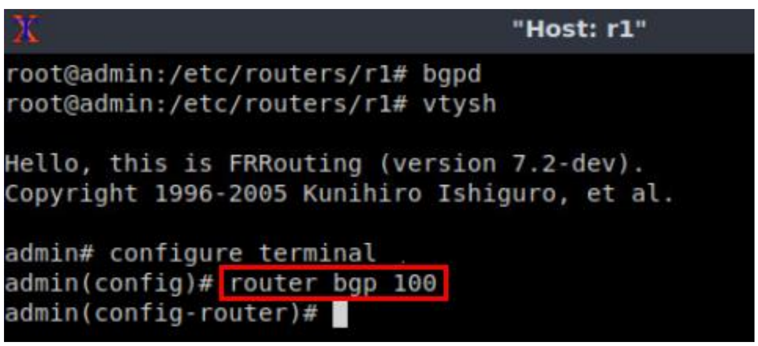

Figure 52. Configuring BGP on router r1.

Step 5. 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
```

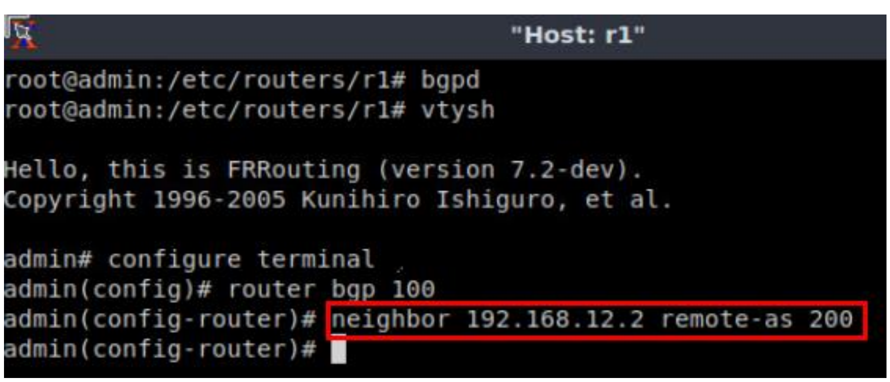

Figure 53. Assigning BGP neighbor to router r1.

Step 6. Similarly, add router r3 (192.168.13.2) in AS 300 as a BGP neighbor to router r1.

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

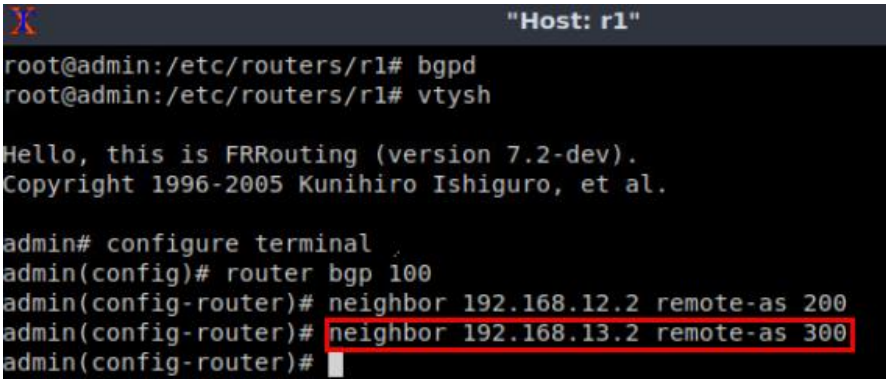

Figure 54. Assigning BGP neighbor to router r1.

Step 7. Router r1 and ONOS controller are connected using a point-to-point network (10.0.0.0/24). The IP address assigned to the controller is 10.0.0.3. As router r1 is the IBGP speaker within the SDN network, it must establish a BGP peering relationship with the controller in its network (AS 100). In order to establish BGP peering relationship with the controller, type the following command.

```
neighbor 10.0.0.3 remote-as 100
```

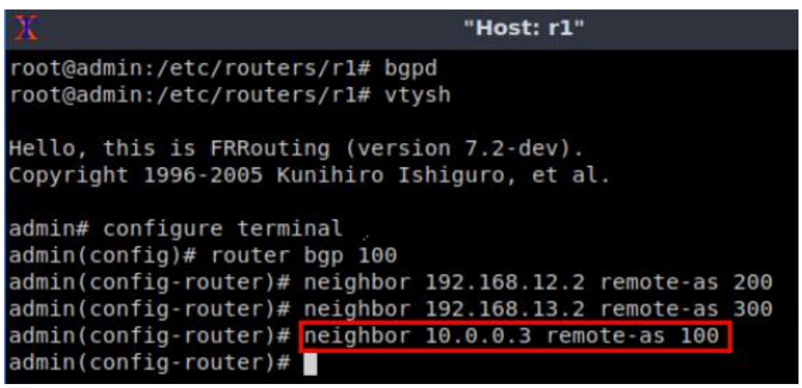

Figure 55. Assigning BGP neighbor to router r1.

Step 8. By default, ONOS listens on TCP port number 2000 for incoming BGP connections, which is not the default BGP port number 179. In order to specify the port for incoming BGP messages from ONOS, write the following command.

```
neighbor 192.168.12.2 port 2000
```

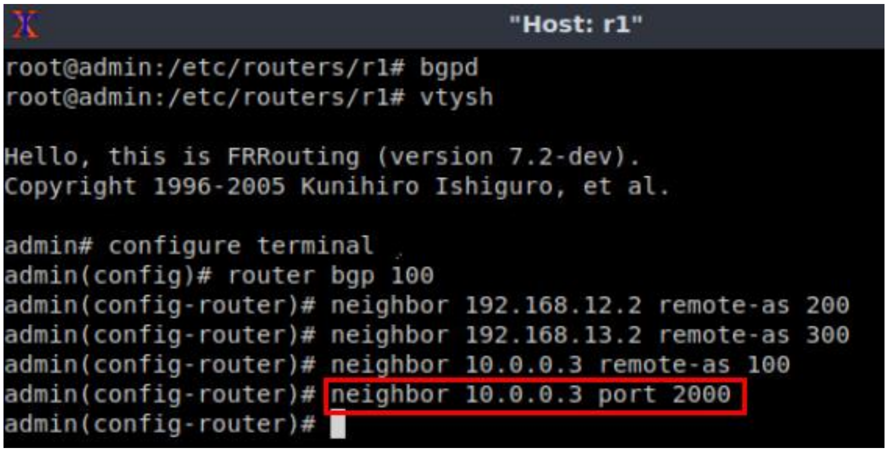

Figure 56. Changing the Listening port for BGP connections.

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

```
end
```

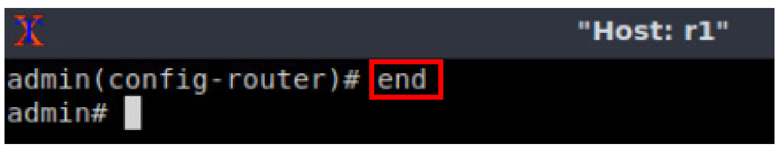

Figure 57. Exiting from configuration mode.

Step 10. 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) or router r3 (192.168.3.0/24) as there is no enabled ONOS application that deals with BGP routes.

```
end
```

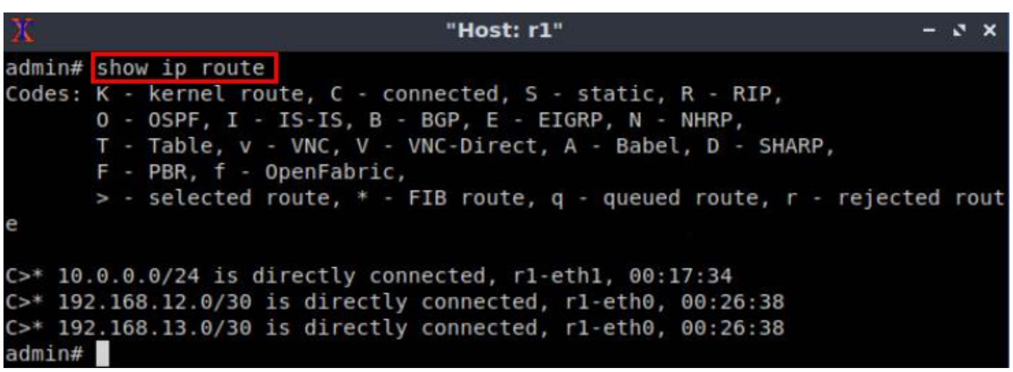

Figure 58. Displaying the routing table of router r1.

5.3 Activating SDN-IP application

In this section, you will activate the SDN-IP application and other dependencies (applications) that will interconnect the SDN network with the legacy network.

Step 1. Go to ONOS terminal.

Figure 59. Opening ONOS terminal.

Step 2. Before activating the SDN-IP application you must start the *org.onosproject.config* application. The latter is an application for the network configuration. In order to activate the config application, type the following command.

```
app activate org.onosproject.config
```

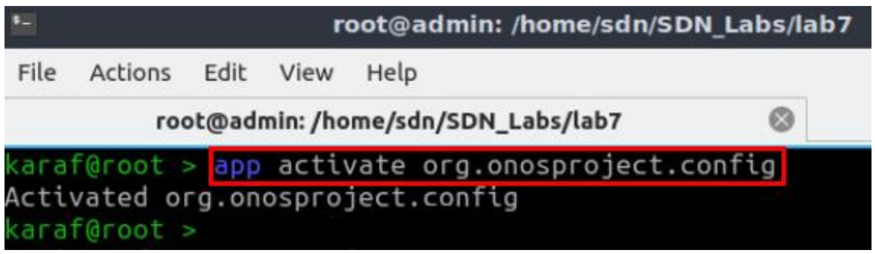

Figure 60. Activating ONOS config appication.

Step 3. SDN-IP application has an additional application dependency that it relies on to ensure Address Resolution Protocol (ARP) requests are resolved properly. This is the *org.onosproject.proxyarp* application that responds to ARP requests on behalf of hosts and external routers.

```
app activate org.onosproject.proxyarp
```

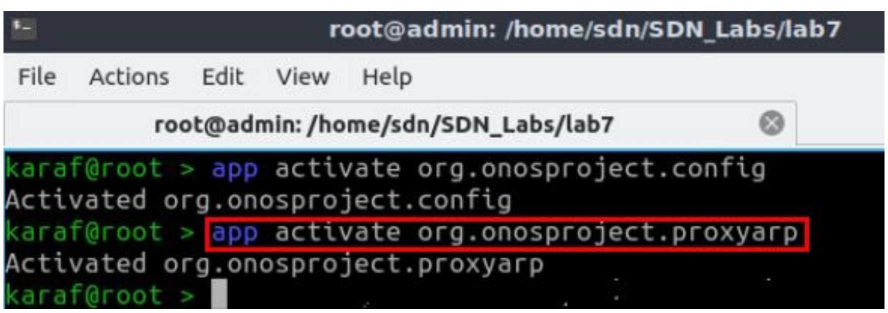

Figure 61. Activating ONOS proxyarp application.

Step 4. Once the dependencies are started, the SDN-IP application can be activated. In order to do that, type the following command.

```
app activate org.onosproject.sdnip
```

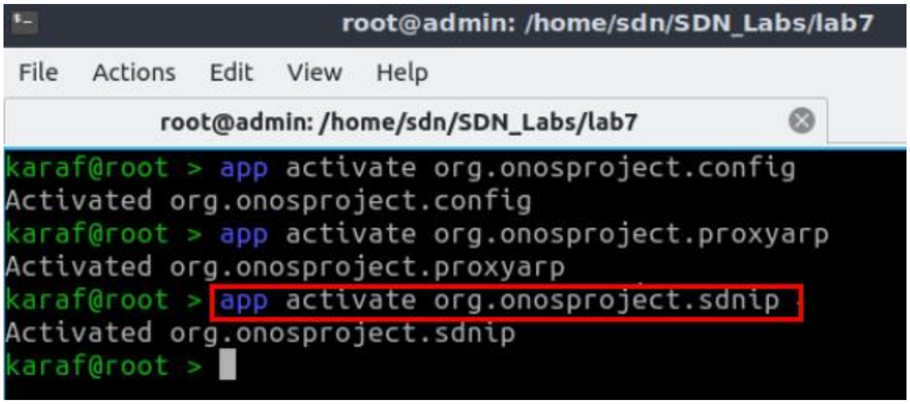

Figure 62. Activating ONOS SDN-IP application.

After activating the above applications, you might have to wait few minutes until the applications discover the topology, and exchange information in order to get correct results.

Step 5. In ONOS terminal, type the following command to show the IBGP neighbors that have connected to SDN-IP application.

```
bgp-neighbors
```

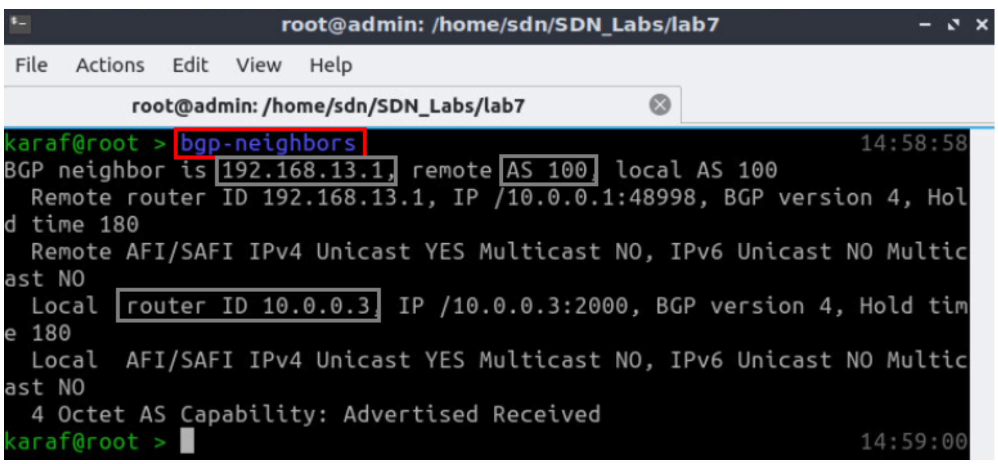

Figure 63. Viewing IBGP neighbors within the SDN network.

Consider Figure 63. The neighbor 192.168.13.1 corresponds to router r1 in AS 100. This is the internal BGP speaker in the SDN network. The local router ID that the SDN-IP application uses is 10.0.0.3.

Step 6. To show the routing table of SDN-IP, type the following command.

```
routes
```

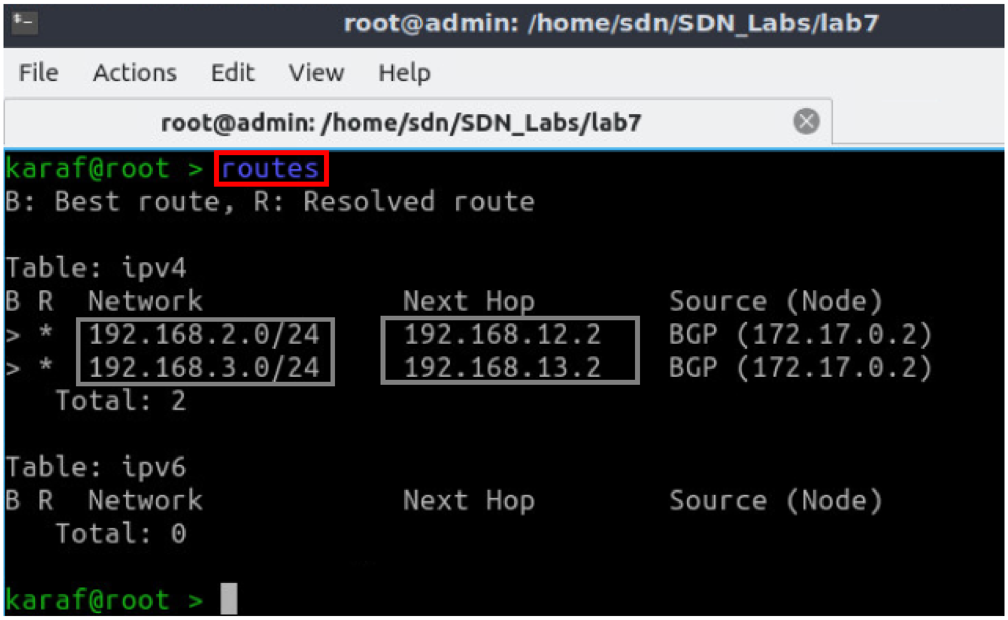

Figure 64. Showing the routing table of the SDN-IP application.

Consider Figure 64. The networks 192.168.2.0/24 and 192.168.3.0/24 are inserted in the routing table of the SDN-IP application.

6 Verifying the connectivity between the networks

Step 1. Open router r1 terminal and type the following command to show the BGP table.

```
show ip bgp
```

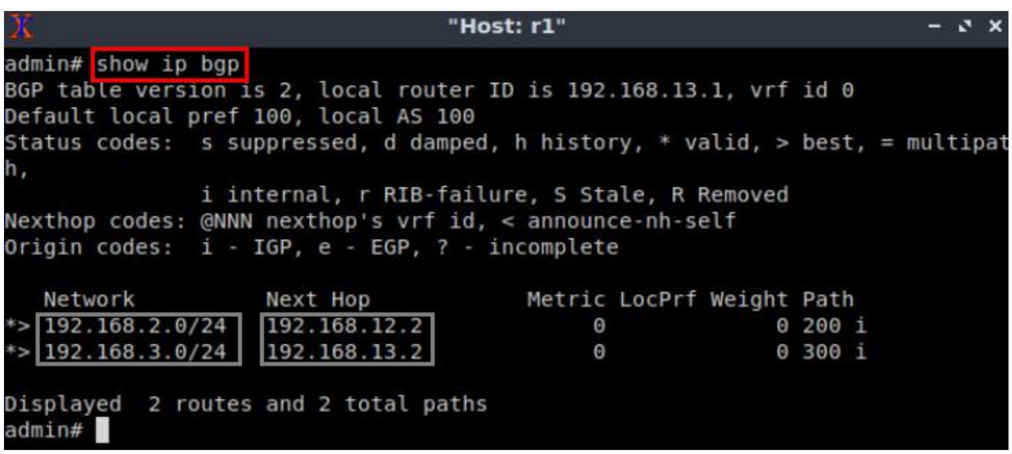

Figure 65. Showing the BGP table of router r1.

Consider Figure 65. The networks 192.168.2.0/24 and 192.168.3.0/24 are inserted in the BGP table of router r1. The next hops to reach these networks are 192.168.12.2 (router r2) and 192.168.13.2 (router r3), respectively.

Step 2. In router r1 terminal, type the following command to show the routing table.

```
show ip route
```

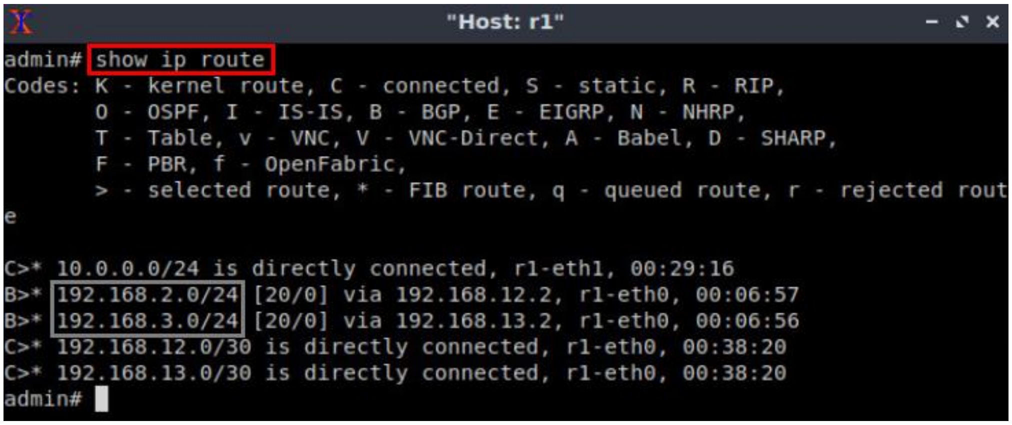

Figure 66. Showing the routing table of router r1.

Consider Figure 66. The networks 192.168.2.0/24 and 192.168.3.0/24 advertised by routers r2 and r3, respectively, are added to the routing table of router r1.

Step 3. Open router r2 terminal and type the following command to show the routing table.

```
show ip route
```

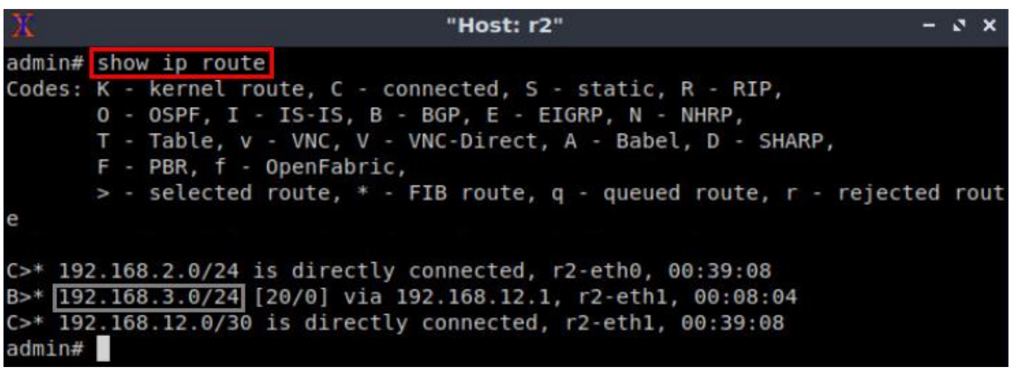

Figure 67. Showing the routing table router r2.

Consider Figure 67. The network 192.168.3.0/24 is added to the routing table of router r2, and it is reachable via 192.168.12.1 (router r1).

Step 4. Open host h1 terminal and type the following command to test the connectivity with host h2.

```
ping 192.168.3.10
```

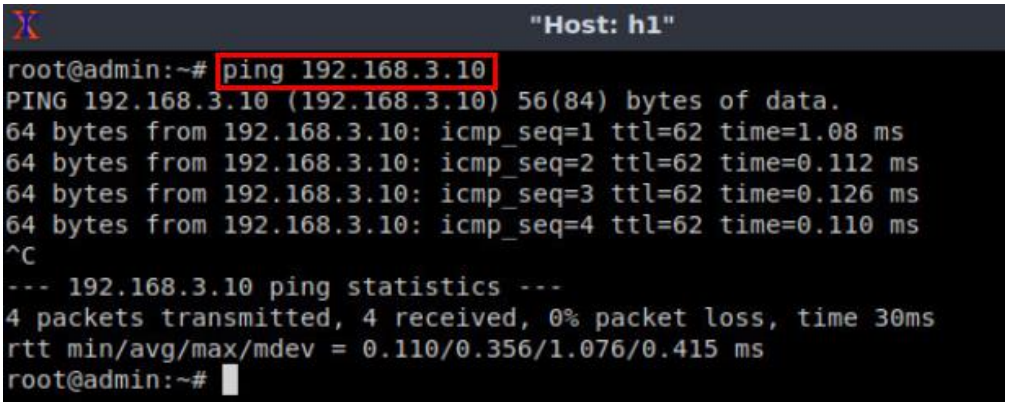

Figure 68. Pinging host h2 from host h1.

Consider Figure 68. The result of pinging host h2 from h1 shows a successful connectivity. Thus, BGP is successfully configured and integrated between legacy and SDN networks.

This concludes Lab 7. Stop the emulation and then exit out of MiniEdit and Linux terminal.

References

1. A. Tanenbaum, D. Wetherall, "*Computer networks*", 5th Edition, Pearson, 2012.
2. P. Goransson, C. Black, T. Culver. "*Software defined networks: a comprehensive approach*". Morgan Kaufmann, 2016.
3. P. Lin, J. Hart, U. Krishnaswamy, T. Murakami, M. Kobayashi, A. Al-Shabibi, K.C. Wang, J. Bi. "*Seamless interworking of SDN and IP*". In Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM, pp. 475-476, 2013.
4. ONOS project, "*SDN-IP*", [Online]. Available: https://wiki.onosproject.org/display/ONOS/SDN-IP.