INTRODUCTION TO P4 PROGRAMMABLE DATA PLANES

Lab 1: Introduction to Mininet

Document Version: 01-25-2022

Contents

Overview

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

Objectives

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

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

Lab settings

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

Table 1. Credentials to access the Client machine.

Device	Account	Password
Client	admin	password

Lab roadmap

This lab is organized as follows:

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

1   Introduction to Mininet

Mininet is a virtual testbed enabling the development and testing of network tools and protocols. With a single command, Mininet can create a realistic virtual network on any type of machine (Virtual Machine (VM), cloud-hosted, or native). Therefore, it provides an inexpensive solution and streamlined development running in line with production networks[1]. 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.
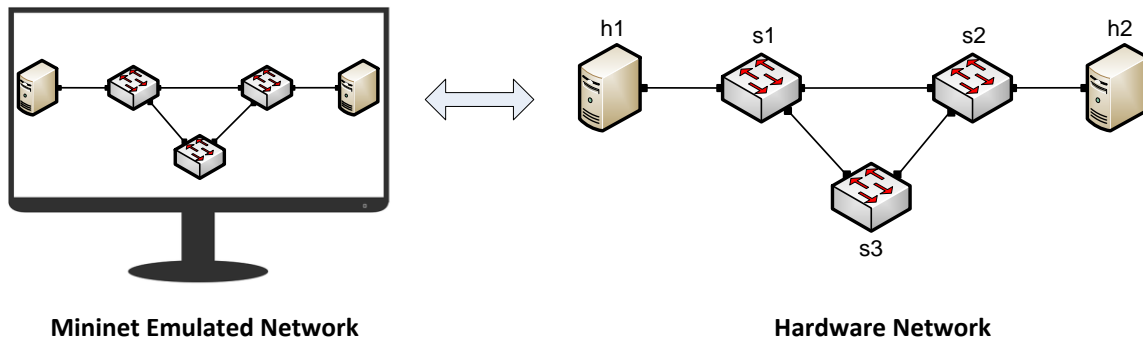- Open-source environment backed by a large community contributing extensive documentation.

Mininet Emulated Network   **Hardware Network**

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 switch[4]. 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.
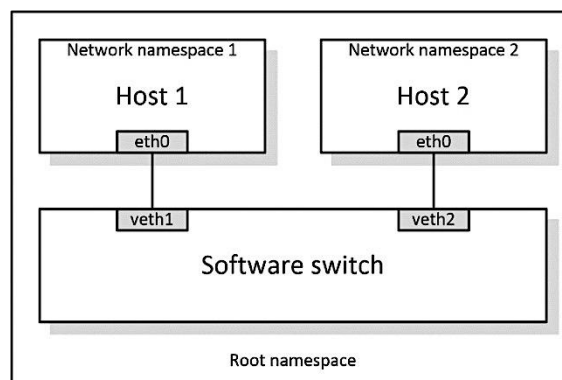

Figure 2. Network namespaces and virtual Ethernet links.

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

In following subsections, you will start Mininet using the Linux CLI.

2.1   Invoke Mininet using the default topology

Step 1. Launch a Linux terminal by clicking on the Linux terminal icon in the task bar.
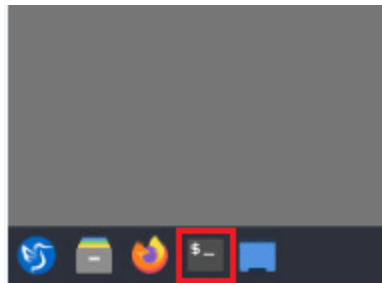
Figure 3. Linux terminal icon.

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

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

```
sudo mn
```

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.
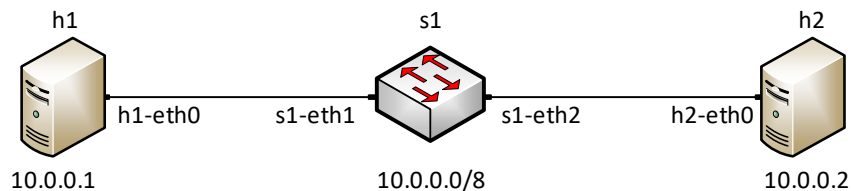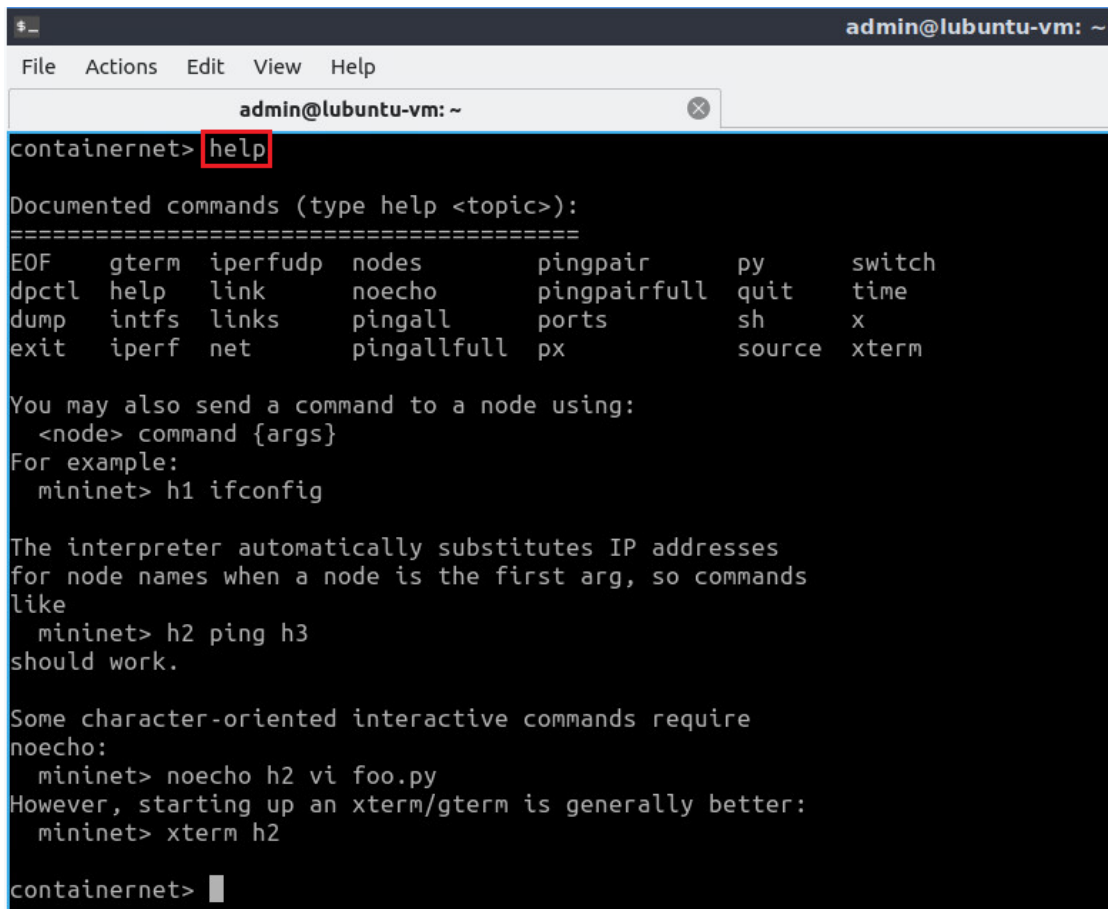

Figure 5. Mininet's default minimal topology.

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

```
containernet>
```

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

```
help
```
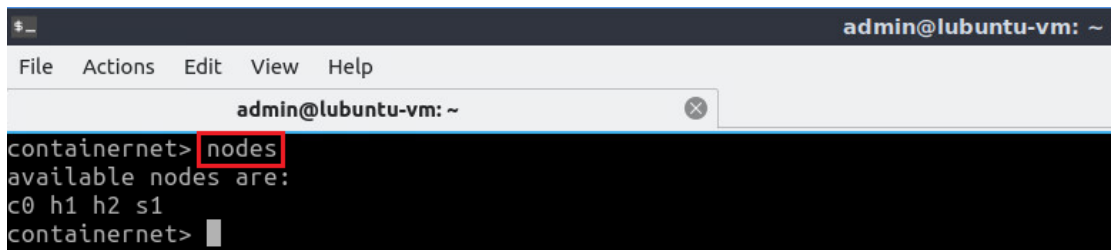
Figure 6. Mininet's `help` command.

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

```
nodes
```


Figure 7. Mininet's `nodes` command.

The output of the `nodes` command shows that there is a controller (c0), 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
```

Figure 8. Mininet's `net` command.

The output of the `net` 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*.
4. Controller c0 does not have any connection.

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
```


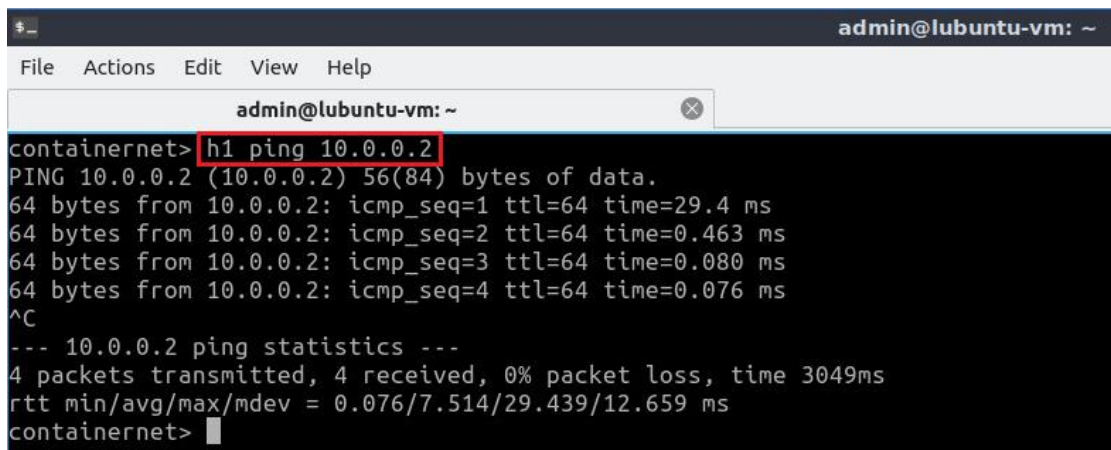Figure 9. Output of `h1 ifconfig` command.

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

2.2   Test connectivity

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

Step 1. On the CLI, type the command shown below. The command `h1 ping 10.0.0.2` 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
```


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

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

```
exit
```

Figure 11. Stopping the emulation using `exit`.

If Mininet were to crash for any reason, the `sudo mn – c` command can be utilized to clean a previous instance. However, the `sudo mn -c` command is often used within the Linux terminal and not the Mininet CLI.

Step 3. After stopping the emulation, close the Linux terminal by clicking the ⊠ in the upper-right corner.
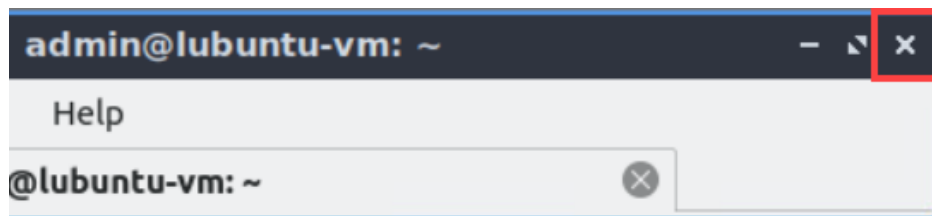

Figure 12. Closing the Linux CLI.

3   Build and emulate a network in Mininet using the GUI

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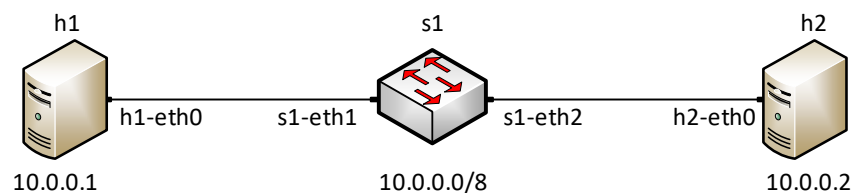

Figure 13. Lab topology.

3.1   Build the network topology

Step 1. A shortcut to MiniEdit is located on the machine's Desktop. Start MiniEdit by double-clicking on MiniEdit's shortcut. When prompted for a password, type `password`. MiniEdit will start, as illustrated below.
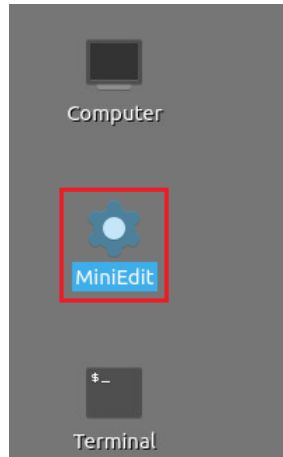
Figure 14. MiniEdit Desktop shortcut.
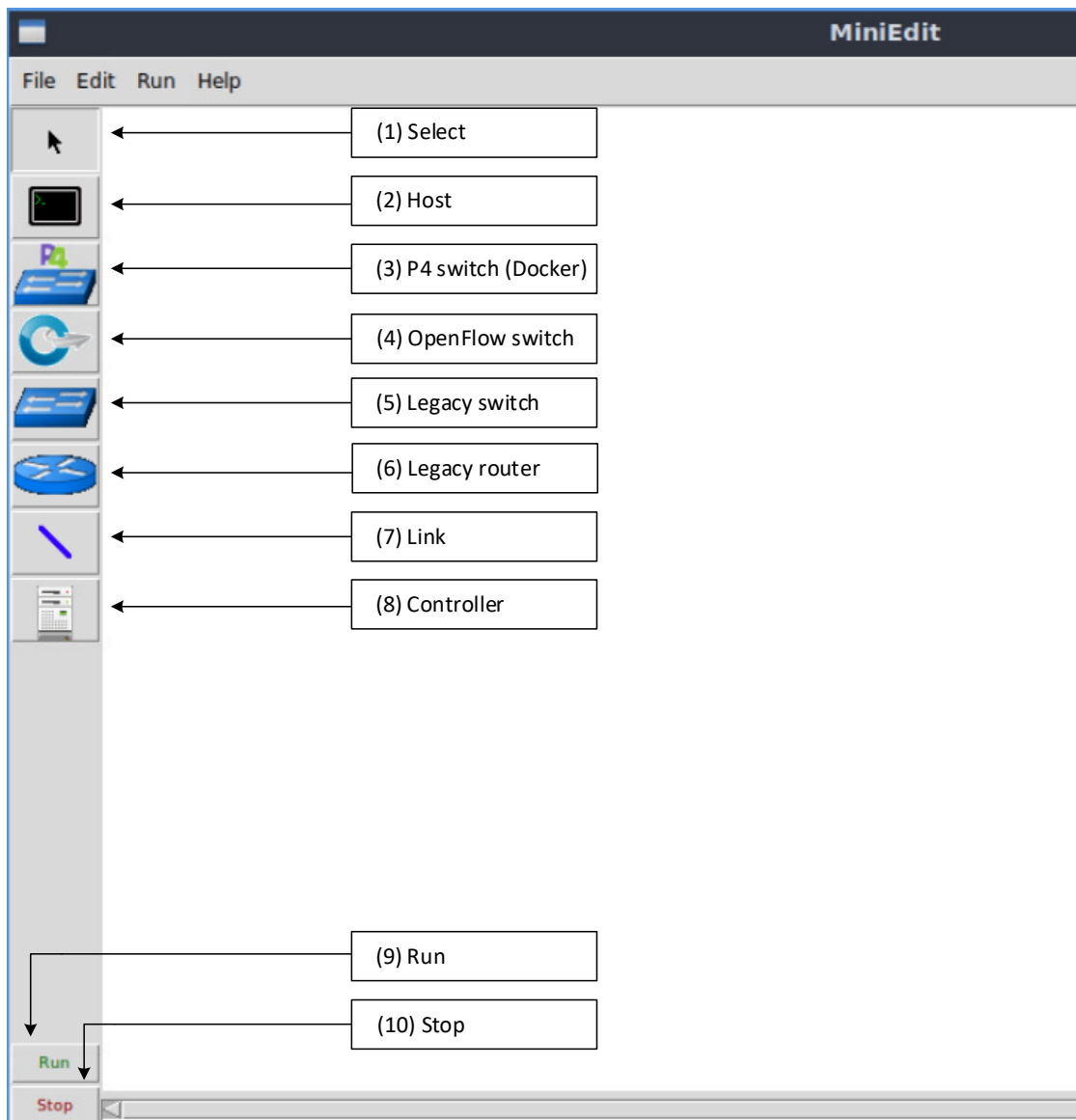
MiniEdit will start, as illustrated below.

Figure 15. 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. P4 switch (Docker): allows the addition of P4 switch. After clicking this button, click anywhere in the blank canvas to insert the P4 switch.
4. OpenFlow switch: allows the addition of a new OpenFlow-enabled switch. After clicking this button, click anywhere in the blank canvas to insert the switch.
5. Legacy switch: allows the addition of a new Ethernet switch to the topology. After clicking this button, click anywhere in the blank canvas to insert the switch.
6. Legacy router: allows the addition of a new legacy router to the topology. After clicking this button, click anywhere in the blank canvas to insert the router.
7. 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.
8. Controller: allows the addition of a new OpenFlow controller.
9. Run: starts the emulation. After designing and configuring the topology, click the run button.
10. Stop: stops the emulation.

Step 2. To build the topology illustrated in Figure 13, two hosts and one switch must be deployed. Deploy these devices in MiniEdit, as shown below.
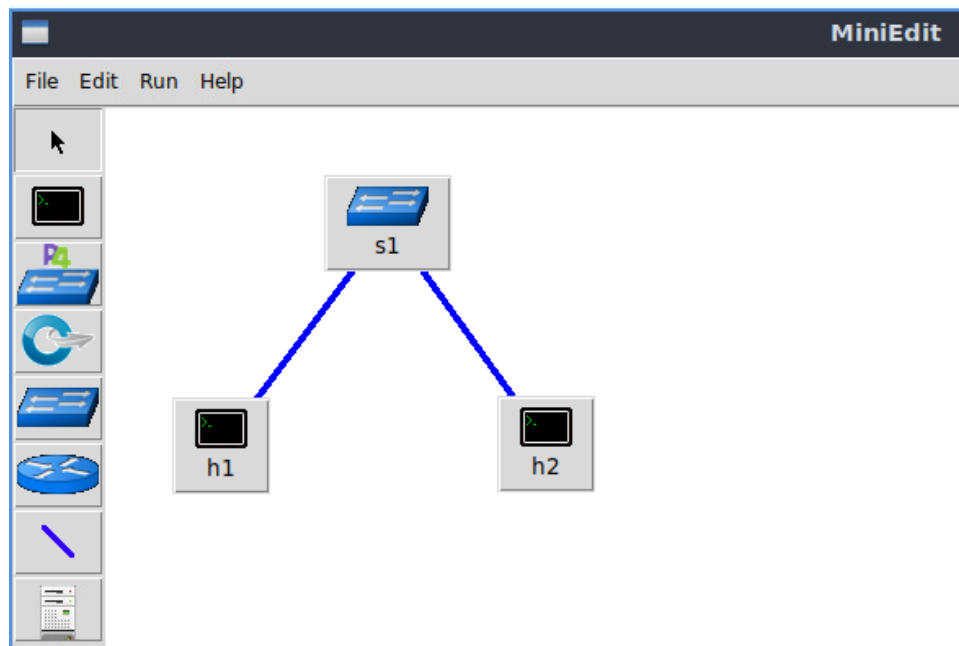

Figure 16. MiniEdit's topology.

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

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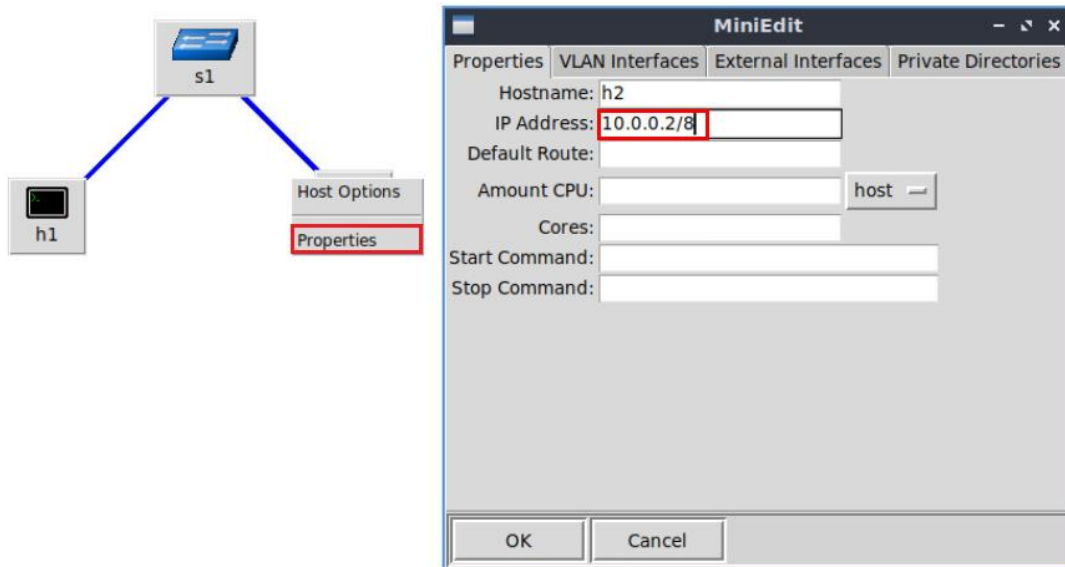

Figure 17. Configuration of a host's properties.

3.2   Test connectivity

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

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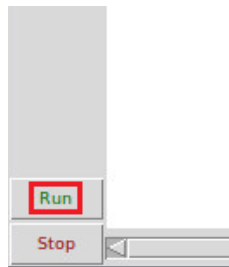

Figure 18. Starting the emulation.

Step 2. Open a terminal by right-clicking on host h1 and select *Terminal*. This opens a terminal on host h1 and allows the execution of commands on the host h1. Repeat the procedure on host h2.
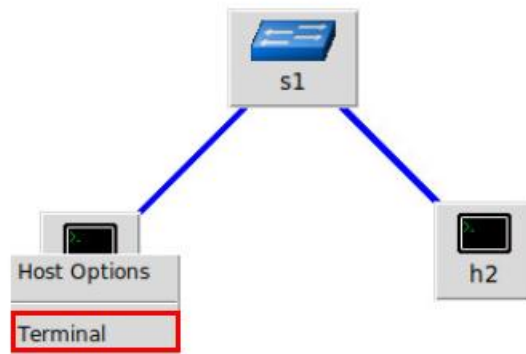
Figure 19. Opening a terminal on host h1.

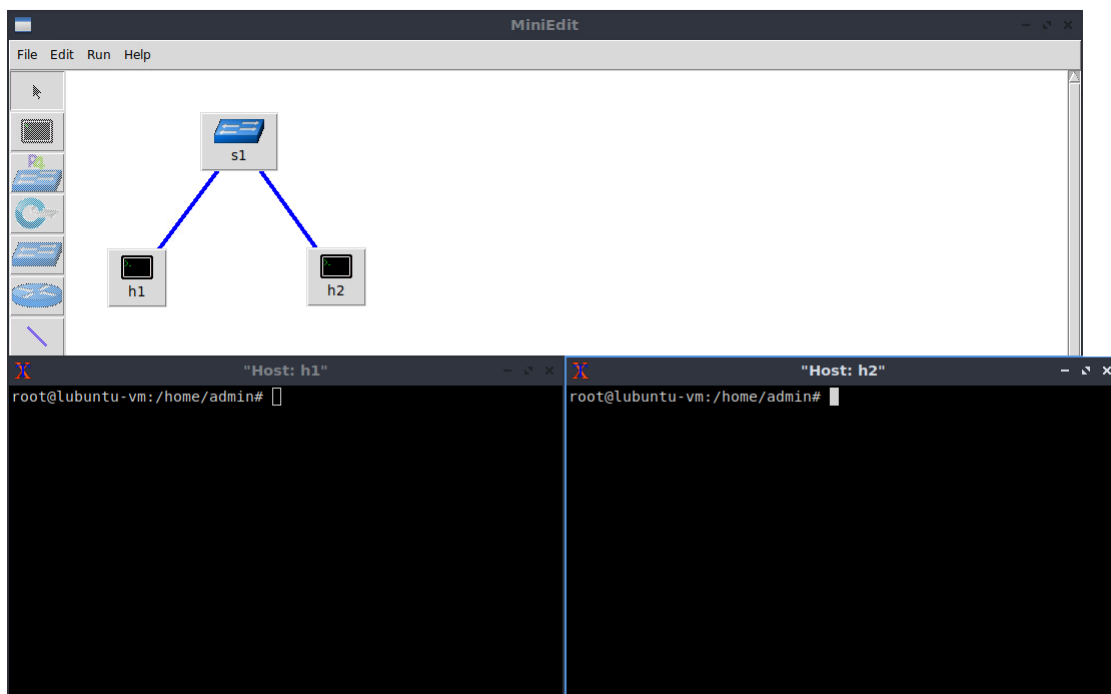The network and terminals at host h1 and host h2 will be available for testing.

Figure 20. 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
```

Figure 21. 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
```


Figure 22. Connectivity test using `ping` command.

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

Figure 23. 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. Right-click on host h1 and select *Properties*. Delete the IP address, leaving it unassigned, and press the *OK* button as shown below. Repeat the procedure on host h2.
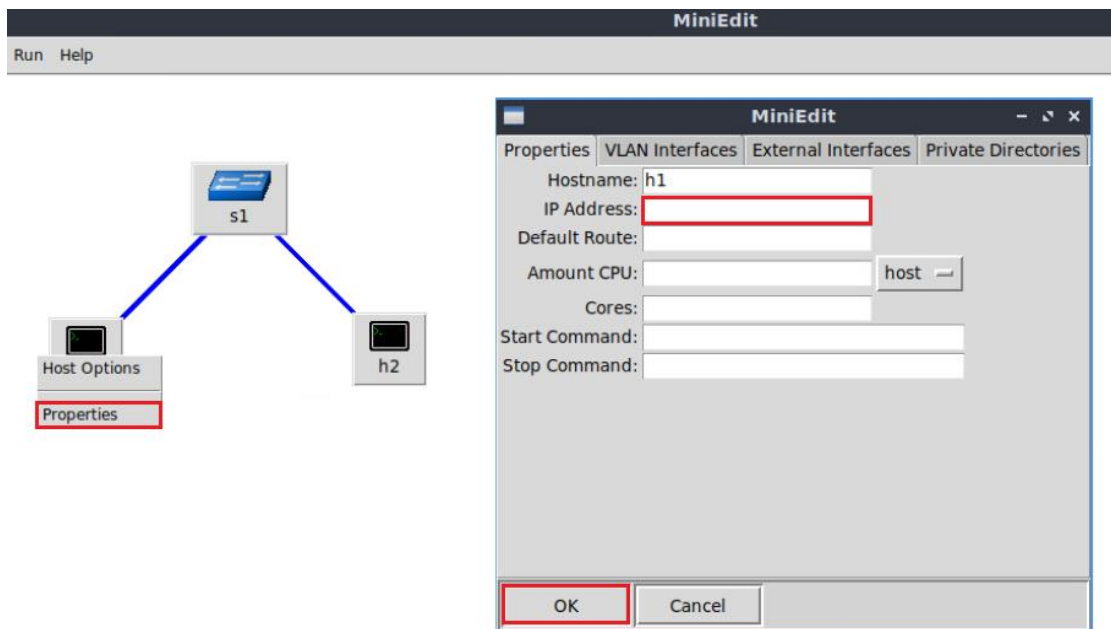

Figure 24. Host h1 properties.

Step 2. In the MiniEdit application, navigate to *Edit > Preferences*. 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.
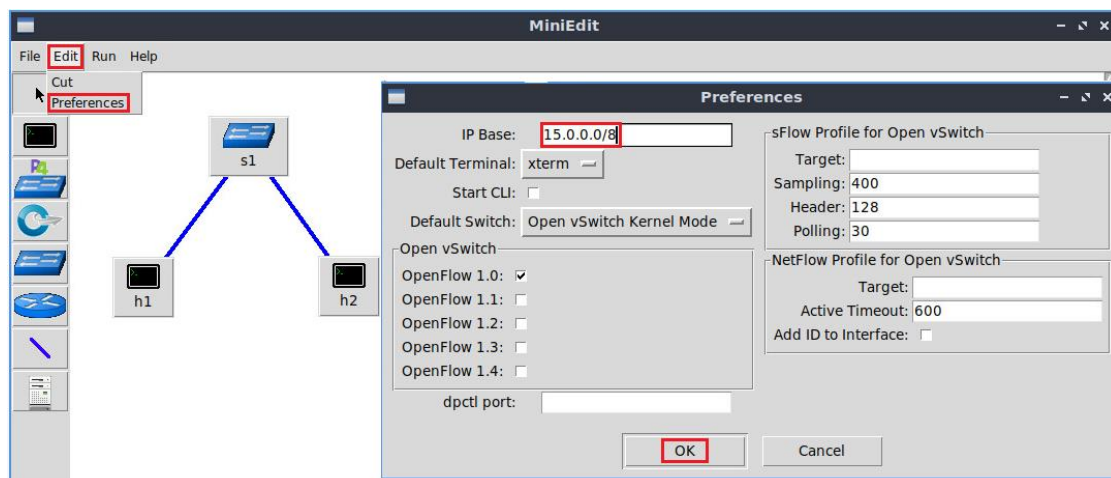

Figure 25. Modification of the IP Base (network address and prefix length).

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.
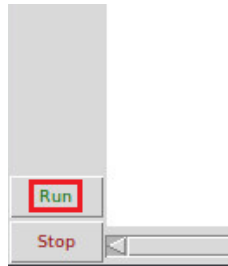

Figure 26. Starting the emulation.

Step 4. Open a terminal by right-clicking on host h1 and select *Terminal*.
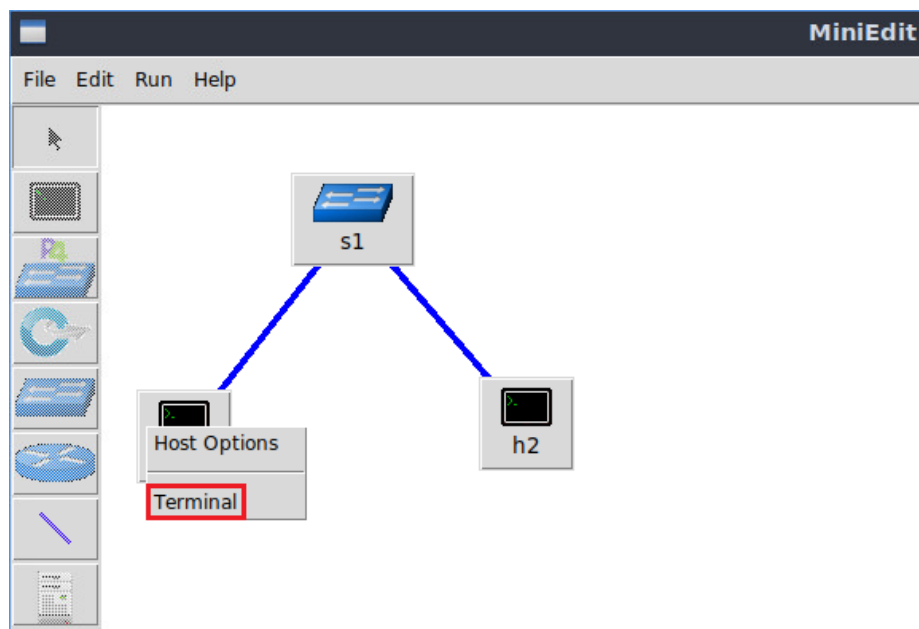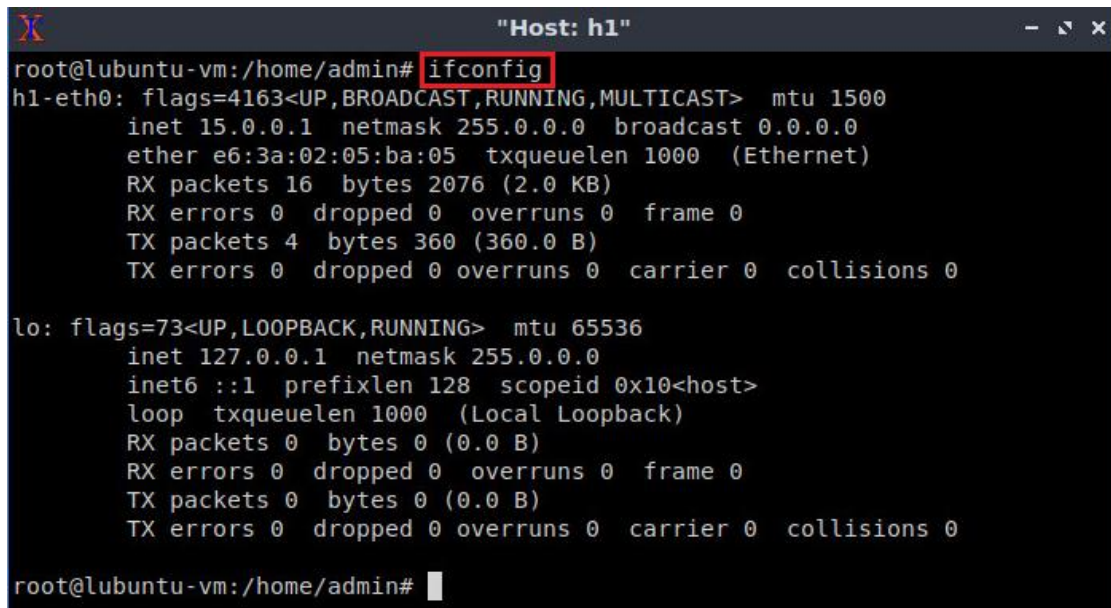

Figure 27. Opening a terminal on host h1.

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
```

Figure 28. 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 29. 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. In the MiniEdit application, save the current topology by clicking *File*. Provide a name for the topology and notice *myTopology* as the topology name. Ensure you are in the *lab1* folder and click *Save*.
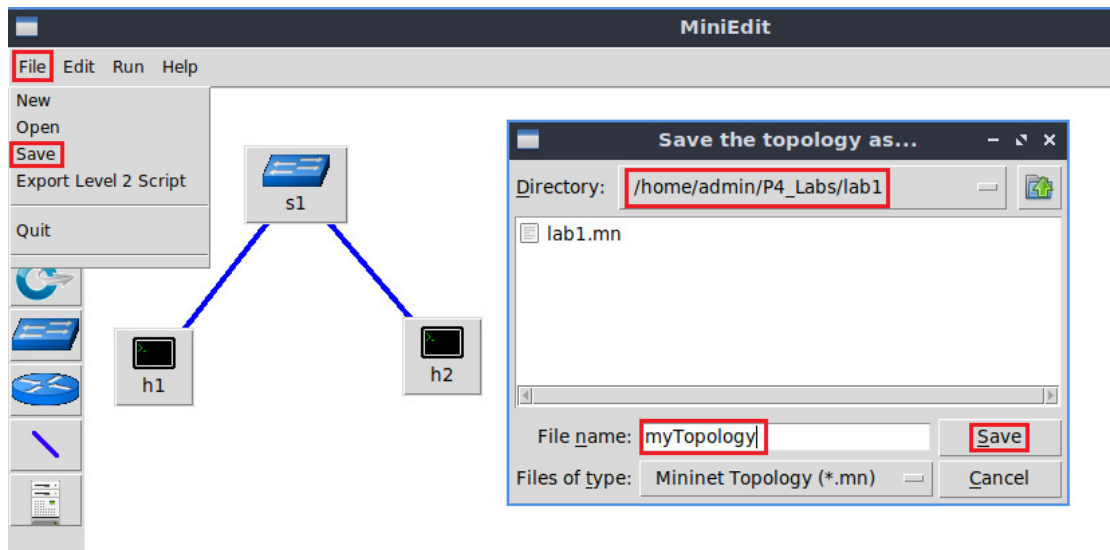
Figure 30. Saving the topology.

Step 2. In the MiniEdit application, load the topology by clicking on *File* then *Open*. Navigate to the *lab1* folder and search for the topology file called *lab1.mn* and click on Open. A new topology will be loaded to MiniEdit.
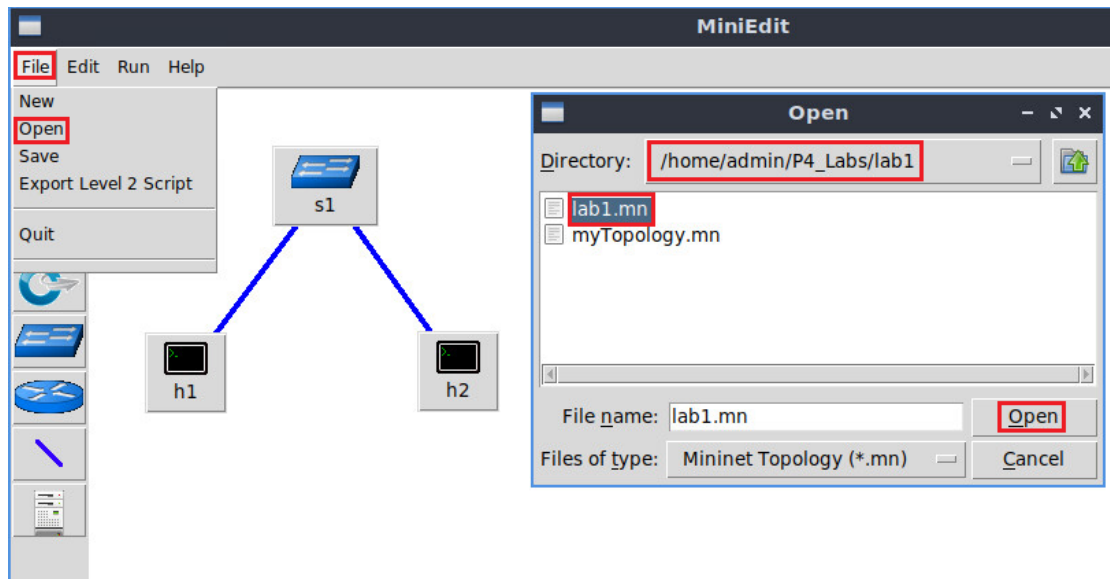

Figure 31. Opening a topology.

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

References

1. Mininet walkthrough. [Online]. Available: http://Mininet.org.
2. Mckeown N., Anderson T., Balakrishnan H., Parulkar G., Peterson L., Rexford J., Shenker S., Turner J., "*OpenFlow*," ACM SIGCOMM Computer Communication Review, vol. 38, no. 2, p. 69, 2008.

3. Esch J., "*Prolog to, software-defined networking: a comprehensive survey*," Proceedings of the IEEE, vol. 103, no. 1, pp. 10–13, 2015.
4. Dordal P., "*An Introduction to computer networks*,". [Online]. Available: https://intronetworks.cs.luc.edu/.
5. Lantz B., Gee G. "*MiniEdit: a simple network editor for Mininet.*" 2013. [Online]. Available: https://github.com/Mininet/Mininet/blob/master/examples.