Software Defined Networking

Laboratorial Class

2nd Semester 2020/2021

Introduction to Mininet and OpenFlow

Eliana Neuza da Silva

Fernando Mira da Silva

April 2121

1. Introduction

Before starting the lab work it is important to understand certain concepts that will be briefly addressed below.

1.1 Mininet and the OpenFlow protocol

Mininet[1] is a network emulator, utilized for testing, that uses lightweight virtualization making a system of hosts, switches, routers, and links appear as a complete realistic network on top of a single Linux kernel. This emulator creates kernel user-space OpenFlow switches, controllers that control the switches, and hosts to communicate over the emulated network. Mininet makes use of network namespaces and connects switches and hosts using virtual ethernet (veth) pairs.

Mininet allows full topologies and packet forwarding customization, introducing an environment for testing and deploying SDN networks, the OpenFlow protocol and other SDN controllers.

OpenFlow[2] is a communication protocol that allows a piece of software to communicate and control the path of network packets through the switches. This management through software is performed in a controller, that has the task of configuring network devices (routers, switches, ...) to achieve the best paths and modify traffic flows according to network, administration and applications needs.

SDN is a network architecture that centralizes control, due to the mentioned controller, allowing network changes or reformulations to happen without reprogramming the physical network elements. Easy network configuration, management, monitoring and introduction of programmability are characteristics offered by the SDN architecture.

SDN is mostly notable by the concept of decoupling the data plane from the control plane. Its architecture has three layers, the application layer, the control layer and the infrastructure layer.

2. Goals

The second lab work will have the main goal of using Mininet to emulate complete networks and their orchestration using the OpenFlow protocol to perform flow configuration either by doing it manually or using an OpenFlow controller.

3. Evaluation

To grade your work for the lab class #2 you will need to deliver a report, through the Fenix platform, by the end of the second class. All reports should have the group number as well as the names and numbers of each member.

The report should explain all the commands and steps that you used to set up and test the topologies for each the second exercise.

[1] mininet.org
[2] McKeown, Nick, et al. "OpenFlow: enabling innovation in campus networks." ACM SIGCOMM Computer Communication Review 38.2 (2008): 69-74.

4. Setup and Testing

Use any Linux distribution with graphic environment, e.g. ubuntu 18.04 desktop, with `root` access or a user with `sudo` privileges, with Mininet and Wireshark.

4.1 Mininet walkthrough

As a first exercise, you should perform the entire Mininet walkthrough[3] to understand its basic functionalities. Mininet's default topology, the one used in the walkthrough, is represented in Figure 1.

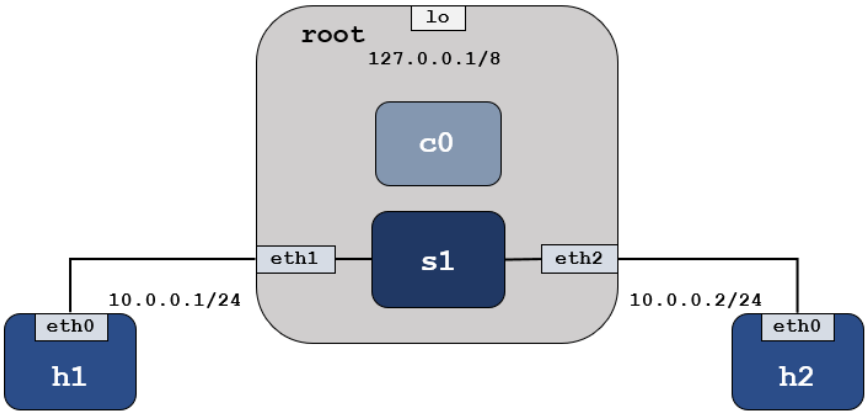

Figure 1 – Mininet's default topology.

As you could see from following the walkthrough, the OpenFlow controller creates flow entries that substitute the IP rules that you would need to place in the switch in the case of not having a controller as shown below:

1. Run Mininet with the following start up configuration:

```
$ sudo mn --mac --controller=none
```

2. Attempt to ping h2 from h1. This won't work.

```
mininet> h1 ping -c5 h2
```

3. Check the flow table of the switch:

```
$ sudo ovs-ofctl dump-flows s1
```

4. Add the rules in the switch:

```
$ sudo ovs-ofctl add-flow s1 in_port=1,actions=output:2
$ sudo ovs-ofctl add-flow s1 in_port=2,actions=output:1
```

5. Test connectivity between h1 and h2.

6. Run Mininet with the following command:

```
$ sudo mn --mac
```

[3] mininet.org/walkthrough

7. Open Wireshark and start a capture on the `loopback` interface, applying the filter `openflow_v1`.

8. Attempt to ping h2 from h1.

```
mininet> h1 ping -c5 h2
```

Explain in full detail the observed behavior on steps 1-8.

Explain, in particular, what is the ovs-ofctl command, how does it work and how does it communicate with s1. Explain also what the option –mac usec in point 6 does.

4.2 Three-node network

For the second exercise you will build the network represented in Figure 2. In this example the red link between **S1** and **S3** has a bandwidth of 100 Mb/s and 10 ms of delay and the other two links have a bandwidth of 1000 Mb/s. The objective is that traffic coming from **h3** to always go through the red link and traffic coming from **h4** always go through the other. To do this you will need to create the topology without a controller and add all the needed forwarding rules (flow entries) to each switch.

You will need to create a custom topology, as shown in the Mininet walkthrough[3], perform Wireshark captures and `iperf` tests to verify the traffic paths and the links velocities respectively.

Your report should have the custom topology that you created, the several Wireshark captures and Mininet screenshots of the tests performed for control traffic, link velocity and connectivity. Pr

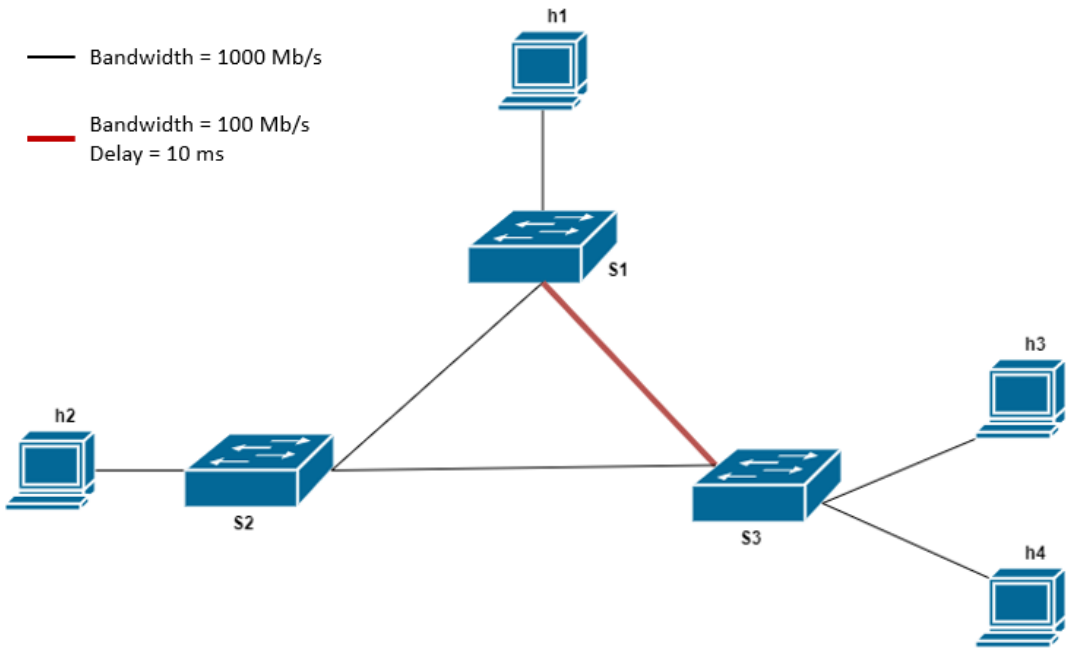

Figure 2 – Mininet custom topology.