ICT1010 Computer Networks (AY2021/22)

Individual Assignment: Self-learning lab worksheet

Mininet Emulation: Software Defined Networks (SDN) – Control & Data Plane

LEARNING OUTCOMES

Upon completion, you should be able to:

- Understand the concept of software defined networking (SDN)
- Apply and infer the working of the SDN concepts, i.e., separation of data and control plane using Mininet Emulation Software with Floodlight controller

NAME: Gu Boyuan

STUDENT ID: 2103210

REQUIRED SOFTWARE

- VMware workstation (trial version) or VMware player or VirtualBox
- Mininet
- Floodlight controller
- Wireshark/Tshark

Submission:

1. Fill this lab worksheet (pg 2 - 5), answer all questions within the box provided and upload to Gradescope by **20th Mar, 2022 (Sun) by 11.59 pm.**
2. This is an individual lab exercise comprising of 20 marks and the weightage of this submission towards the continuous assessment of the module is **5%.**
3. Clearly fill in your name, ID above in the space provided (in blue) before uploading this worksheet to the Gradescope. Otherwise, your submission will not be recognized by Gradescope. You may access Gradescope from the module LMS page or directly from the below link. https://www.gradescope.com/courses/376505

Part1 – Setup Mininet with Default Controller:

Mininet is a network **emulation** tool widely used for educational purposes. The objective of Part 1 is to get a good understanding of Software Defined Networking, OVS, and the OpenFlow protocol through the Mininet emulator. Note that the OVS code running on the emulator is the exact replica of the code that is running on any OpenFlow supported hardware.

1.1. Download the Mininet VM 2.3.0 Ubuntu VM Image (latest release) https://github.com/mininet/mininet/releases/

1.2. Boot the Mininet VM and setup http://mininet.org/vm-setup-notes/

1.3. Log in to VM with username: mininet and password: mininet

1.4. Find the IP address of your VM using *ifconfig*

1.5. Come out of your VM (do not close the VM). From your host machine SSH to your mininet VM (e.g., ssh mininet@*IPaddress*)

1.6. Start the mininet emulator with $sudo mn

1.7. This will create a default mininet topology (Fig. 1) with:

   a. 2 end hosts & 1 switch (data plane) and

   b. a default reference SDN controller (control plane)

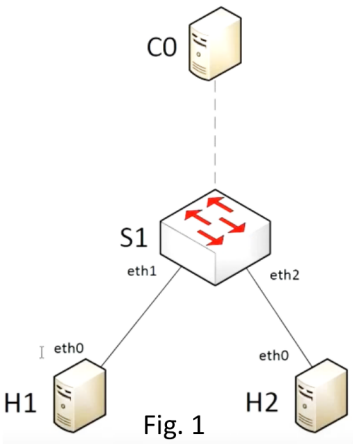

Fig. 1

1.8. Go to the mininet prompt and type help command to see all the available commands

1.9. Test whether h1 can ping h2 from the mininet prompt.

a. Is the ping successful?
b. How will the first packet from h1 be handled by the switch s1?
c. How is the first packet handled different from the traditional network?

Answer 1.9 a, b & c in the below box *[3 marks]*

a. Yes

b. The switch first adds the mac address of h1 to its forwarding table. In addition, since the switch initially did not know which port h2 is connected to, it duplicates the packets and sends them out as a broadcast, except the port that the packet was initially received on.

c. The first packet is sent out from the switch as a broadcast due to the mac address of h2 not being in the forwarding table, as compared to the later packets which are forwarded to h2 as h2's mac address is now present in the forwarding table of S1.

1.10. What are all the available nodes in the topology? Which command will you use to see all the details of the nodes in the network?

Include the screenshot(s) in the below box and describe *[2 mark]*

1.11. From the mininet prompt, type the command *iperf* and *iperfudp* to measure the tcp and udp bandwidth, respectively.

1.12. Quit from mininet and do a cleanup using the command *sudo mn -c*.

1.13. Write a python script to create a custom topology (custom_topo.py) as shown in Fig 2 (below) with 3 switches, and 2 end hosts in the /home/mininet/ folder and run using mininet (refer to sample script in /mininet/custom).

sudo mn --custom custom_topo.py --topo mytopo

(Note: For Fig.2 topology :wqthe eth_no. need not be the same)

1.14. Provide the python script used to create the topology in Fig.2 in the below box *[6 marks]*

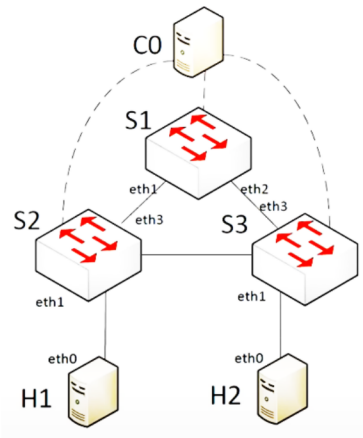

Fig. 2

1.15. On the mininet prompt ping from h1 to h2.

(a) What output do you get?
Include the screenshot of the output below in the below box *[1 marks]*

(b) Run Wireshark/Tshark from the mininet vm and capture traffic from the *lo* (loopback) interface. When you ping from h1 to h2 what OpenFlow messages do you see?
Include the screenshot of the output below in the below box *[3 marks]*

Part2 – Mininet with Remote the Controller Floodlight:

Floodlight is an OpenFlow enterprise-class SDN controller developed in Java. We will be using the Floodlight controller to connect to the hardware SDN switches (without mininet) in Lab 8. The objective of Part 2 is to get familiarized with the Floodlight SDN controller.

2.1. Run *sudo mn -c* from the terminal to cleanup old configs

2.2. From the Mininet terminal, download dependencies for Floodlight master and above:
sudo apt-get install build-essential ant maven python-dev

2.3. To download and build Floodlight, git clone git://github.com/floodlight/floodlight.git

2.4. After step 2.3, installation is complete, follow the below steps to build the Floodlight controller and set root permissions.

```
$ cd floodlight
$ git submodule init
$ git submodule update
$ sudo ant
```

Ensure BUILD SUCCESSFUL, then enter:

```
$ sudo mkdir /var/lib/floodlight
$ sudo chmod 777 /var/lib/floodlight
```

2.5. **Alternatively** (instead of step 2.1 - 2.4), you can download the Floodlight VM that has a build in Floodlight+Mininet+Wireshark installation.
https://floodlight.atlassian.net/wiki/spaces/floodlightcontroller/pages/8650780/Floodlight+VM
After installing the Floodlight VM, build and install Floodlight SDN controller java program (username: floodlight and password: floodlight)

```
floodlight$ sudo ant
```

2.6. Start the Floodlight controller after installation in Step 2.4 or 2.5

```
floodlight$ java -jar target/floodlight.jar
```

2.7. Add the Floodlight controller as the remote controller when running Mininet for topology in Fig 2.
For example:

```
mininet@mininet-vm:~$ sudo mn --custom ./customtopo.py --topo mytopo --controller=remote,ip=192.168.74.130,port=6653 --switch ovsk,protocols=OpenFlow13
*** Creating network
```

2.8. Is the remote controller added to the network emulator? How do you show this?
Include the relevant screenshot of the output below in the below box *[3 marks]*

2.9. Issue pingall command to from the mininet prompt. Does is ping successfully or fails? why?
Include the relevant screenshot of the output below in the below box [2 marks]

----------END OF ASSIGNMENT ----------

Estimated time to complete Assignment is 1-3 days based on familiarity with Linux & VMs

If you are familiar with Linux, you might as well try out the native installation of Mininet and Floodlight controller from the source below.
http://mininet.org/download/
https://floodlight.atlassian.net/wiki/spaces/floodlightcontroller/pages/1343544/Installation+Guide

Reference Links:

1. https://floodlight.atlassian.net/wiki/spaces/floodlightcontroller/pages/1343544/Installation+Guide
2. https://floodlight.atlassian.net/wiki/spaces/floodlightcontroller/pages/8650780/Floodlight+VM
3. https://github.com/mininet/mininet/releases/
4. http://mininet.org/vm-setup-notes/
5. http://mininet.org/
6. http://mininet.org/download/
7. http://mininet.org/walkthrough/#custom-topologies