Software-Defined Networking

Lab 4

Advanced Mininet, Open vSwitch, and SDN Controllers

University of Colorado Boulder

Department of Computer Science

Professor Levi Perigo, Ph.D.

# Lab Summary

Understanding Mininet and popular SDN controllers is critical to understanding SDN. The purpose of this lab is to explore some of the advanced features of Mininet and become familiar with some of the popular SDN controllers in industry. Knowing how to use advanced features in Mininet and understanding specific commands and output, will aid in your understanding of SDN. The experience gained from using the controllers in this lab will facilitate your understanding of SDN controllers as you expand on the foundations from this lab in the rest of the course.

The objectives of this lab are to be used as guidelines, and additional exploration by the student is strongly encouraged.

# Objective 1:

# PART A: Connect Mininet to OpenDaylight (ODL)

1. Refer the Lab 0 document to initialize OpenDaylight (ODL).
2. Enable the ODL web interface and all the features (use the Lab 0 document for assistance).
3. Login to the ODL web interface.
4. Login to Mininet.
   1. Start a new Mininet topology that connects to a remote controller (ODL), uses easy to read MAC addresses, uses OpenFlow v1.3, and uses a network topology that is a tree with depth of 2 and a fanout of 3.
   2. Explain in detail what the tree topology is including the depth and fanout settings. **[5 points]**

The tree topology creates 2^n switches based on the depth specified and the fanout is the number of hosts attached to the branches of the tree.

1. Within the GUI of ODL, does the topology show any hosts connected to the switches? Why or why not? If not, how would you get the hosts in Mininet to be displayed in the ODL topology? **[5 points]**

Initially it does not, I would have to issue the “pingall” command to the hosts to get the hosts to show up.

1. Login to the Mininet VM with a duplicate session.
   1. Within the Mininet VM, you are going to issue Open vSwitch commands to view statistics and flow table entries of the switches.
   2. Provide a combined screenshot for the following operations: **[5 points]**
   3. Command to show all ports and connections for the network.
   4. Command to show all ports for Switch 2.

A computer screen shot of a computer program

Description automatically generated

A computer screen shot of a program

Description automatically generated

* 1. Write steps or provide a screenshot of the same functionality as in [b (ii)] in the GUI of ODL. **[5 points]**

A screenshot of a computer

Description automatically generated

* 1. Issue the command that shows all the current flow table flow entries in switch 2.
     1. What command did you use? **[5 points]**

A screen shot of a computer

Description automatically generated

* + 1. Provide a screenshot of this in the GUI of ODL**. [5 points]**

A screenshot of a computer

Description automatically generated

* + 1. What is the flow entry with the lowest priority value? Explain what that entry is. **[5 points]**

The flow entry with the lowest priority value is the first one with priority 2. It is saying that if traffic comes in from port 3, send it out port 1,2,4 and the controller port.

* + 1. Generate traffic from hosts connected to switch 2 and refresh the flow table; were additional flow entries added? Why or why not? **[5 points]**

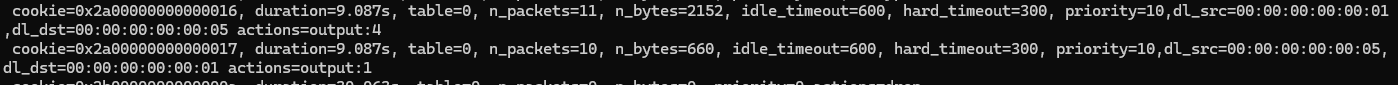
Starting from a blank topology and then pinging h1 and h3, two flows were added. They were added because the switch is in l2 learning mode and adds flows for l2 connectivity.

* 1. Find HTTP Flows
     1. On Host1 run a HTTP server.
     2. Perform a “wget” from H5 to the web server (H1).
        1. What command did you use? **[5 points]**

h1 python -m SimpleHTTPServer 80 &

h5 wget 10.0.0.1

* + 1. Within ODL GUI or OVS, select a switch in the path, and highlight the flow entry that shows the HTTP traffic **[10 points]**



* 1. Exit and clean up Mininet.

# PART B: Flow management with Cisco OpenFlow Manager (OFM)

1. Refer the Lab 0 document to configure and initialize OFM.
2. Explain what the grunt command does? **[5 points]**

Grunt is a JavaScript Task Runner that basically runs a set of commands given a grunt config type file. It is like having an ansible playbook and running ansible on that playbook. It is running OFM for us with the GUI

1. On the ODL VM, start the ODL controller by installing the following features:

feature:install odl-restconf-all odl-openflowplugin-all odl-l2switch-all webconsole

1. On Mininet, clear any existing Mininet topologies and build a new topology comprising of a single switch and five hosts, easy to read mac addresses, ODL as remote controller, running on OpenFlow 1.3 protocol. Paste the output of the pingall command. **[5 points]**.

A screen shot of a computer program

Description automatically generated

1. Paste the screenshot of your browser displaying Cisco OFM. It should display the Mininet topology. [**5 points]**

A computer network diagram with blue dots and black rectangles

Description automatically generated

1. Create a flow entry on switch1 using Cisco OFM to provide the following functionality:
   1. Match the packets with source MAC address of host five and destination MAC address of host one. If there are any matches, then switch1 drops the corresponding packet.
   2. Set the priority of the flow entry as 50 and cookie value as 0x99.
   3. Paste the screenshot of the flow entry that’s created in Cisco OFM. **[10 points]** *(Hint: Use the Flow Management tab inside Cisco OFM)*



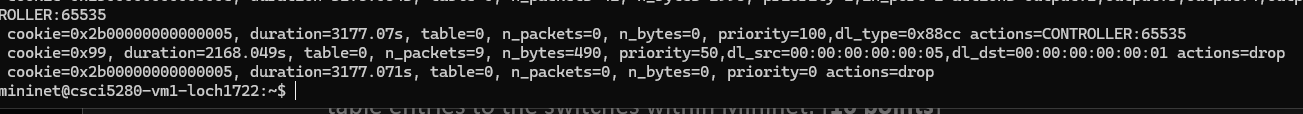
1. Explain the significance of priority value and cookie value of a flow entry. **[5 points]**

Priority is what the switch will use to determine a flow in the same table, the higher the priority, it will go first. The cookie value denotes what type of data will be transferred (packet, frame).

1. Issue pingall command in Mininet. Did all the hosts ping each other? If yes, why? If not, why? **[5 points]**

No because we added a flow to drop packets between h1 and h5.

1. From the Mininet VM, paste the screenshot of the command and corresponding output that shows the flow table highlighting the flow added by OFM. **[10 points]**
2. Explain the “flow logic” of how the Cisco OFM application is able to add OpenFlow flow table entries to the switches within Mininet. [**10 points**]



# Objective 2 - Connect Mininet to ONOS

1. Refer the Lab 0 document to initialize ONOS.
2. Follow the instructions from the Lab 0 document to install apps and activate them.
3. Then start a Mininet topology that connects to a remote controller, uses easy to read MAC addresses, uses OpenFlow v1.3, and uses a network topology that is a torus with 3 and 3, and change the IP address scheme from the default to 192.168.1.0/24.
   1. Explain in detail what the torus topology is including the additional “3” settings. **[5 points]**

A torus topology has a grid structure with rows and columns with all nodes connecting to one right, left, top and bottom. Where increasing the rows or columns keeps this connection type topology. The 3 additional setting is for the columns or rows.

* 1. Issue the “pingall” command in Mininet to bring up all the devices.

1. Within the GUI of ONOS:
   1. Provide a screenshot of the topology (from the previous Mininet commands, including all hosts). **[10 points]**

A screen shot of a computer

Description automatically generated

* 1. Navigate around the ONOS GUI and familiarize yourself with the features.
     1. Follow the “ONOS Tutorial” link on the desktop for additional information.

1. Within the Mininet VM, you are going to issue Open vSwitch commands to view statistics and flow table entries of the switches.
   1. Dump all OpenFlow flow tables from Switch3.
      1. What command did you use? How many flow tables were there? How do you know? What is the difference between flow tables and flow entries **[20 points]**

A black screen with white text

Description automatically generated

sudo ovs-ofctl dump-flows s3x1

There is one flow table. I know because the only entries show table=0. The difference between flow tables and flow entries is that a switch shows flow entries and specify which flow table it is with the table flag.

* + 1. Provide this same information by using the ONOS GUI.

iii. Provide a screenshot of the flow tables and flow entries from Switch3 from the GUI. **[10 points]**

A screenshot of a computer

Description automatically generated

1. View the ONOS topology.
   1. Shutdown the link from Switch1 to Host1 (from within Mininet).
      1. What command did you use? **[5 points]**

link h1x1 s1x1 down

* + 1. Did the ONOS GUI topology change in real-time or did you have to refresh? Provide two examples of how this Mininet command could be useful for testing. **[10 points]**

It changed in real time.

* To test host to host connectivity
* To check if flows entries are being updated based on delete timer
  1. Bring the link back up from Switch1 to Host 1.
     1. What command did you use? **[2 points]**

Link h1x1 s1x1 up

1. Navigate around the ONOS GUI.
   1. Go to the main ONOS Platform menu.
      1. Then navigate to Applications.
      2. Explain how you could activate additional applications and provide a screenshot as an example. **[10 points]**

simply click on the application and click activate in the top right corner.

A screenshot of a computer

Description automatically generated

* 1. Intentions:
     1. What are intentions in ONOS? **[5 points]**

Intentions in ONOS are high level network policies that can be applied to a SDN without specifically creating flows. Like I want h1 to accept http traffic only and by saying that it can be abstracted to the user.

* + 1. Create a Host to Host flow between two hosts in the topology.
       1. Generate traffic between these two hosts.
       2. Provide a screenshot of the flow as well as the green link indicating the bandwidth on that flow in the GUI. **[15 points]**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* + - 1. If you are having trouble with this based on this topology, you can ping from a host to another host in Mininet and then “show related traffic” in the GUI after clicking on the host you initiated the ping from. Then you can use the “a” shortcut to show all traffic flows with bandwidth.
      2. Based on the knowledge you have thus far, provide three functionalities/readability you can do in the ONOS controller that you can’t do in ODL. **[10 points]**
* Live Updates
* Intents
* Live port stats
  + - 1. With your experience thus far, do you like the ONOS or ODL controller better? Why? **[5 points]**

ONOS is WAY better than ODL given that base UI is so much more useful and provides live updates

# Objective 3 - Connect Mininet to Ryu

1. Refer the Lab 0 document to initialize Ryu.
2. Enable the Ryu web interface with a basic L2 switch (use the Lab 0 document for assistance).
3. Login to the Ryu web interface.
4. Login to Mininet.
   1. Start a new Mininet topology that connects to a remote controller (Ryu), uses easy to read MAC addresses, and uses a network topology that is a tree with a fanout of three and depth of three.
   2. Issue the “pingall” command in Mininet to bring up all the devices.
      1. What messages do you see on the Ryu CLI debug? Indicate why you see these messages. **[5 points]**

I see ofp\_events. I see these messages because flows are being added to the switches and the –verbose option is active showing what is happening within the network.

* 1. Within the GUI of Ryu, provide a screenshot of the topology. **[10 points]**

A network diagram of blue objects connected to white lines

Description automatically generated

# Objective 4 - Connect Mininet to Floodlight

1. Refer to the Lab 0 document to initialize Floodlight.
2. Access the Floodlight Web UI.
3. Open a new terminal and get into the Floodlight directory. Now start Mininet, and build a tree topology with a depth of 3, a fanout of 3, using OpenFlow version 1.3, and port number 6653:
   1. What command did you use? **[5 points]**

sudo mn --topo=tree,3,3 --mac --controller=remote,ip=10.224.78.231,port=6653 --switch ovsk,protocols=OpenFlow13

* 1. Paste the screenshot of the topology created on the Floodlight controller dashboard. **[5 points]**

A computer network diagram with many computers connected

Description automatically generated

A screenshot of a computer

Description automatically generated

1. After the topology is created on your Floodlight Web UI, proceed to evaluate the ‘Switch Details’ on any switch you desire. Post a screenshot of the switch details screen. **[10 points]**

A screenshot of a computer

Description automatically generated

1. Confirm you have issued a ‘pingall’ operation on Mininet for the topology created and evaluate the flow table entries. Post a screenshot of the updated flow table entries. **[10 points]**

A screenshot of a computer

Description automatically generated

1. Initiate any topology from Mininet and create a firewall rule of your choice. Post the screenshot of the firewall setting activated on the GUI. Also post screenshots of the Floodlight console logs and Mininet pings to prove that the firewall setting is working. **[10 points]**

A screenshot of a computer

Description automatically generated

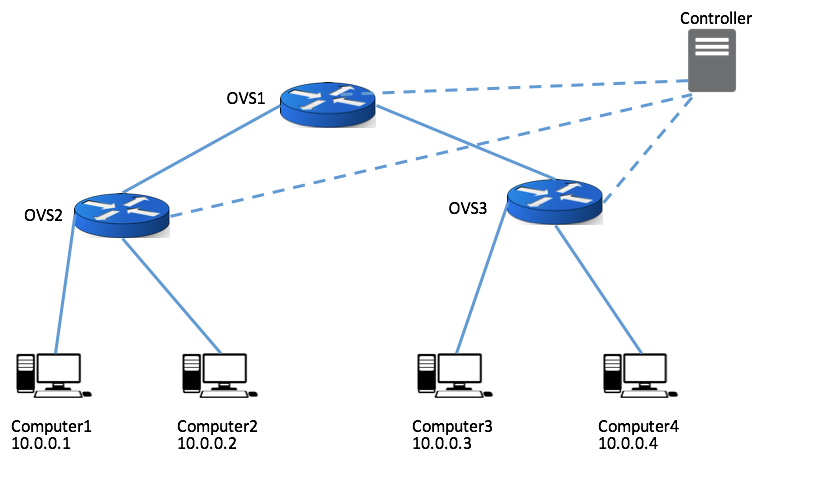
A computer screen with white text

Description automatically generated

A screenshot of a computer

Description automatically generated

# Objective 5 - Import Custom Topology in Mininet



**Figure 1.** Mininet custom topology

In this objective, you are required to build a custom topology for Mininet as shown in Figure 1 above. All links marked with solid lines need to be set to 10Mbps.

1. Start the controller of your choice, write down the IP of the controller.
2. Build the topology in MiniEdit in your Mininet VM.
   1. Create the topology, set device names, and link speed. Set controller to remote and specify the IP address and port number.
   2. Provide a screenshot of the topology. **[10 points]**

A computer network diagram with blue and red lines

Description automatically generated

1. Select “Export Level 2 Script” in the File menu, save the file as .sh file.
2. In your Mininet VM run the following command:

chmod +x [your .sh file]

Explain what this command does. **[2 points]**

It makes the file executable

1. Run the following command to start Mininet
   1. sudo ./[your .sh file]
   2. pingall
   3. Provide a screenshot of the command and Mininet running with results of the pingall **[10 points]**

A screenshot of a computer program

Description automatically generated

1. Now you loaded Mininet with a custom topology using a .sh file. There is another way to load a custom topology, by using the following command:
   1. sudo mn --custom topology.py --topo=mytopo --controller=remote,ip=xx.xx.xx.xx
   2. pingall
2. Modify the .sh file into .py file that can be loaded in the above command. More details can be found here <http://mininet.org/walkthrough/#custom-topologies> and here <http://mininet.org/api/classmininet_1_1topo_1_1Topo.html>.
   1. Provide a screenshot of the command and Mininet running with results of the “pingall” command. **[10 points]**

A screenshot of a computer

Description automatically generated

1. Provide original .sh file, modified .sh file, and the .py file with your report to get points.

# Objective 6 - Real World

1. Use any controller you like.
2. Create a huge topology of a multitude of switches or hosts or both (be careful not to crash your PC!).
   1. Provide a screenshot of the nightmare topology from the controller GUI. **[1 points]**

A screenshot of a computer

Description automatically generated

b. Explain why this topology could possibly be realistic of a real world SDN network design. **[5 points]**

**This topology could be used in a real world SDN because we might a campus with different buildings and each building has an access switch. Hosts connect to access and the access switches send data to the core switch in the middle.**

# Objective 7 – Report questions

1. After completing this lab, which controller would you prefer and why? [**5 points**]

I would prefer ONOS for its live updates, ability to create flows easily, and how easy it is to add applications within the GUI.

1. How can we make a controller communicate with a traditional router/switch not running OpenFlow for configuration/monitoring purposes? [**5 points**]

If there is a flow on a switch on a specific port to send specific traffic like SNMP to the controller, then we can monitor it. All we need is that pathway to the controller and back, and it can work.

# Total Score \_\_\_\_\_\_\_\_\_\_\_\_\_ / 320