Advanced Networking 2020

Lab #5: OpenFlow
Total points: 12 pts
Assignment
Lab date: March 10 2020
Submission date: 2:59PM March 13 2020 CET

Author: Jamila Alsayed Kassem, based on earlier works of mininet GitHub tutorial, Sara Shakeri and Peter Prjevara
Lab Teachers: Jamila Alsayed Kassem, Sara Shakeri, Vincent Breider and Peter Prjevara
Email: j.alsayedkassem@uva.nl, s.shakeri@uva.nl, peter@os3.nl, vincent@os3.nl
University of Amsterdam

Abstract

In this assignment you will get some hands-on experience with OpenFlow. You will learn about the communication between an OpenFlow switch (simulated with mininet) and two OpenFlow controller types. First, you'll set up a network topology, add a controller, and then see it behave under different scenarios (static entries, dynamic, packet redirection, and firewalling).

Preparation

1. Ask a colleague to team up with you.
2. Register your group on Canvas. Remember your group number.
3. You will be working with mininet, a network emulation tool, to create an Openflow network
4. You can download the vagrant box snapshot from `http://software.os3.nl/AdvancedNetworking/an2020-lab5.tar.gz` which has the mininet working environment already pre-installed
5. Vagrant environment should already be installed on your machines and loaded into kernel modules from lab 1. If not:

   ```
   sudo apt install vagrant virtualbox wget
   sudo modprobe vboxdrv vboxnetadp vboxnetflt vboxpci
   ```

6. Unzip the downloaded tar file and navigate into the folder that contains lab5.box
7. `vagrant box add lab5.box --name lab5`
8. vagrant up vagrant ssh

There is an example topology in the vagrant image built for you to try and play around. Please spend some time to get comfortable with the configuration commands of mininet and try to understand how it works.

- `sudo mn` #This will simulate a basic network
- `> nodes` #outputs all nodes in the network
- `> <node name> ifconfig`
- To get information on nodes `> dump`
- To test connectivity `> pingall || <node name> ping -c3 <node name>` where 3 is the number of packets
- For further information `https://github.com/mininet/mininet/wiki/Documentation`

When preparing the report make sure you do the following: For all tasks where relevant, show the commands used to insert the flows and the relevant part of the flow table of the switch.

Task 0: Network Setup

We will virtually build an Openflow network topology that consists of 3 hosts, switch, and a controller. The following figure 1 is an illustration of that. As you can see the controller will be running on the loopback interface.

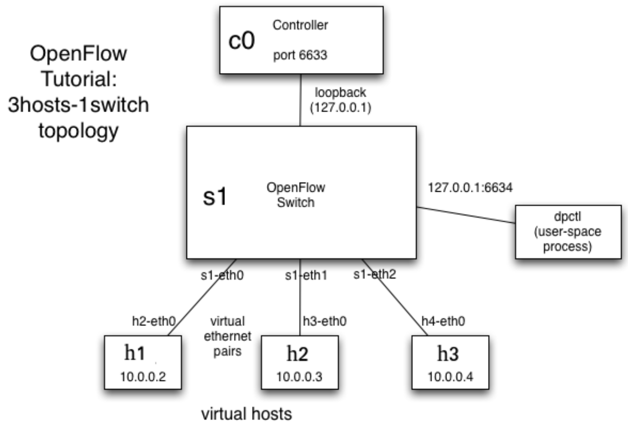

Figure 1: The network topology. *Source:* mininet GitHub tutorial

Task 1: Static flows (2 points)

In this task you will create the network topology illustrated in figure 1. To do that run the follow command.

```
sudo mn --topo single,3 --mac --switch ovsk --controller remote
```

At this point the nodes are unreachable due to the empty flow table.

1. Please explain what a flow table is?
2. Add entries to the flow table manually to make nodes reachable.

   Tip: ovs-ofctl is a utility that comes with Open vSwitch and enables visibility and control over a single switch's flow table. You will have to run the ovs-ofctl command in seperate terminal outside the mininet console
3. What type of flow instantiation was that? Discuss briefly.

Task 2: Reactive flow and Packet Capture (4 points)

Exit and clear the mininet console for this exercise.

You will now use a python based controller to simulate the behaviour of a reactive flow learning. You will edit the python code to make it behave as L2 learning switch module. We will use POX; which is a Python-based SDN controller platform geared towards research and education. Run the following to download the tools.

- `$ git clone http://github.com/noxrepo/pox`.
- `$ cd pox`.
- `git checkout betta`.
- `$ ./pox.py log.level --DEBUG misc.of_tutorial`; this runs the basic hub functionality.
- Run again the mininet console on a separate terminal to see the network connected to the POX controller .

1. Ping, capture packets on all interfaces using a tool of your choosing, then illustrate the traffic with a sequence diagram between h1, h2, h3, s1, and the controller.
2. Modify the python code pox/misc/of_tutorial.py to act_like_switch().
3. Ping again, capture packets, and explain main differences and why they occur.

Task 3: Using an OpenFlow controller

In the following tasks, you'll be using a different Openflow controller, floodlight. For more information, you can check the documentation provided on http://www.projectfloodlight.org/floodlight. The controller is already deployed on the Vbox provided.

- In the vagrant box, navigate to the floodlight directory and run the command `ant`.
- Then, start floodlight using `java -jar target/floodlight.jar`.
- If make fails with unicode errors set `export JAVA_TOOL_OPTIONS=-Dfile.encoding=UTF8` and then run `make clean; make`.
- After having the floodlight controller running on one terminal, you can connect it to a certain switch by running on a different terminal a mininet command that can simulate the topology with the Openflow switch.
- Connect to the current mininet network and retrieve all available information *Tip:* Create a different, potentially more suitable topology of hosts/network on mininet.
- Figure out how to use the Floodlight REST API to create new flows to do the coming tasks.

Task 3.1: Packet Redirection(3 points)

In this exercise you will create packet redirection using the floodlight openflow controller.

Send data from host 1 to host 2 and then set the Flow table rules in the openflow switch in such a way that:

- The packets which are sending to host 2 redirecte to host 3, and
- This redirection should be totally hidden from host 1, it should not have any idea that it is not talking to host 2.

1. Verify your configuration.
2. Copy your configuration settings to your submission document before moving on to the next exercise and explain the settings.

Tip: You can use the floodlight API to complete this task

Task 3.2: Traffic firewalling (3 points)

Here we are going to set access control on the traffic.

1. Create the following scenario using firewall module/API. Show all of the required configurations.
   - host 3 can SSH to host 1.
   - host 3 can ping host 1.
   - host 3 can perform HTTPS requests to host 2.
   - host 2 can send UDP traffic to host 3.
2. For each rule, test your configuration and include the output of your tests in the report.
3. Copy your configuration settings to your submission document and explain the settings.

Submission

Every group of two should submit the following file:

- **report** (PDF format) that describes in detail steps performed and answers.

  `lab5-report-\$groupnumber.pdf`

Note: it is sufficient that one of the two group members submits in Canvas!

Any other kind of submission will not be taken into account.