Aum Amriteswaryai Namah

LAB Assignment 3

For each question submit the final screenshot.

Q1. In this exercise, you need to use miniedit to create a Network Topology.

Invoke miniedit using the miniedit.py in the examples folder under mininet installation directory.host

1. You add two controllers by selecting the control icon. Give the controllers port as 6653 and 6654.
2. Then add two OVS switches s1 and s2 by selecting appropriate icon.
3. Add two hosts h1 and h2. Then using Link connect h1 with s1 and h2 with s2.
4. Right click on Switches and hosts and give them unique IP addresses.
5. Connect a link between s1 and s2 and both of the switch should be connected to a controller.
6. After building the topology, you may select the following option “Export Level 2 script”
7. You may get more help from the following site.

https://www.brianlinkletter.com/2015/04/how-to-use-miniedit-mininets-graphical-user-interface/

1. The saved python script should be modified by adding/modifying the following lines for Ubuntu20.04 users:
   1. from mininet.node import Controller, OVSBridge
   2. net = Mininet( controller=Controller, switch=OVSBridge)
2. Run the pythin script to have a topology with two controllers.
3. h1 should be able to ping h2

Q2. In the next experiment, we use two Routers to build a topology with three subnets

Python API given in <http://mininet.org/api/annotated.html> to create the topology.

1. You need to create two routers with two interfaces. Make sure they have IP address belonging to two different subnets:
   1. E.g. Router1 can have 192.68.1.1/24, 10.0.0.3/8, and Router2 can have 10.0.0.2/8, 172.16.0.1/16
   2. Router should use static IP routing (It is specified as an option in addNode)
2. Add one switch connected to each of the router interfaces
3. Add one host connected to each switch. Make sure that host belongs to the same Subnet as the router’s interface.
4. Router1 eth2 and Router2 eth1 should be connected. Router1 eth1 should be connected to a host h1 and Router2 should be connected to Switch s2 and s3 which in turn should be connected to hosts h2 and h3.
5. H1 (h1) should have default gateway as Router1 and h2 and h3 should have default gateway as Router2
6. Use the skeleton given in myrouter2.py and add more statements required for the steps given above.
7. Start the network simulation.
8. H1, h2, h3 should be able to ping each other.

Q3. In this question, you will install Pox controller unless it’s already installed on your system.

1. Install pox following the instructions in <https://noxrepo.github.io/pox-doc/html/>
2. Then run a L2 learning switch component by running the following command in the directory where pox.py is found. This component makes OpenFlow switches act as a type of L2 learning switch.

./pox.py samples.pretty\_log forwarding.l2\_learning

1. In another window, run mininet with any sample topology and with the option remote controller as pox.

sudo mn –topo=single,4 --controller pox

1. Invoke Wireshark in another window or inside mininet and start capturing the packets.
2. Do a ping between two hosts. Once you find ping is working, Stop capturing the packets in Wireshark.
3. Now locate the following Openflow messages
   1. Packet in
   2. Echo,
   3. Hello and Barrier.