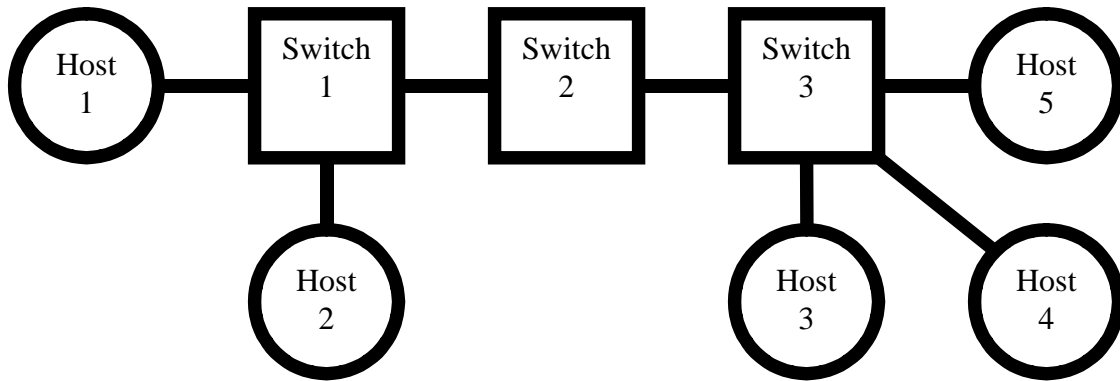


Homework #1: Networking with Mininet/NOX

The purpose of this assignment is to give you hands-on experience in developing a virtual network that can run customized software.

Setup

1. Download the tutorial and appropriate files.
2. Follow the tutorial up to developing a network topology and create the following topology in Mininet. Verify that the topology is correct by analyzing the Wireshark output.



Fun with NOX

In steps 1 and 2, you are going to be direct all network traffic to the controller that will then decide how each packet should be directed. In steps 3 and 4, you will be installing microflow rules and wildcard rules, respectively, so that the majority of network traffic will be handled by the switch as opposed to the controller.

Attention: Include the python code for all four parts in the write-up.

1. Hub Controller

All traffic arriving at the switches will be directed to the NOX controller. We will begin with a simple Ethernet hub where all network traffic is **flooded on all ports** except the port that it arrived on. This is the default behavior of the controller code that we provide in `COS561Test.py`. Analyze what occurs on the network topology above with the Ethernet hub controller.

- a. Have *H1 ping -c 100 H2*. How long did it take to ping? Which of the hosts and switches observe traffic as a result of the ping? Have *H1 ping -c 100 H5*. How long does this take? Is there a difference?
- b. Perform a *pingall* and copy the output, verifying that all hosts are pingable.

2. MAC Learning Controller

Once again, all traffic arriving at the switches will be directed to a NOX controller. The NOX controller will perform **MAC learning**. The basic algorithm is provided in `COS561Test.py`. Analyze network behavior with the MAC learning controller.

- a. Have *H1 ping -c 100 H2*. How much time does it take to ping the first packet? Is there any difference with the Hub Controller?
- b. Have *H1 ping -c 100 H5*. How long does this take? Have *H1 ping -c 100 H4*. Is there any difference? Why or why not?
- c. Perform a *pingall* and dump the output. Verify that all hosts are reachable.

3. MAC Learning Switch via MicroFlow Rules

Modify the MAC Learning Controller to **install microflow rules** in the Openflow switch reactively to incoming traffic. Thus, the controller will only receive packets for which there are no matching microflow rules in the switch.

- a. Have *H1 ping -c 100 H2*. How does it compare to the learning switch and hub from earlier?
- b. Perform a *pingall* and dump the output. Verify that all hosts are pingable. Also dump the microflow rules that are installed in each switch.

4. IP Router via Wildcard Flow Rules

Place H1 and H2 in a different IP subnet from H3, H4, and H5. Install IP matching wildcard flow rules in Switch 2 so that it **behaves like a router**. Switch 1 and Switch 3 should still behave as reactive MAC learning switches from above (3).

- a. Perform a *pingall* and dump the output. Dump the wildcard rules of each switch.
- b. How does this network compare to the previous controllers? Which one is better?