Spring 2016: EE 555 Project (Due April 29, 11:59am, on D2L)

1 Open Flow Protocol

OpenFlow is a communications protocol that gives access to the forwarding plane of a network switch or router over the network [1]. It is used to govern the communication between a controller and the switch in a software defined network (SDN) environment.

2 Project Description

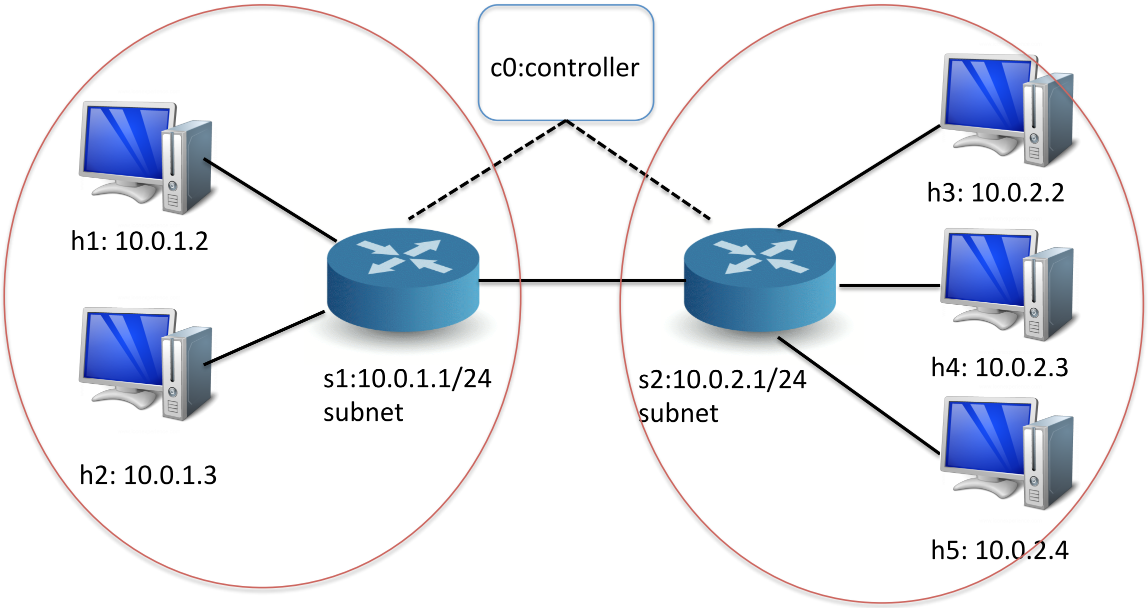
The project consists of two parts.

The tutorial in https://github.com/mininet/openflow-tutorial/wiki provides step-by-step instructions on implementing open flow applications in your own laptops. This is assisted by software MININET, which is able to simulate a realistic virtual network on a single machine [2]. This tutorial provides instructions on multiple open flow controller platforms. In this project, you are required to use POX (a Python-based SDN controller platform).

The first part of this project requires you to do all the parts until **and including** the Router Exercise part in the tutorial given in the above link. (Skip the parts on controller platforms other than POX and the section Control a Slice of a real Network.)

The second part of this project requires you to repeat the Router Exercise part under the topology given in the picture. Make sure your implementation satisfies the following requirements. You should test every item before the submission and record the results in your report. You will also be required to demonstrate this part to your assigned TA.

1. Attempts to send from a host to an unknown address range should yield an ICMP destination unreachable message.
2. Packets sent to hosts on a known address range should have their MAC dst field changed to that of the next-hop router.
3. The router should be pingable, and should generate an ICMP echo reply in response to an ICMP echo request.
4. All hosts must be connected to each other. This can be verified using 'pingall'.
5. Flow Mods isn't necessary. But you get bonus points if it works for both parts 1 and 2.



3 Submission Guideline

- Each student should submit a single zip file on D2L.

- The zip file must contain two folders, 'Part 1' and 'Part 2' that contain your code for parts 1 and 2 of the project.

- Each folder must contain a README file (ASCII only. No MS-Word/PDF) that contains a detailed description of your code files and instructions on running them.

- Submit a report (PDF) that describes the steps taken in completing this project. Include implementation details and/or problems faced, but DO NOT includes the entire code (snippets are allowed) in the report.

4 Reference

1. Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. 2008. OpenFlow: enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.* 38, 2 (March 2008), 69-74.

2. Nikhil Handigol, Brandon Heller, Vimal Jeyakumar, Bob Lantz, and Nick McKeown. Reproducible Network Experiments using Container-Based Emulation. CoNEXT 2012, December 10-13, 2012, Nice, France.