Spring 2017: EE 555 Project (Due April 30, 11:59pm, on D2L)

OpenFlow 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.

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 openflow 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 openflow controller platforms. In this project, you are required to use POX (a Python-based SDN controller platform).

Part 1

Part 1 is worth 60 points.

The first part of this project requires you to do all the parts **until and including** the Router Exercise section, 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.

Part 2

Part 2 is worth 40 points.

The second part of this project requires you to

1. complete the Advanced Topology section of the tutorial. Make sure your implementation satisfies the following requirements:
    a. Attempts to send from a host to an unknown address range should yield an ICMP destination unreachable message.
    b. Packets sent to hosts on a known address range should have their MAC dst field changed to that of the next-hop router.
    c. The router should be pingable, and should generate an ICMP echo reply in response to an ICMP echo request.
    d. All hosts must be connected to each other. This can be verified using 'pingall'.

2. complete the Create Firewall section of the tutorial.

Bonus

This bonus section is worth 20 points.

You are *not required* to complete this bonus section but, as the name suggests, you'll receive bonus points if you do.

Repeat the Advanced Topology section of the tutorial, but this time using a topology of your own. Specifically, you need to use a topology that satisfies the following requirements:

- Your topology has at least 3 routers.
- Each router has at least 1 host attached.
- You have at least 4 hosts in total.

Submission Guidelines

Make sure that you **strictly follow** the guidelines given below, including the instructions on folder names and file formats. Any deviation from the guidelines given below is likely to incur **heavy penalties**. Given that this is a lousy way to lose points, kindly save yourself (and us!) the trouble.

- Complete this project in groups of 2 or 3.
  - o You can choose to complete this project individually only if you are a DEN student and are not able to find a partner.
- Each group should submit a single zip file on D2L. Note that you **will be penalized** if multiple members of the group submit. Designate one member (it doesn't matter who) to upload the zip file.
- The zip file must contain 3 folders,
  - o `part_1`
  - o `part_2`
  - o `bonus`

  that contain your code for Part 1, Part 2 and the Bonus section respectively. You can choose to omit the `bonus` folder if you don't want to attempt the Bonus section.
- 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 only**. **No MS-Word**) 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. The report must clearly indicate the names of the group members.

You should test every item before the submission and record the results in your report. Please be prepared to demonstrate your working project.

References

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.