TLEN 5830 – Next Generation Networks

Lab 9 – SDN Hardware and Apps

University of Colorado Boulder

Interdisciplinary Telecom Program

Professor Levi Perigo, Ph.D.

# Lab Summary

The final lab is intended to incorporate all the theory and practice learned throughout the course into a summarization of skills in a hands-on, real-world examination. This lab will focus on thinking outside of the box, and will force the students to implement technologies learned throughout the course in a virtual environment, and apply them in hardware. Students will utilize the skills learned throughout this course such as:

* setting up and installing an SDN controller on an x86 platform
* configuring Open vSwitch on bare metal boxes
* understanding the CLI of prominent vendors that support OpenFlow
* basic routing, switching, and network engineering design
* troubleshooting OpenFlow and network design
* programming applications via API and OpenFlow

This lab combines the conceptual discussions and theory about SDN with a deeper dive into how SDN applications are implemented in the real world. In this lab, you will create an SDN application that will communicate via the NBI. This will allow the traffic to be manipulated in physical hardware based on an application you wrote.

# Objective 1 – Configure Internet Router

1. Connect 1 port of your Cisco router to the switch labelled ‘NGN - Internet’ and configure it as a DHCP client. It should receive an IP in the 10.0.0.0/8 subnet.
2. Configure the other port on the router to act as a DHCP server, it will be used for one of the applications, and for interVLAN routing.

# Objective 2 – Network Design

1. Physically cable the devices according to the network diagram.
2. Configure the Management network with a VLAN carrying only the traffic meant for your devices. (Doing this will enable a single switch to be used for multiple groups).

# Objective 3 – Configure OpenFlow Switches (ABMX/HP servers/Pica8)

1. Load Open vSwitch on the hardware (if not already installed).
2. Configure interfaces in proper IP subnet / VLAN for your design:
   1. Apply ports in OpenFlow network to use OpenFlow.
   2. Configure static route for non-OpenFlow / out-of-band network.

# Objective 4 – Configure Vendor Hybrid Switches (Arista/Dell/HP)

1. Follow Arista/HP/Dell documentation instructions to activate OpenFlow ports on the hardware.
2. Configure Management network for non-OpenFlow network.
3. Configure the OpenFlow ports and connect the switch to the controller
4. Configure OpenFlow flows on Arista/HP/Dell switches using a Controller of your choice.

# Objective 5 – Install and Configure the SDN Controller

1. Load the controller of your choice on the HP server.
2. Configure the controller’s IP address to be in the non-OpenFlow network upon connecting it to the Management network.
3. Ensure that Controller has Internet Connectivity (Either connect it to the Internet Router or configure Internet through other means).

# Objective 6 – Application Programming

I. DNS Blacklist

1. Create a program that will force DNS traffic, from any network device, to the controller, which will route the traffic to the “checker” website’s API to determine if the website is “good / bad.” Depending on the results returned from the website, the switches should “block/deny” traffic to the “bad” website and “allow/forward” traffic to the “good” website. OpenFlow table flow entries can be used, but a simple drop/forward is sufficient.
2. Three websites will be provided to accompany this lab:

* **Site-Checker**: <http://checksite.herokuapp.com>
* **Good-Site**: <http://goodweb.herokuapp.com>
* **Bad-Site**: <http://badweb.herokuapp.com>

Any other website passed through this site checker app will not work properly. You can use a DNS checker of your choice as long as the DNS site checking requests get initiated from the Controller.

1. **Bonus implementation**: If the requested URL is bad, instead of doing nothing, the PC will receive a modified DNS reply that will let the browser display contents of “warning" or “blocked’ website.

II. DHCP application

1. Create a program that will force DHCP requests originated from hosts to the Internet router (DHCP server).
2. Ensure that the DHCP requests forwarded from the switches to the server is unicast and not broadcast.

III. SDN trace application

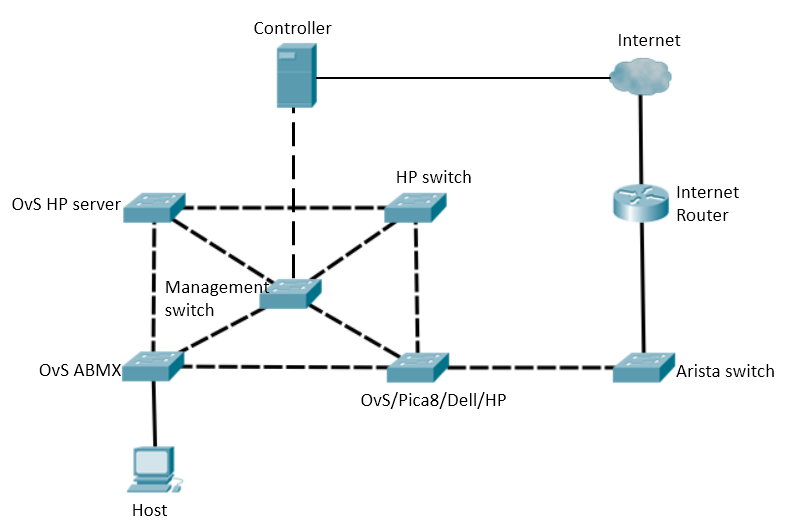
1. The objective of this application is to trace the path in an SDN OpenFlow network.
2. The application should take source and destination inputs from the user via either the CLI or GUI (Flask).
3. Using OpenFlow messages, the application traces the path (i.e. prints out the DPID’s of the OpenFlow switches in the path).
4. **Bonus implementation**: the application displays ingress/egress ports along the path

# Objective 7 – Final Testing & Deliverables

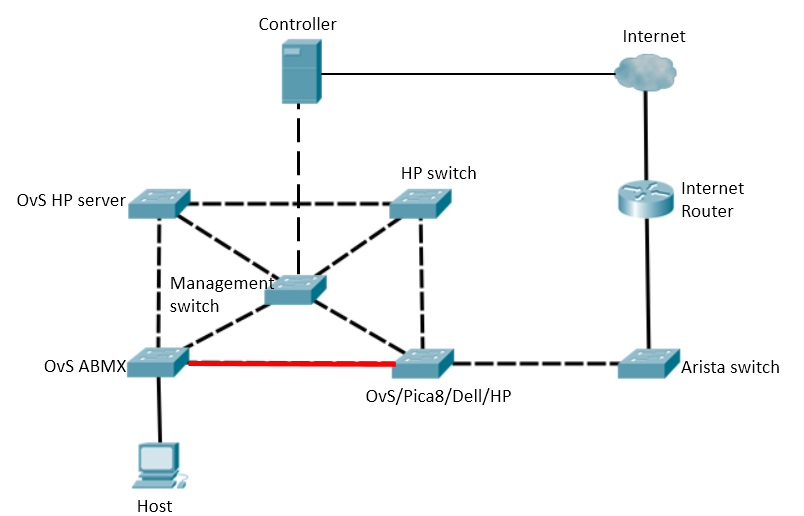
1. Demonstrate functionality to TAs and Professor.
2. Answer interview questions individually about final lab.
3. Team submission:
   1. Application code
   2. Lab document: relevant hardware configurations, explanation of application
4. Each student submits:  
   What each person (including yourself) on the team contributed.  
   Grade each other (including yourself) for contributions.

# Network Diagrams and Completion

1. Network diagram:



1. The network should be able to converge even when the cable marked “Red” goes down:



1. Setting up the network hardware according to the diagram. [**Receive 50% of grade**]
2. Implement the DNS application on hardware. [**Receive 20% of grade**]
3. Implement the DHCP application on hardware. [**Receive 10% of grade**]
4. Implement the SDN trace application on hardware. [**Receive 20 % of grade**]
5. Implement the applications via REST. [**Receive 20 % of grade**]