

Assignment 4

Understanding network emulation software, Mininet

Deadline: Nov 21, 2023

Total: 9 points

The purpose of this assignment is to give you hands-on experience in developing a custom virtual network using the mininet network emulation software. The mininet network supports software Openflow switches, Open vSwitch (a.k.a., OVS), and an OpenFlow SDN controller, POX, that will help us realize a virtual Software Defined Network.

The SDN paradigm separates the control plane (e.g., routing algorithms) from the data plane (e.g., forwarding). The control plane runs as a logical centralized program on commodity servers (instead of switches).

Setting up your machine:

1. Install virtual box (if not already installed)
 - Download and install the appropriate executable for [Windows](#)
 - Download the appropriate executable for [Ubuntu](#) and follow the instructions:
2. Download the VM from the following [link](#). This VM has mininet, OVS, and POX installation.

Warming up:

We must build certain fundamentals before starting to configure OVS and run the SDN controller over the mininet network emulation platform. Go through the videos mentioned for each topic.

1. Mininet utilizes process network namespaces in Linux which allows you to do lightweight emulation (like virtualization). Watch the following video (watch only until 13:55 mins): [Introduction to Linux Network Namespaces](#)
2. Watch this video for an introduction to mininet and its usage. You can run the commands in parallel on your VM for understanding. [Introduction to Mininet](#)
3. Watch this video for an introduction to Open vSwitch and its usage. You will understand packets such as packet In, packet Out, and flow mod communicated between the OVS and the controller. [Introduction to Open vSwitch](#)
4. This video facilitates understanding of running an SDN application over a mininet+OVS+POX environment, [Customizing SDN control](#) The slides for this video are available [here](#) (slide number 35 – 53)

Miscellaneous links:

1. If you want to start mininet installation from scratch on your machine (**not recommended** for this assignment). [Download/Get Started With Mininet](#)
 - a. If you want the VM with GUI do the following.

```
$sudo apt-get install xinit x11-xserver-utils lxde
$startx lxde
```

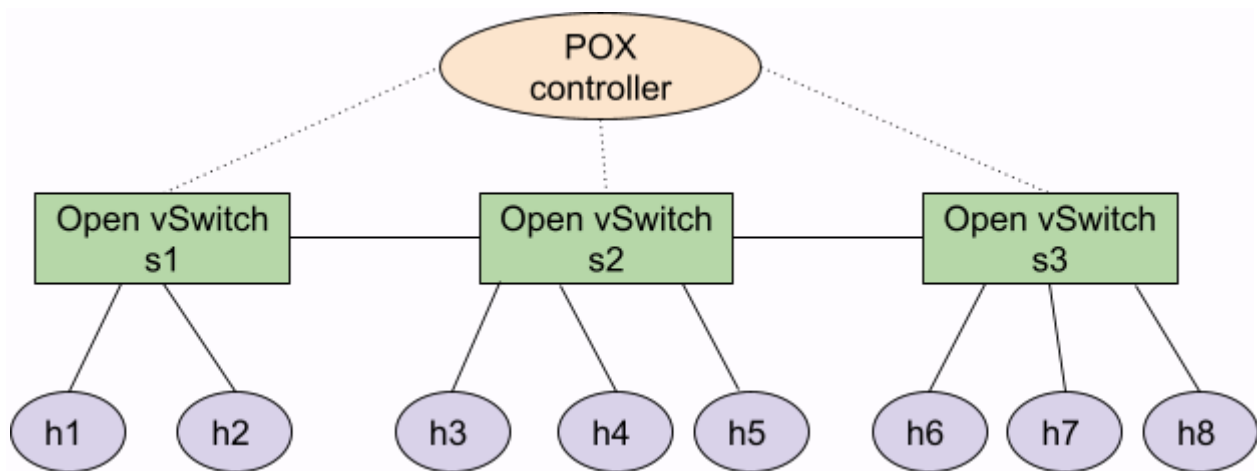
2. Mininet commands and usage with OVS and POX. [Mininet Walkthrough](#) [Mininet Home Page](#)

Q.1. Setting up the topology for the assignment:

- A. Follow the Mininet tutorial to create a custom network topology shown in the figure below. **(2 points)**
- Take hints from the example custom Mininet topology which is available at `~/mininet/custom/topo-2sw-2host.py`
 - You can run Mininet with this custom topology as follows.

```
$sudo mn --custom ~/mininet/custom/topo-2sw-2host.py --topo mytopo --mac --switch ovsk --controller remote
```
- B. Verify the topology by checking the network configuration (within the mininet prompt) as follows. **(1 point)**

```
mininet> net
```



Q.2. Refer to the tutorials 5 and 7 to answer this question.

- Create a bottleneck at the interface of **host h1** using **"tc"**. **(1 point)**
- Generate TCP traffic between the hosts **h1** and **h6** using **iperf**. **(1 point)**
- Use **Wireshark/tcpdump** at **host h1** to capture the packets generated in (b) **(1 point)**
- Plot the congestion window using the packet dump obtained from (c) **(3 points)**

What to submit:

Submit the following documents in one folder.

- A short write-up with snapshots/plots followed by an explanation of your observations
- Include the python code for the custom topology, scripts for Q.2, and the plot