Lab2: Play with Mininet

Mininet emulates a complete network of hosts, links, and switches on a single machine. https://github.com/mininet/mininet/wiki/Introduction-to-Mininet may be helpful if you would like to learn Mininet in more depth. Here, we provide a basic instruction.

Install Mininet and Open vSwitch:

In the terminal of your virtual machine, enter the following commands:

```
git clone https://github.com/mininet/mininet.git
mininet/util/install.sh -fnv
```

You may be temporarily unable to access the repository due to abnormal internet connection, like shown in the following screenshot. Check your connection and then re-run the command.

```
fatal: unable to access 'https://github.com/mininet/openflow/': Failed to connect to github.com
port 443: Connection timed out
```

After the installation has completed, test the basic Mininet functionality:

```
sudo mn --switch ovsbr --test pingall
```

A detailed introduction of getting started with Mininet could be found at Download/Get Started With Mininet - Mininet. And Mininet Walkthrough - Mininet provides a walkthrough that demonstrates most Mininet commands, as well as its typical usage in concert with the Wireshark dissector. Before you start dealing with problems in Lab 2, it is recommended to finish the above walkthrough.

Test a few Examples:

Mininet provides many examples in its source code. From these examples, you can learn how to create your own topologies and run your own tests. In this lab, it is recommended that you read, comprehend and test the following examples:

1. mininet/examples/linuxrouter.py
2. mininet/examples/linearbandwidth.py

3. mininet/examples/simpleperf.py

Use the following commands to test the examples above.

```
cd mininet/examples/
sudo python linuxrouter.py
```

```
help
pingall
nodes
```

Useful Commands:

```
ifconfig
sudo ovs-vsctl show
sudo ovs-ofctl dump-flows s1
```

You may also use OVS commands to set up your own flow rules:

```
sudo ovs-ofctl add-flow "s3" "in_port=3 actions=output:1,2"
```