Lab 2: Mininet: An Instant Virtual Network

# Objectives

* Introduce about basic python programming, debugging skills
* A brief introduction on Python networking programming
* Practice mininet with exercises

# Lab Tutorial

## Reference to go through

1. Introduction to Mininet <https://github.com/mininet/mininet/wiki/Introduction-to-Mininet>
2. Mininet Tutorial
3. Iperf: <https://iperf.fr/>
4. Google: A universal reference
   1. **Mininet Installation**

**Note: Please DO NOT follow the Mininet Tutorial to download the Mininet VM.**

1. Environments

You can choose from the following hypervisors:

**On Windows:**

**VirtualBox**

**VMware WorkStation**

**On Mac:**

**VirtualBox**

**VMware Fusion 7**

Recommended: install the VMware Tools/ VirtualBox Guest Additions to enhance your environment.

1. Install mininet on Ubuntu

**sudo apt-get update**

**sudo apt-get install mininet**

* 1. **Basic Mininet Demo**
     1. **Mininet CLI**

1. Creating a first mininet topology

**sudo mn --topo=single,2**

1. Typical mininet options

**--topo=TOPO** linear|minimal|reversed|single|tree[,param=value...]

**--mac** automatically set host MACs

**--test=TEST** cli|build|**pingall**|**pingpair**|iperf|all|iperfudp|none

1. Mininet CLI commands

<host> **ping** <host> ping from one host to another

**xterm** <host> Open up a terminal of a host

<host> **ifconfig** Check the interface configuration of a host

* + 1. **Customize network topology with mininet python script**

Create a python script named **topo-2sw-2host.py**:

1. """Custom topology example
3. Two directly connected switches plus a host for each switch:
5. host --- switch --- switch --- host
7. Adding the 'topos' dict with a key/value pair to generate our newly defined
8. topology enables one to pass in '--topo=mytopo' from the command line.
9. """
11. **from** mininet.topo **import** Topo
13. **class** MyTopo( Topo ):
14. "Simple topology example."
16. **def** \_\_init\_\_( self ):
17. "Create custom topo."
19. # Initialize topology
20. Topo.\_\_init\_\_( self )
22. # Add hosts and switches
23. leftHost = self.addHost( 'h1' )
24. rightHost = self.addHost( 'h2' )
25. leftSwitch = self.addSwitch( 's3' )
26. rightSwitch = self.addSwitch( 's4' )
28. # Add links
29. self.addLink( leftHost, leftSwitch )
30. self.addLink( leftSwitch, rightSwitch )
31. self.addLink( rightSwitch, rightHost )

34. topos = { 'mytopo': ( **lambda**: MyTopo() ) }

Run the script to create a customized topology

**sudo mn --custom ./topo-2sw-2host.py --topo mytopo --test pingall**

* + 1. **Iperf – a simple bandwidth measurement and traffic generation tool**

1. Creat a linear topology with two switches and two hosts

**sudo mn --topo=linear,2**

1. Measuring bandwidth between two virtual hosts

**Iperf -s** at the server side (IP address 10.0.0.1)

**Iperf -c** 10.0.0.1 at the client side (IP address 10.0.0.2)

1. Measuring the bandwidth between two virtual hosts

a. Create a single switch connecting mininet

* 1. **Exercise 1: Customized Network**

By customize the mininet script, you can customize a wide range of network properties. In this exercise, you are required to customize the topology as shown in Figure 1.



Figure 1. Customized Topology.

The following parameters to be customized as shown above:

1. Topology
2. Host IP address
3. Host MAC address
4. Link latency
5. Link bandwidth
6. Link packet loss rate

**Task 1 (To Submit in Lab Report)**

1. Students are required to submit the script to implement the above. The script should be named as “**customized\_topo.py**”
2. The TA should be able to build the topology and verify the connectivity using the following command:

**sudo mn --custom ./customized\_topo.py --topo mytopo --test pingall**

1. The TA should be able to verify the end-to-end bandwidth and delay using Iperf:

**H1 – H2: 10Mbps with ~12ms latency**

**H2 – H4: 18Mbps with ~22ms latency**

**H3 – H4: 10Mbps with ~12ms latency**

* 1. **Exercise 2: Running Program on Virtual Hosts**

****

Figure 2. Iperf on mininet virtual network

In this exercise, the students are required to run TCP flow generator Iperf on Hosts **automatically** after the topology is established.

As shown in Figure 2, H1 will generate a TCP flow to H3 with maximum rate from T=0sec (topology built) to T=20sec; H2 will generate a TCP flow to H4 from T=10sec to T=30sec.

The end-to-end bandwidth measurement can be obtained at the server side.

**Task 2: (To Submit in Lab Report)**

1. Students are required to submit the script to implement the above. The script should be named as “**host\_iperf.py**”
2. The student should submit the bandwidth measurements snapshot. They should contain the bandwidth measured every 0.5 second for Flow 1 and Flow 2. A sample result is should be in the similar format as follows,

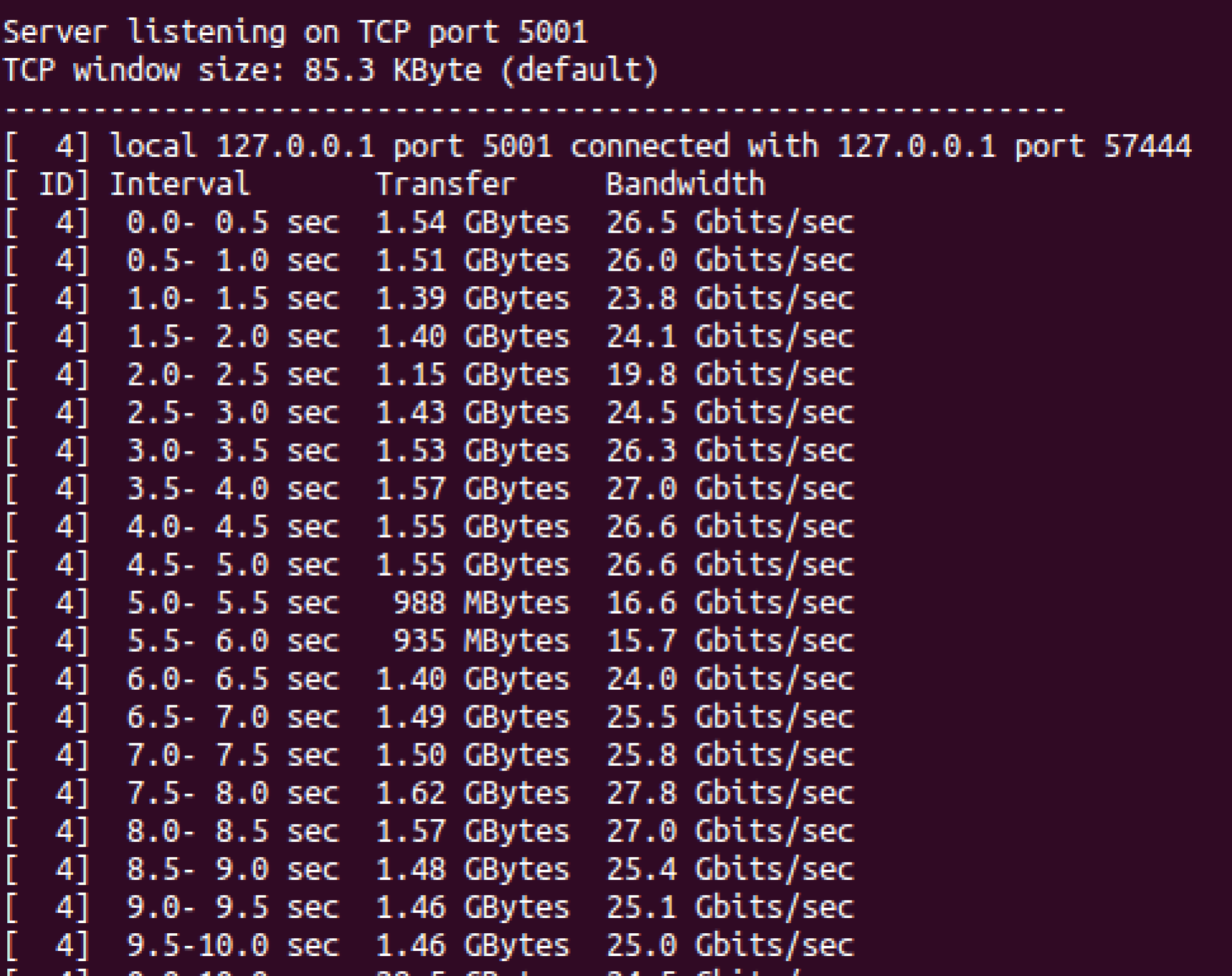


Figure 3. Sample Iperf bandwidth measurement result

**Submission Direction**

**Demo at the last session of Lab2.**

**Files submit as a Zip to New Class’s assignment before 12:00p.m. Oct. 15.**

**2 Script files: customized\_topo.py and host\_iperf.py**

**1 lab report consisting of the screenshot, and any explanations you want to include about your code.**