Computer Comm & Networks - ITCS 6166

Assignment - 2 Problem Statement **Due on: 02/26/2019 11:59pm**

Goal: Learn to build topologies, configure link parameters and troubleshooting tools.

Note: Please go through the reference document first before going through this document.

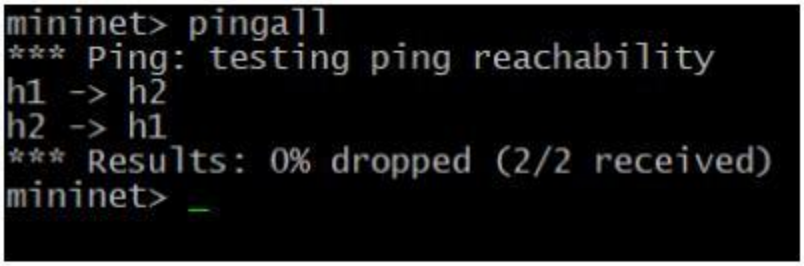
Build the custom topology and configure the link parameters for the topology in figure shown on the bottom of the document, using Mininet python API. For this assignment you must use OVS controller.
You should familiarize yourself with the following important classes, methods, functions and variables to create the network topology

- **Overview: [This is same as provided in the reference document]**

  You should familiarize yourself with the following important classes, methods, functions and variables to create the network topology
  1. Topo: the base class for Mininet topologies
  2. addSwitch(): adds a switch to a topology and returns the switch name
  3. addHost(): adds a host to a topology and returns the host name
  4. addLink(): adds a bidirectional link to a topology (and returns a link key, but this is not important). Links in Mininet are bidirectional unless noted otherwise.
  5. Mininet: main class to create and manage a network
  6. start(): starts your network
  7. pingAll(): tests connectivity by trying to have all nodes ping each other
  8. stop(): stops your network
  9. net.hosts: all the hosts in a network
  10. dumpNodeConnections(): dumps connections to/from a set of nodes.
  11. setLogLevel('info' | 'debug' | 'output'): set Mininet's default output level; 'info' is recommended as it provides useful information.

Verification:

1. Use pingall command in mininet terminal and ensure all hosts are reachable with 0% dropped.



```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet> _
```

2. Use iperf command to ensure the link parameters are enforced properly as in the given network topology
Iperf is a traffic generation / network performance measurement tool that can operate in client / server mode. The server mode operation will open a port and accept the incoming connection requests. Client mode operation will generate requests towards the Iperf server. We will be using this tool extensively for measuring the end-to-end network bandwidth.

Example: To measure the end-to-end bandwidth between h1 and h2, do the following

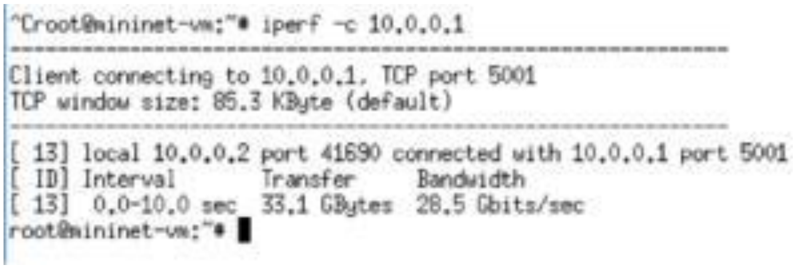
start iperf server on host h1 use the following command in terminal:

```
iperf -s
```

To start the iperf client on host h2 use the following command in terminal:

iperf -c 10.0.0.1

Now, you will should be able to see the end-to-end network bandwidth as shown in the below figure:



```
^Croot@mininet-vm:~# iperf -c 10.0.0.1
------------------------------------------------------------
Client connecting to 10.0.0.1, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[ 13] local 10.0.0.2 port 41690 connected with 10.0.0.1 port 5001
[ ID] Interval       Transfer     Bandwidth
[ 13]  0.0-10.0 sec  33.1 GBytes  28.5 Gbits/sec
root@mininet-vm:~# █
```

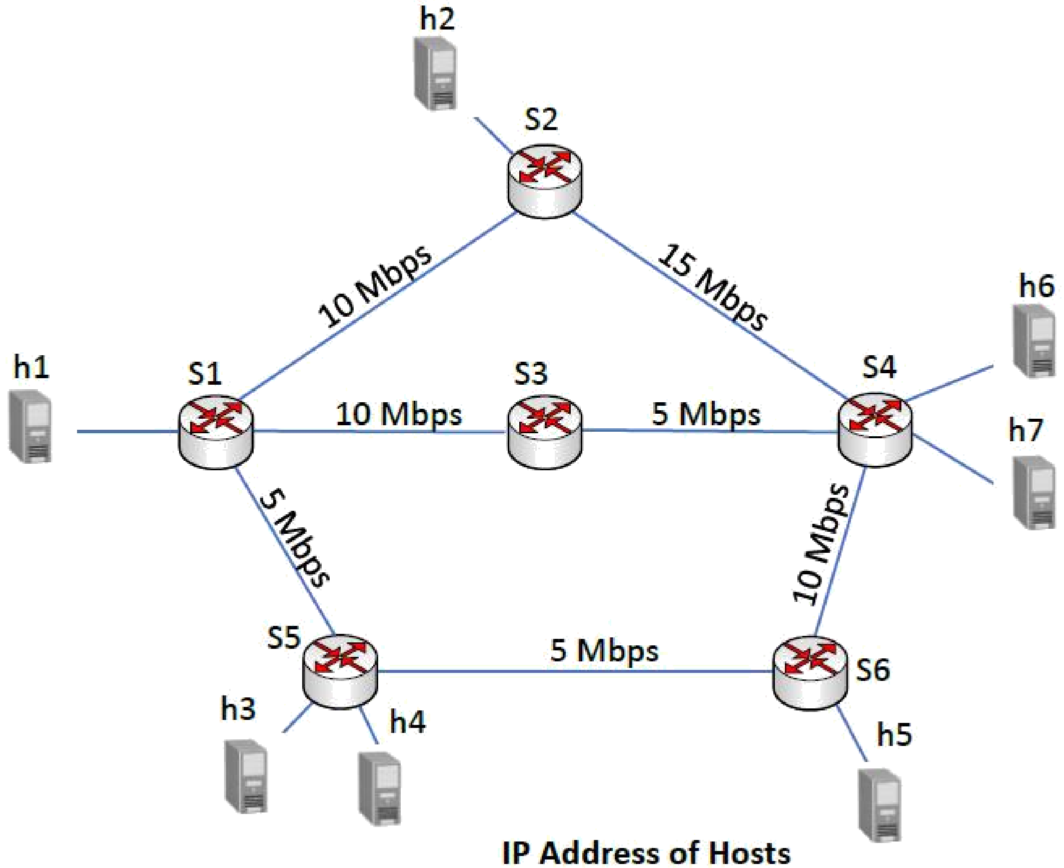
You should turn in your python script on CANVAS course site assignment section with filename named in the following format:
Please include the below things in your assignment submission:
1. Python script: Save file as topo_yourfirstname.py
2. Snapshot of pingall and iperf verification.
Upload the Compressed file as Assignment2_yourfirstname.zip

Note: Please follow the deadline strictly as we would not accept any submissions after the deadline.

Also in case of any question post your query in discussion portal.

References:

1. http://mininet.org/api/hierarchy.html
2. https://github.com/mininet/mininet/wiki/Documentation
3. https://www.tutorialspoint.com/python/



IP Address of Hosts

h1 – 10.0.0.1

h2 – 10.0.0.2

h3 – 10.0.0.3

h4 – 10.0.0.4

h5 – 10.0.0.5

h6 – 10.0.0.6

h7 – 10.0.0.7

Instructions:

- Understand the reference document and how the simple network topologies have been build using mininet, you must be able to build the above topology using your own logic.
- Not only focusing on the solution by achieving the expected output, you should equally focus on the efficiency and best practices [code comments] while implementing the solution.
- The most you would learn is from troubleshooting your own code. So, please spend time on troubleshooting the code.
- It is not expected to post simple code implementation errors/issues on the discussion board.