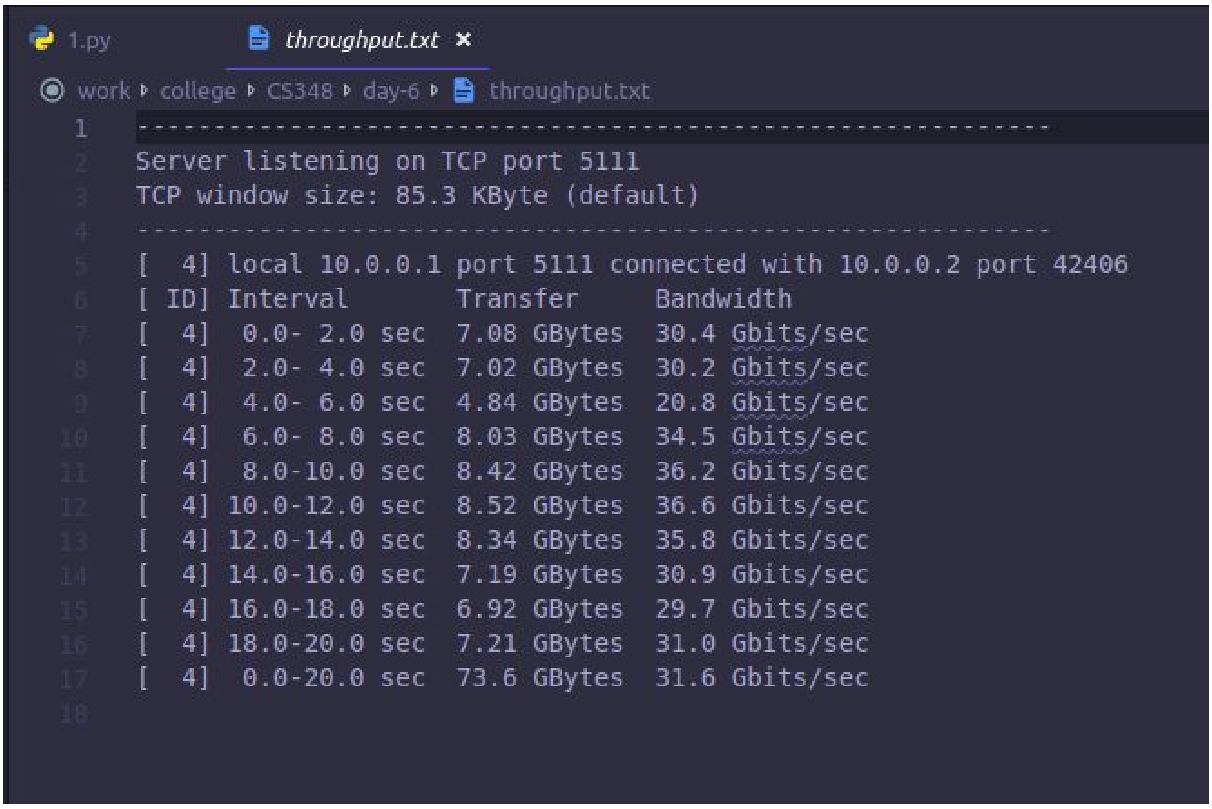
Assignment - 6
Name: Anurag
Roll no: 1601CS05

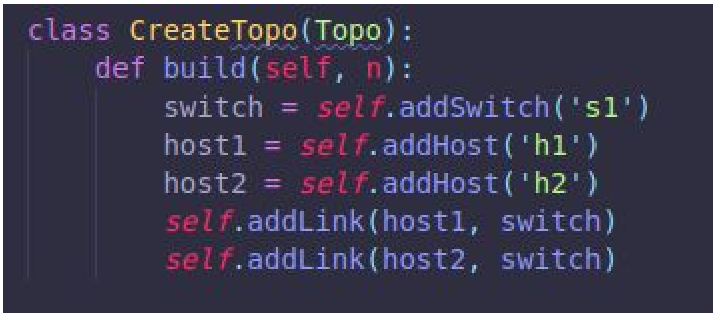
1.



```
------------------------------------------------------------
Server listening on TCP port 5111
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  4] local 10.0.0.1 port 5111 connected with 10.0.0.2 port 42406
[ ID] Interval       Transfer     Bandwidth
[  4]  0.0- 2.0 sec  7.08 GBytes  30.4 Gbits/sec
[  4]  2.0- 4.0 sec  7.02 GBytes  30.2 Gbits/sec
[  4]  4.0- 6.0 sec  4.84 GBytes  20.8 Gbits/sec
[  4]  6.0- 8.0 sec  8.03 GBytes  34.5 Gbits/sec
[  4]  8.0-10.0 sec  8.42 GBytes  36.2 Gbits/sec
[  4] 10.0-12.0 sec  8.52 GBytes  36.6 Gbits/sec
[  4] 12.0-14.0 sec  8.34 GBytes  35.8 Gbits/sec
[  4] 14.0-16.0 sec  7.19 GBytes  30.9 Gbits/sec
[  4] 16.0-18.0 sec  6.92 GBytes  29.7 Gbits/sec
[  4] 18.0-20.0 sec  7.21 GBytes  31.0 Gbits/sec
[  4]  0.0-20.0 sec  73.6 GBytes  31.6 Gbits/sec
```

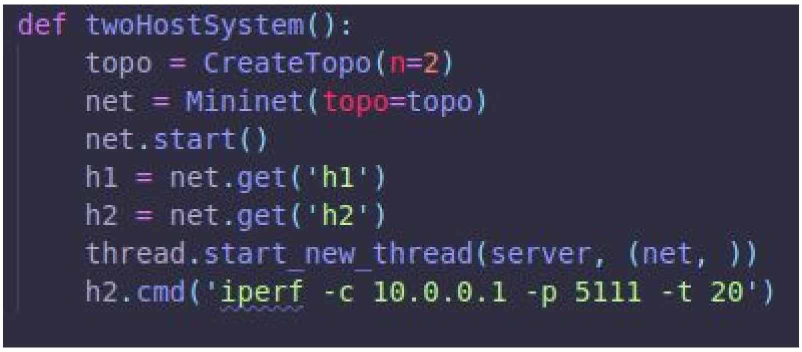
In this question i created a topology which comprises of two hosts (h1,h2)connected to a single switch. The pair of hosts will transfer TCP packets among each other using sockets.

This code create the topology.



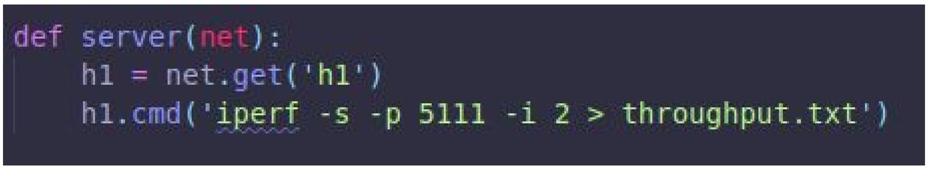
```python
class CreateTopo(Topo):
    def build(self, n):
        switch = self.addSwitch('s1')
        host1 = self.addHost('h1')
        host2 = self.addHost('h2')
        self.addLink(host1, switch)
        self.addLink(host2, switch)
```

The client host (h2) will send packets for 20 seconds.



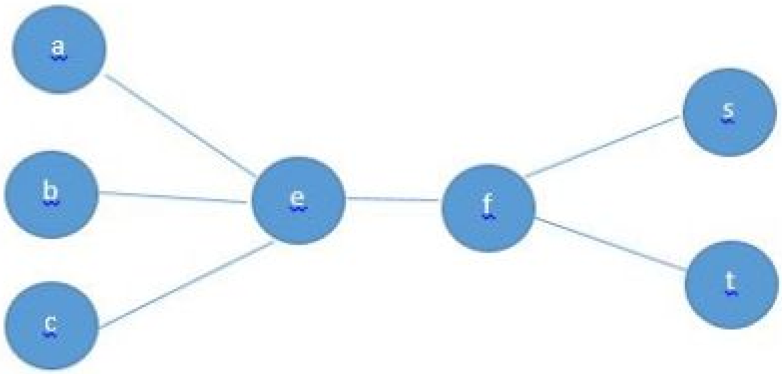
```
def twoHostSystem():
    topo = CreateTopo(n=2)
    net = Mininet(topo=topo)
    net.start()
    h1 = net.get('h1')
    h2 = net.get('h2')
    thread.start_new_thread(server, (net, ))
    h2.cmd('iperf -c 10.0.0.1 -p 5111 -t 20')
```

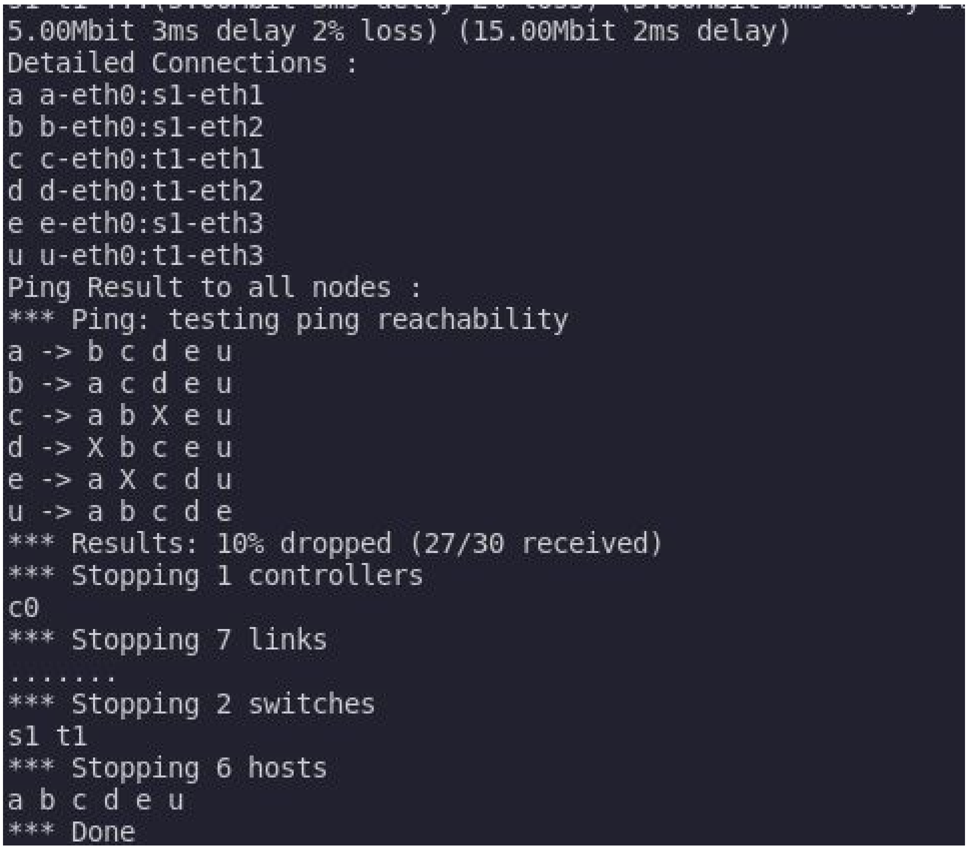
This code server host (h1) will have port number 5111 and monitor the results every 2 seconds (obtain the statistics after every 2 seconds).



```
def server(net):
    h1 = net.get('h1')
    h1.cmd('iperf -s -p 5111 -i 2 > throughput.txt')
```

2. In this code question, I created network topology that comprises of five hosts namely,a,b,c,s and t, and two switches namely e and f and the links between.

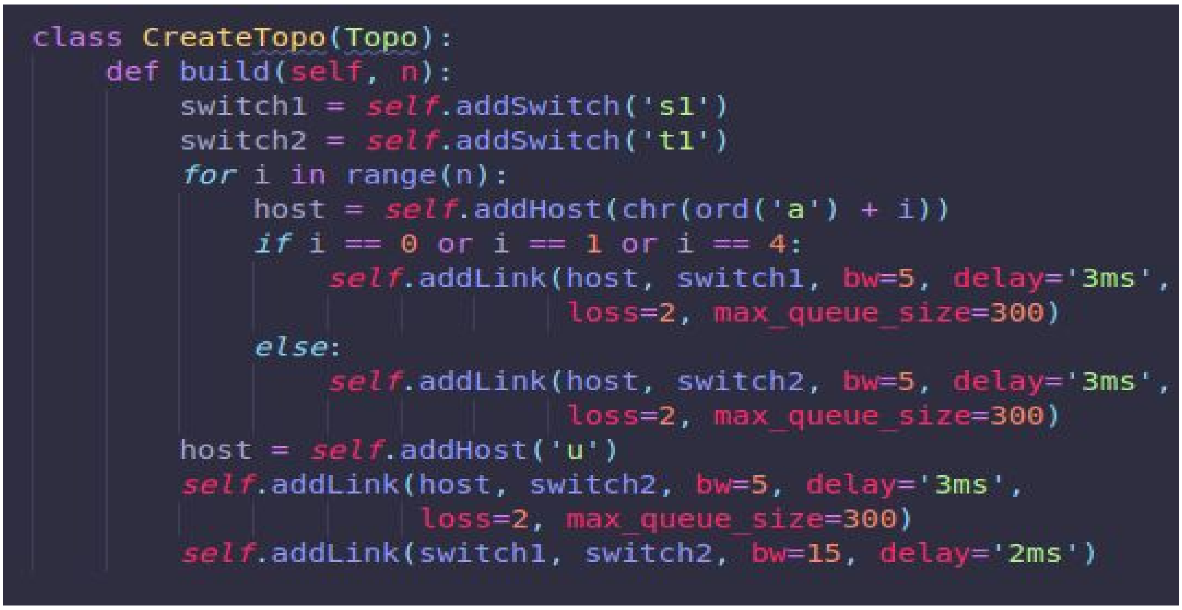




```
5.00Mbit 3ms delay 2% loss) (15.00Mbit 2ms delay)
Detailed Connections :
a a-eth0:s1-eth1
b b-eth0:s1-eth2
c c-eth0:t1-eth1
d d-eth0:t1-eth2
e e-eth0:s1-eth3
u u-eth0:t1-eth3
Ping Result to all nodes :
*** Ping: testing ping reachability
a -> b c d e u
b -> a c d e u
c -> a b X e u
d -> X b c e u
e -> a X c d u
u -> a b c d e
*** Results: 10% dropped (27/30 received)
*** Stopping 1 controllers
c0
*** Stopping 7 links
.......
*** Stopping 2 switches
s1 t1
*** Stopping 6 hosts
a b c d e u
*** Done
```

In my case three packets are lost it was denoted by X out of 30.

This code will create topology. With the given information in question. *Addlink is* a mininet api to create link between host and switch with given information.



```python
class CreateTopo(Topo):
    def build(self, n):
        switch1 = self.addSwitch('s1')
        switch2 = self.addSwitch('t1')
        for i in range(n):
            host = self.addHost(chr(ord('a') + i))
            if i == 0 or i == 1 or i == 4:
                self.addLink(host, switch1, bw=5, delay='3ms',
                             loss=2, max_queue_size=300)
            else:
                self.addLink(host, switch2, bw=5, delay='3ms',
                             loss=2, max_queue_size=300)
        host = self.addHost('u')
        self.addLink(host, switch2, bw=5, delay='3ms',
                     loss=2, max_queue_size=300)
        self.addLink(switch1, switch2, bw=15, delay='2ms')
```

addhost and addSwitch is a mininet api to create host and switch respectively.