CS348 Computer Networks
Assignment 6

Shikhar Jaiswal
1601CS44

Q1) We have to write Python in Mininet to create 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. The server host (h1) will have port number 5111 and monitor the results every 2 seconds (obtain the statistics after every 2 seconds). The client host (h2) will send packets for 20 seconds. We then show the throughput for the transfer.

We start by importing the necessary modules:

```
from mininet.topo import Topo
from mininet.net import Mininet
from mininet.util import dumpNodeConnections
from mininet.log import setLogLevel
from mininet.cli import CLI
import thread
from time import sleep
```

Then, we define the `server(net)` function, and pings to all the hosts on the port number 5111 every two seconds and writes the output to `throughput.txt` file:

```
def server(net):
      """
      Base class for defining the topology.
      """
      h1 = net.get('h1')
      h1.cmd('iperf -s -p 5111 -i 2 > throughput.txt')
```

After that, we define the `SingleSwitch(Topo)` class for defining our topology by adding hosts, switches and links:

```
class SingleSwitch(Topo):
      """
      Base class for defining the topology.
      """
      def build(self):
      switch = self.addSwitch('s1')
      host1 = self.addHost('h1')
      host2 = self.addHost('h2')
      self.addLink(host1, switch)
      self.addLink(host2, switch)
```

We also define the `runner()` function to initialize the network, and assign an IP address and port number to one of the hosts to make it a server for 20 seconds:

```
def runner():
      topo = SingleSwitch()
      net = Mininet(topo = topo)
      net.start()
      dumpNodeConnections(net.hosts)
      h2 = net.get('h2')
      thread.start_new_thread(server, (net,))
      h2.cmd('iperf -c 10.0.0.1 -p 5111 -t 20')
```

At last, we define the procedure to read the output written to the `throughput.txt` file:

```
if __name__ == '__main__':
      # Tell mininet to print useful information
      setLogLevel('info')
      runner()
      count = 0
      print("Required Throughput")
      print("Time Interval        Throughput")
      with open("throughput.txt") as f:
      contents = f.readlines()
      contents = [x.strip() for x in contents]
      for i in range(6, len(contents)):
      line = contents[i].split()
      if i <= 9:
            print("[%s %s %s] ---> %s %s" % (line[2], line[3],
```

```
line[4], float(line[5]) / 2, line[6] + "/sec"))
      elif i == len(contents) - 1:
            print("[%s %s] ---> %s %s " % (line[2], line[3],
float(line[4]) / 20, line[5] + "/sec"))
      else:
            print("[%s %s] ---> %s %s " % (line[2], line[3],
float(line[4]) / 2, line[5] + "/sec"))
```

Then we finally get the result:

```
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
Required Throughput
Time Interval    Throughput
[0.0- 2.0 sec] ---> 1.435 GBytes/sec
[2.0- 4.0 sec] ---> 1.42 GBytes/sec
[4.0- 6.0 sec] ---> 1.41 GBytes/sec
[6.0- 8.0 sec] ---> 1.405 GBytes/sec
[8.0-10.0 sec] ---> 1.44 GBytes/sec
[10.0-12.0 sec] ---> 1.43 GBytes/sec
[12.0-14.0 sec] ---> 1.425 GBytes/sec
[14.0-16.0 sec] ---> 1.435 GBytes/sec
[16.0-18.0 sec] ---> 1.43 GBytes/sec
[0.0-20.0 sec] ---> 1.42 GBytes/sec
```

Q2) We have to write Python to create a 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 them are as shown in the figure. Each host and switch have an IP address and MAC address set. The topology has the following characteristics:

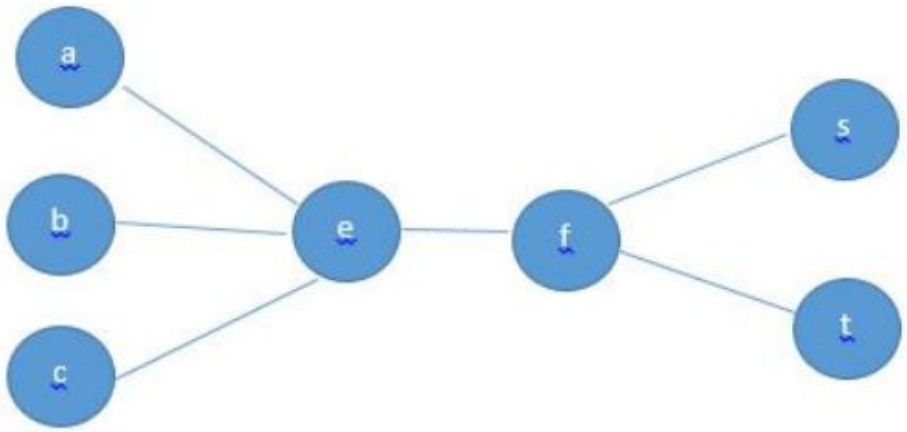

Figure 1: A Topology

The bandwidth between the hosts and the switch is 5Mbps with transmission delay of 3ms and 2% packet drop. The queue in switch can accommodate a maximum of 300 packets. The bandwidth between the switches is 15Mbps and has a transmission delay of 2ms. The program shows the detailed connectivity of the network and further checks it by ping command.

We start by importing the necessary modules:

```
from mininet.topo import Topo
from mininet.net import Mininet
from mininet.link import TCLink
from mininet.util import dumpNodeConnections
from mininet.log import setLogLevel
```

Then we define the `MultiSwitch(Topo)` class for defining our topology by adding hosts, switches and links:

```
class MultiSwitch(Topo):
      """
      Base class for defining the topology.
      """
```

```
    def build(self):
    e = self.addSwitch('e1', ip = '192.168.0.100/24', mac =
'00:00:00:00:00:09')
    f = self.addSwitch('f1', ip = '192.168.0.101/24', mac =
'00:00:00:00:00:10')
    a = self.addHost('a', ip = '192.168.0.102/24', mac =
'00:00:00:00:00:00')
    b = self.addHost('b', ip = '192.168.0.103/24', mac =
'00:00:00:00:01:00')
    c = self.addHost('c', ip = '192.168.0.104/24', mac =
'00:00:00:00:02:00')
    s = self.addHost('s', ip = '192.168.0.106/24', mac =
'00:00:00:00:04:00')
    t = self.addHost('t', ip = '192.168.0.107/24', mac =
'00:00:00:00:05:00')
    self.addLink(a, e, bw = 5, delay = '3ms', loss = 2,
max_queue_size = 300)
    self.addLink(b, e, bw = 5, delay = '3ms', loss = 2,
max_queue_size = 300)
    self.addLink(c, e, bw = 5, delay = '3ms', loss = 2,
max_queue_size = 300)
    self.addLink(s, f, bw = 5, delay = '3ms', loss = 2,
max_queue_size = 300)
    self.addLink(t, f, bw = 5, delay = '3ms', loss = 2,
max_queue_size = 300)
    self.addLink(e, f, bw = 15, delay = '2ms')
```

We also define the `runner()` function to initialize the network, and initiate a `pingAll()`:

```
def runner():
    topo = MultiSwitch()
    net = Mininet(topo = topo, link = TCLink)
    net.start()
    print "Dumping host connections"
    dumpNodeConnections(net.hosts)
    print "Testing network connectivity"
    net.pingAll()
    net.stop()
```

At last, we define the procedure to call the `runner()` function:

```
if __name__ == '__main__':
      # Tell mininet to print useful information
      setLogLevel('info')
      runner()
```

Then we finally get the result:

```
*** Creating network
*** Adding controller
*** Adding hosts:
a b c s t
*** Adding switches:
e1 f1
*** Adding links:
(5.00Mbit 3ms delay 2% loss) (5.00Mbit 3ms delay 2% loss) (a, e1)
(5.00Mbit 3ms delay 2% loss) (5.00Mbit 3ms delay 2% loss) (b, e1)
(5.00Mbit 3ms delay 2% loss) (5.00Mbit 3ms delay 2% loss) (c, e1)
(15.00Mbit 2ms delay) (15.00Mbit 2ms delay) (e1, f1) (5.00Mbit 3ms
delay 2% loss) (5.00Mbit 3ms delay 2% loss) (s, f1) (5.00Mbit 3ms
delay 2% loss) (5.00Mbit 3ms delay 2% loss) (t, f1)
*** Configuring hosts
a b c s t
*** Starting controller
c0
*** Starting 2 switches
e1 f1 ...(5.00Mbit 3ms delay 2% loss) (5.00Mbit 3ms delay 2% loss)
(5.00Mbit 3ms delay 2% loss) (15.00Mbit 2ms delay) (5.00Mbit 3ms
delay 2% loss) (5.00Mbit 3ms delay 2% loss) (15.00Mbit 2ms delay)
Dumping host connections
a a-eth0:e1-eth1
b b-eth0:e1-eth2
c c-eth0:e1-eth3
s s-eth0:f1-eth1
t t-eth0:f1-eth2
Testing network connectivity
*** Ping: testing ping reachability
a -> b c s t
b -> a c s t
```

```
c -> X b s t
s -> a b c t
t -> a b c s
*** Results: 5% dropped (19/20 received)
*** Stopping 1 controllers
c0
*** Stopping 6 links
......
*** Stopping 2 switches
e1 f1
*** Stopping 5 hosts
a b c s t
*** Done
```