Assignment 3: Susmita Mondal (20310071)

Mininet-based

- Tool Setup:
  1) Virtual Box
  2) Mininet-VM
  3) Iperf3
  4) Xming server :0.0
  5) Putty x-11 enable in ":0.0"
  6) Putty Session IP of required virtual machine

- Migrating to Mininet & Analysis

1. A. Need to define a program where server and client fetch the previous program
   - The program is in the same directory as the previous server and client program
   - the topology is to be defined in the program for mininet

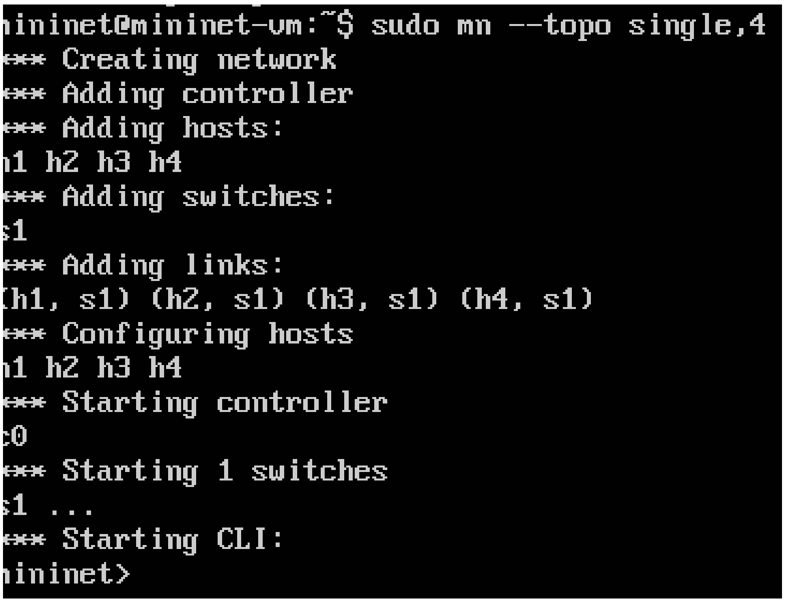
2. B. (a) Single Topology

   - swich 4 host. 1 host act as server, rest as client.
   - Sudo mn –topo single,4

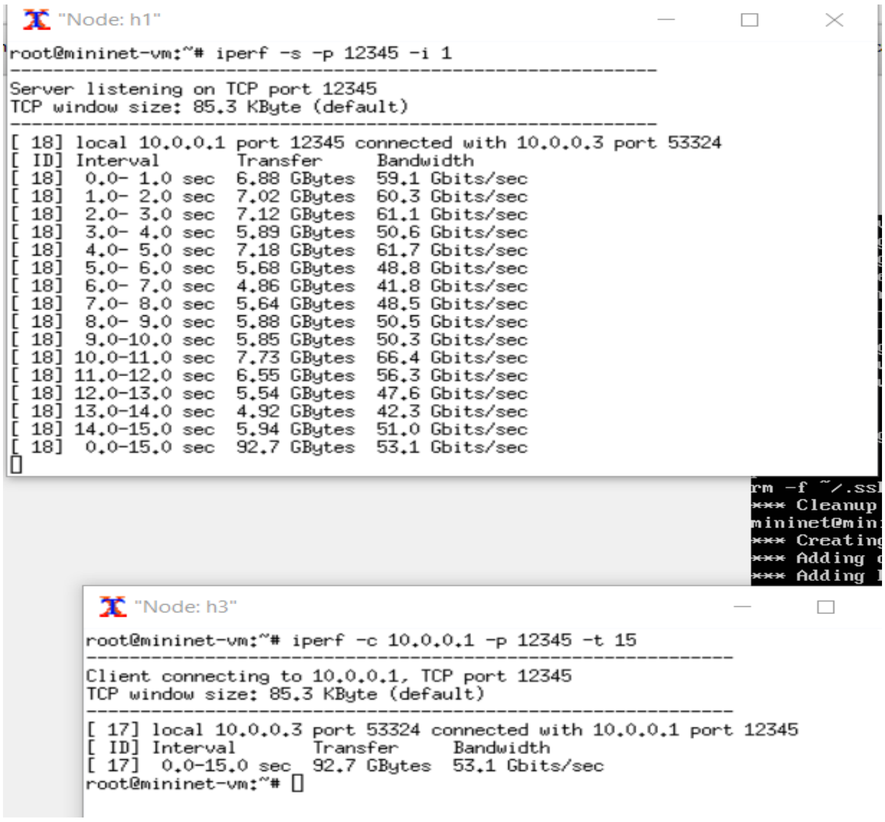
   i. Server-client model on port 12345

   ii. The program is written as single.py

   iii. Sudo python3 single.py to run



```
ininet@mininet-vm:~$ sudo mn --topo single,4
*** Creating network
*** Adding controller
*** Adding hosts:
1 h2 h3 h4
*** Adding switches:
1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1) (h4, s1)
*** Configuring hosts
1 h2 h3 h4
*** Starting controller
0
*** Starting 1 switches
1 ...
*** Starting CLI:
ininet>
```

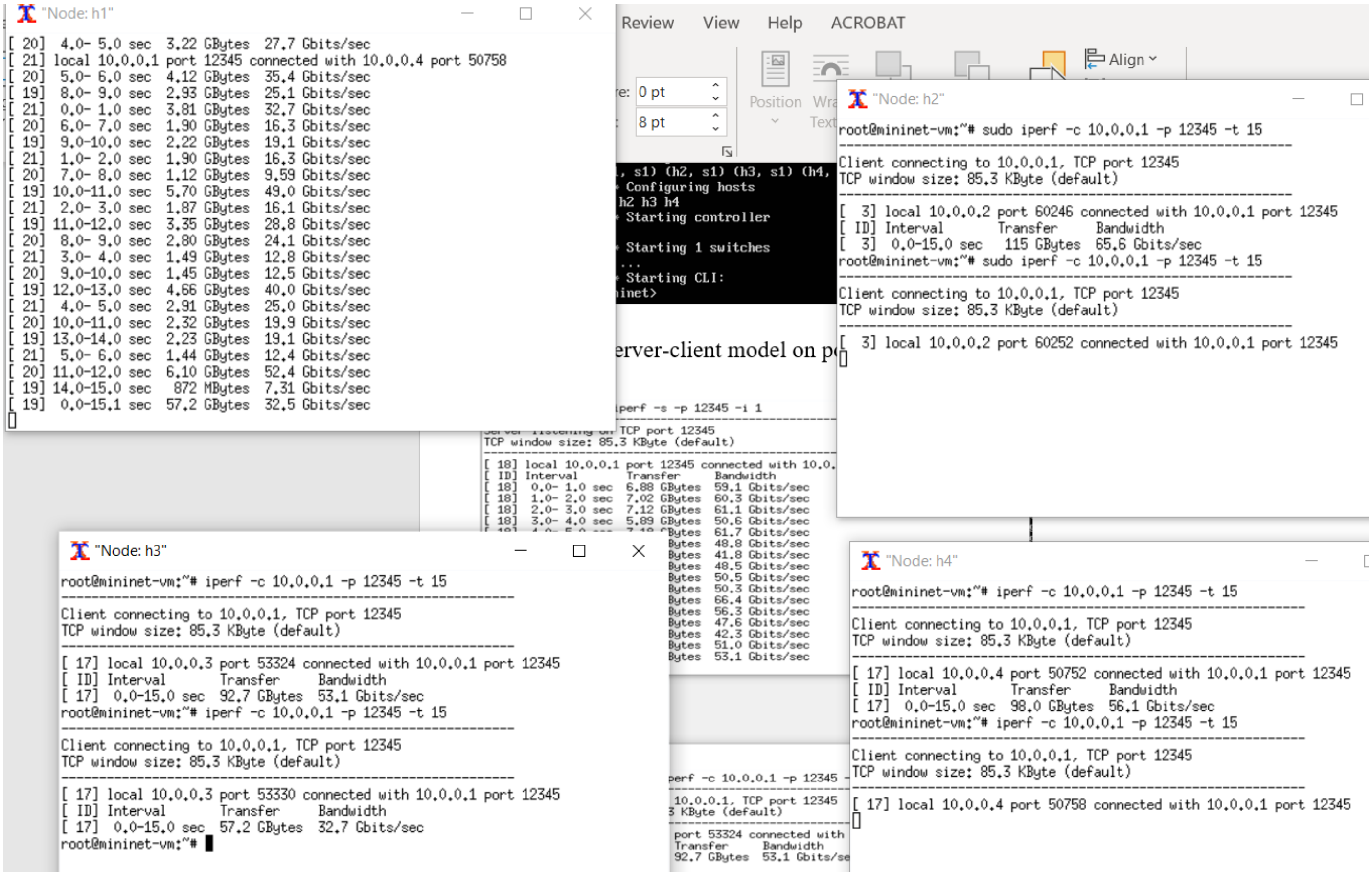


```
root@mininet-vm:~# iperf -s -p 12345 -i 1
------------------------------------------------------------
Server listening on TCP port 12345
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[ 18] local 10.0.0.1 port 12345 connected with 10.0.0.3 port 53324
[ ID] Interval       Transfer     Bandwidth
[ 18]  0.0- 1.0 sec  6.88 GBytes  59.1 Gbits/sec
[ 18]  1.0- 2.0 sec  7.02 GBytes  60.3 Gbits/sec
[ 18]  2.0- 3.0 sec  7.12 GBytes  61.1 Gbits/sec
[ 18]  3.0- 4.0 sec  5.89 GBytes  50.6 Gbits/sec
[ 18]  4.0- 5.0 sec  7.18 GBytes  61.7 Gbits/sec
[ 18]  5.0- 6.0 sec  5.68 GBytes  48.8 Gbits/sec
[ 18]  6.0- 7.0 sec  4.86 GBytes  41.8 Gbits/sec
[ 18]  7.0- 8.0 sec  5.64 GBytes  48.5 Gbits/sec
[ 18]  8.0- 9.0 sec  5.88 GBytes  50.5 Gbits/sec
[ 18]  9.0-10.0 sec  5.85 GBytes  50.3 Gbits/sec
[ 18] 10.0-11.0 sec  7.73 GBytes  66.4 Gbits/sec
[ 18] 11.0-12.0 sec  6.55 GBytes  56.3 Gbits/sec
[ 18] 12.0-13.0 sec  5.54 GBytes  47.6 Gbits/sec
[ 18] 13.0-14.0 sec  4.92 GBytes  42.3 Gbits/sec
[ 18] 14.0-15.0 sec  5.94 GBytes  51.0 Gbits/sec
[ 18]  0.0-15.0 sec  92.7 GBytes  53.1 Gbits/sec
```

```
rm -f ~/.ssl
*** Cleanup
mininet@min
*** Creating
*** Adding
*** Adding
```

```
root@mininet-vm:~# iperf -c 10.0.0.1 -p 12345 -t 15
------------------------------------------------------------
Client connecting to 10.0.0.1, TCP port 12345
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[ 17] local 10.0.0.3 port 53324 connected with 10.0.0.1 port 12345
[ ID] Interval       Transfer     Bandwidth
[ 17]  0.0-15.0 sec  92.7 GBytes  53.1 Gbits/sec
root@mininet-vm:~# 
```

Concurrent mode:



```
[ 20]  4.0- 5.0 sec  3.22 GBytes  27.7 Gbits/sec
[ 21] local 10.0.0.1 port 12345 connected with 10.0.0.4 port 50758
[ 20]  5.0- 6.0 sec  4.12 GBytes  35.4 Gbits/sec
[ 19]  8.0- 9.0 sec  2.93 GBytes  25.1 Gbits/sec
[ 21]  0.0- 1.0 sec  3.81 GBytes  32.7 Gbits/sec
[ 20]  6.0- 7.0 sec  1.90 GBytes  16.3 Gbits/sec
[ 19]  9.0-10.0 sec  2.22 GBytes  19.1 Gbits/sec
[ 21]  1.0- 2.0 sec  1.90 GBytes  16.3 Gbits/sec
[ 20]  7.0- 8.0 sec  1.12 GBytes  9.59 Gbits/sec
[ 19] 10.0-11.0 sec  5.70 GBytes  49.0 Gbits/sec
[ 21]  2.0- 3.0 sec  1.87 GBytes  16.1 Gbits/sec
[ 19] 11.0-12.0 sec  3.35 GBytes  28.8 Gbits/sec
[ 20]  8.0- 9.0 sec  2.80 GBytes  24.1 Gbits/sec
[ 21]  3.0- 4.0 sec  1.49 GBytes  12.8 Gbits/sec
[ 20]  9.0-10.0 sec  1.45 GBytes  12.5 Gbits/sec
[ 19] 12.0-13.0 sec  4.66 GBytes  40.0 Gbits/sec
[ 21]  4.0- 5.0 sec  2.91 GBytes  25.0 Gbits/sec
[ 20] 10.0-11.0 sec  2.32 GBytes  19.9 Gbits/sec
[ 19] 13.0-14.0 sec  2.23 GBytes  19.1 Gbits/sec
[ 21]  5.0- 6.0 sec  1.44 GBytes  12.4 Gbits/sec
[ 20] 11.0-12.0 sec  6.10 GBytes  52.4 Gbits/sec
[ 19] 14.0-15.0 sec   872 MBytes  7.31 Gbits/sec
[ 19]  0.0-15.1 sec  57.2 GBytes  32.5 Gbits/sec
```

```
, s1) (h2, s1) (h3, s1) (h4,
 Configuring hosts
h2 h3 h4
 Starting controller

 Starting 1 switches
...
 Starting CLI:
inet>
```

erver-client model on p

```
root@mininet-vm:~# sudo iperf -c 10.0.0.1 -p 12345 -t 15
------------------------------------------------------------
Client connecting to 10.0.0.1, TCP port 12345
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.2 port 60246 connected with 10.0.0.1 port 12345
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-15.0 sec   115 GBytes  65.6 Gbits/sec
root@mininet-vm:~# sudo iperf -c 10.0.0.1 -p 12345 -t 15
------------------------------------------------------------
Client connecting to 10.0.0.1, TCP port 12345
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.2 port 60252 connected with 10.0.0.1 port 12345
```

```
root@mininet-vm:~# iperf -c 10.0.0.1 -p 12345 -t 15
------------------------------------------------------------
Client connecting to 10.0.0.1, TCP port 12345
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[ 17] local 10.0.0.3 port 53324 connected with 10.0.0.1 port 12345
[ ID] Interval       Transfer     Bandwidth
[ 17]  0.0-15.0 sec  92.7 GBytes  53.1 Gbits/sec
root@mininet-vm:~# iperf -c 10.0.0.1 -p 12345 -t 15
------------------------------------------------------------
Client connecting to 10.0.0.1, TCP port 12345
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[ 17] local 10.0.0.3 port 53330 connected with 10.0.0.1 port 12345
[ ID] Interval       Transfer     Bandwidth
[ 17]  0.0-15.0 sec  57.2 GBytes  32.7 Gbits/sec
root@mininet-vm:~# 
```

```
root@mininet-vm:~# iperf -c 10.0.0.1 -p 12345 -t 15
------------------------------------------------------------
Client connecting to 10.0.0.1, TCP port 12345
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[ 17] local 10.0.0.4 port 50752 connected with 10.0.0.1 port 12345
[ ID] Interval       Transfer     Bandwidth
[ 17]  0.0-15.0 sec  98.0 GBytes  56.1 Gbits/sec
root@mininet-vm:~# iperf -c 10.0.0.1 -p 12345 -t 15
------------------------------------------------------------
Client connecting to 10.0.0.1, TCP port 12345
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[ 17] local 10.0.0.4 port 50758 connected with 10.0.0.1 port 12345
```

All data can be requested in a concurrent fashion.

Transfer speed is much higher than earlier (non-mininet)

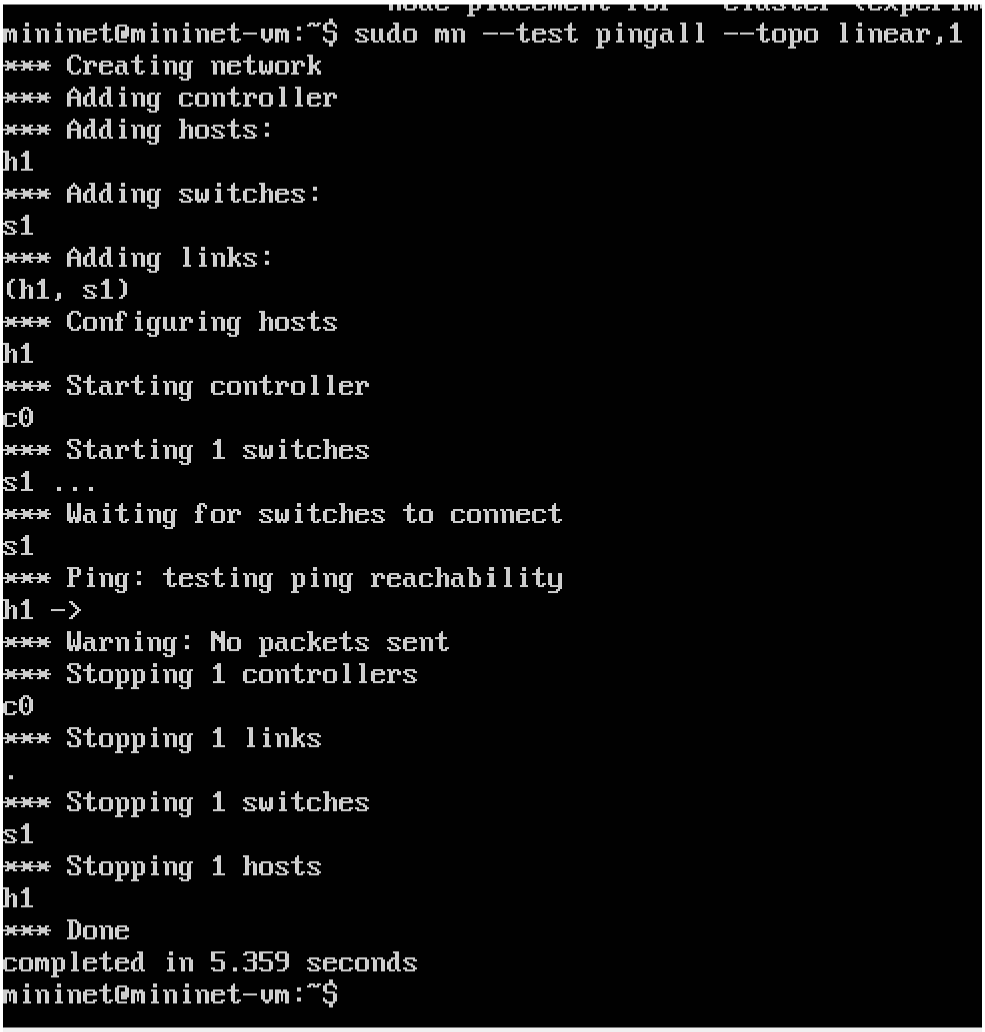
(b) Linear Topology Bandwidth

-2 node 1 switch

- link bandwidth 10Mb, 100 Mb and 1Gbps respectively

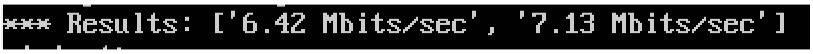
  - sudo mn –test pingall –topo liner,1

  - Sudo mn –link tc,bw=10
  - Sudo mn –link tc,bw=100
  - Sudo mn –link tc,bw=1Gb
  - Server-client model on port 12345
  - The program is written as bw.py
  - Sudo python3 bw.py to run



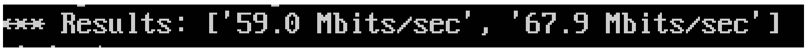
```
mininet@mininet-vm:~$ sudo mn --test pingall --topo linear,1
*** Creating network
*** Adding controller
*** Adding hosts:
h1
*** Adding switches:
s1
*** Adding links:
(h1, s1)
*** Configuring hosts
h1
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Waiting for switches to connect
s1
*** Ping: testing ping reachability
h1 ->
*** Warning: No packets sent
*** Stopping 1 controllers
c0
*** Stopping 1 links
.
*** Stopping 1 switches
s1
*** Stopping 1 hosts
h1
*** Done
completed in 5.359 seconds
mininet@mininet-vm:~$
```

➔ 10Mb result



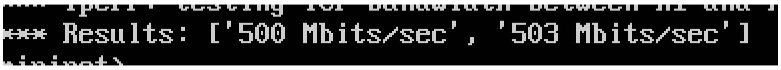
```
*** Results: ['6.42 Mbits/sec', '7.13 Mbits/sec']
```

➔ 100Mb result



```
*** Results: ['59.0 Mbits/sec', '67.9 Mbits/sec']
```

➔ 1 Gb result



```
*** Results: ['500 Mbits/sec', '503 Mbits/sec']
```

Basically, the 10Mb link has the best throughput among these three links.

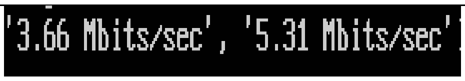
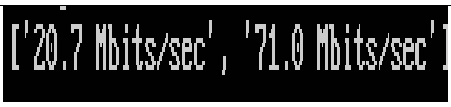
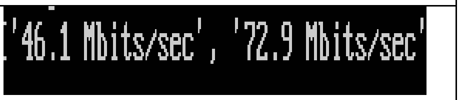
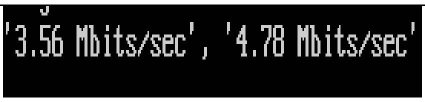
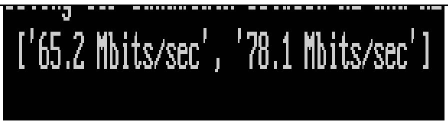
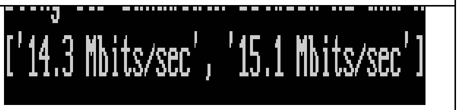
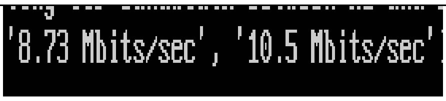
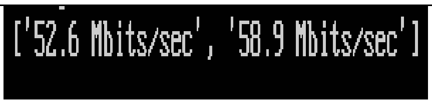
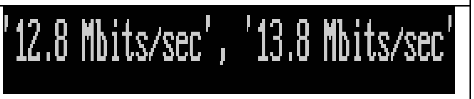
(c) Linear Topology Delay

-2 node 1 switch

- link bandwidth 10 Mb, 100 Mb and 1Gbps respectively

    - sudo mn –test pingall –topo liner,1

- Sudo mn –link tc,bw=10, delay=1ms / 2ms / 5ms
- Sudo mn –link tc,bw=100, delay=1ms / 2ms / 5ms
- Sudo mn –link tc,bw=1Gb, delay=1ms / 2ms / 5ms
- Server-client model on port 12345
- The program is written as delay.py
- Sudo python3 delay.py to run

Link	10Mb	100Mb	1Gb
Delay = 1ms	 '3.66 Mbits/sec', '5.31 Mbits/sec'	 ['20.7 Mbits/sec', '71.0 Mbits/sec']	 '46.1 Mbits/sec', '72.9 Mbits/sec'
Delay = 2ms	 '3.56 Mbits/sec', '4.78 Mbits/sec'	 ['65.2 Mbits/sec', '78.1 Mbits/sec']	 ['14.3 Mbits/sec', '15.1 Mbits/sec']
Delay = 5ms	 '8.73 Mbits/sec', '10.5 Mbits/sec'	 ['52.6 Mbits/sec', '58.9 Mbits/sec']	 '12.8 Mbits/sec', '13.8 Mbits/sec'

- 10Mb link has better throughput in 5ms delay,
- 100Mb link has better throughput in 2ms delay
- 1Gb link has better throughput in 1ms delay
  Throughput vary depending upon Link speed as well as delay

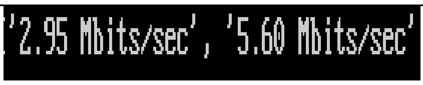
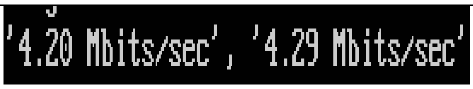
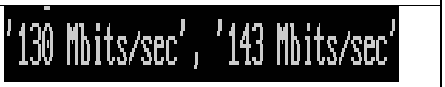
(d) Linear Topology Loss

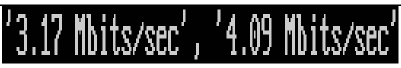
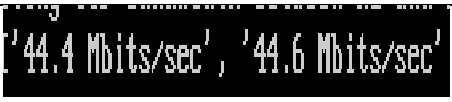
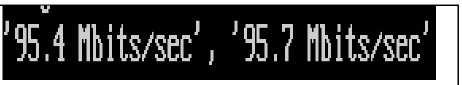
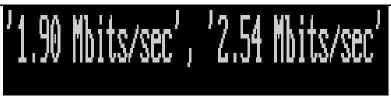
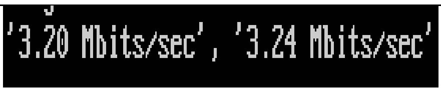
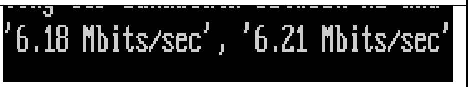
-2 node 1 switch

link bandwidth 10 Mb, 100 Mb and 1Gbps respectively

sudo mn –test pingall –topo liner,1

- Sudo mn –link tc,bw=10, loss =1% / 2% / 5%
- Sudo mn –link tc,bw=100, loss =1% / 2% / 5%
- Sudo mn –link tc,bw=1Gb,, loss =1% / 2% / 5%
- Server-client model on port 12345
- The program is written as loss.py
- Sudo python3 loss.py to run

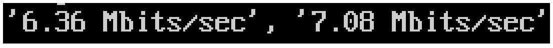
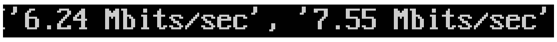
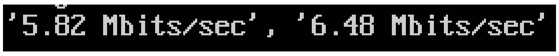
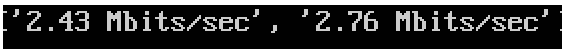
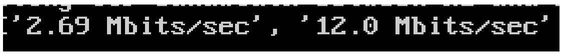
Link	10Mb	100Mb	1Gb
Loss 1%	 '2.95 Mbits/sec', '5.60 Mbits/sec'	 '4.20 Mbits/sec', '4.29 Mbits/sec'	 '130 Mbits/sec', '143 Mbits/sec'

Loss 2%	 '3.17 Mbits/sec', '4.09 Mbits/sec'	 '44.4 Mbits/sec', '44.6 Mbits/sec'	 '95.4 Mbits/sec', '95.7 Mbits/sec'
Loss 5%	 '1.90 Mbits/sec', '2.54 Mbits/sec'	 '3.20 Mbits/sec', '3.24 Mbits/sec'	 '6.18 Mbits/sec', '6.21 Mbits/sec'

2% loss has given a better performance overall

(e) Linear Topology Hops

-2 host (n) nodes 2 switches (k)

- Switches goes to n or n-1 depending upon even and odd number
- 2 hops are taken starting from 2 up to 10 switches (eg. 2,4,6,8,10)
- Server-client model on port 12345
- The program is written as hops.py
- Sudo python3 hops.py to run
  - ➔ 2 hops  '6.36 Mbits/sec', '7.08 Mbits/sec' - 0.006sec
  - ➔ 4 hops  '6.24 Mbits/sec', '7.55 Mbits/sec' - 0.017 sec
  - ➔ 6 hops  '5.82 Mbits/sec', '6.48 Mbits/sec' - 0.046 sec
  - ➔ 8 hops  '2.43 Mbits/sec', '2.76 Mbits/sec'- 0.091 sec
  - ➔ 10 hops  '2.69 Mbits/sec', '12.0 Mbits/sec'- 0.163sec
- The more hops, the more time taken and less bandwidth.

- Custom Topology

{i} Varying Link Bandwidth

-Aggregated 6.9Mbps-12.0Mbps

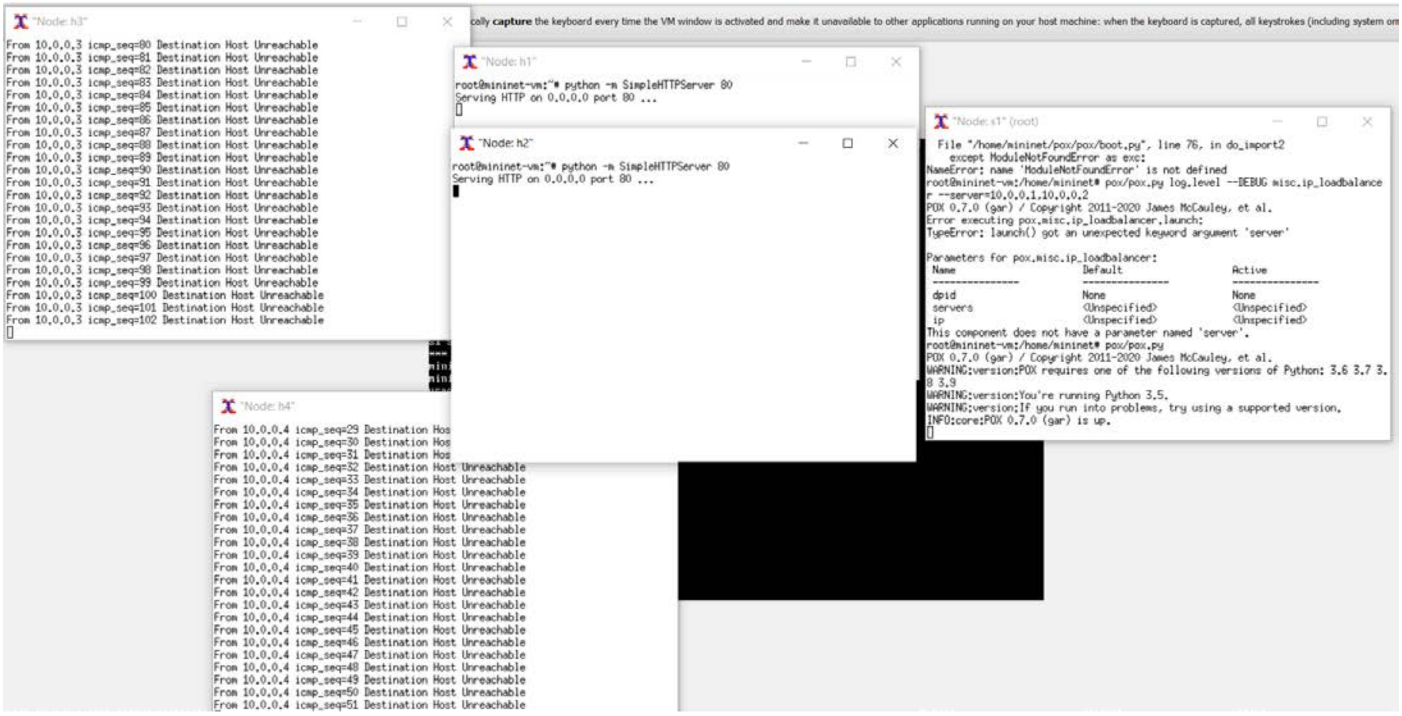
Though the highest capacity of the link is 40Mbps the throughput is very less than that. Consider bottle neck link which is 10Mbps

- The program is written as Bandwidth.py
- Sudo python3 Bandwidth.py to run
- Tree like topology with depth 3

{ii} Horizontal scaling and Load Balancing

- The program is written as Scale.py & Load.py
- Sudo python3 Scale.py & Load.py to run
- Tree like topology with depth 3

- Load Balancing



cally **capture** the keyboard every time the VM window is activated and make it unavailable to other applications running on your host machine: when the keyboard is captured, all keystrokes (including system om

```
From 10.0.0.3 icmp_seq=80 Destination Host Unreachable
From 10.0.0.3 icmp_seq=81 Destination Host Unreachable
From 10.0.0.3 icmp_seq=82 Destination Host Unreachable
From 10.0.0.3 icmp_seq=83 Destination Host Unreachable
From 10.0.0.3 icmp_seq=84 Destination Host Unreachable
From 10.0.0.3 icmp_seq=85 Destination Host Unreachable
From 10.0.0.3 icmp_seq=86 Destination Host Unreachable
From 10.0.0.3 icmp_seq=87 Destination Host Unreachable
From 10.0.0.3 icmp_seq=88 Destination Host Unreachable
From 10.0.0.3 icmp_seq=89 Destination Host Unreachable
From 10.0.0.3 icmp_seq=90 Destination Host Unreachable
From 10.0.0.3 icmp_seq=91 Destination Host Unreachable
From 10.0.0.3 icmp_seq=92 Destination Host Unreachable
From 10.0.0.3 icmp_seq=93 Destination Host Unreachable
From 10.0.0.3 icmp_seq=94 Destination Host Unreachable
From 10.0.0.3 icmp_seq=95 Destination Host Unreachable
From 10.0.0.3 icmp_seq=96 Destination Host Unreachable
From 10.0.0.3 icmp_seq=97 Destination Host Unreachable
From 10.0.0.3 icmp_seq=98 Destination Host Unreachable
From 10.0.0.3 icmp_seq=99 Destination Host Unreachable
From 10.0.0.3 icmp_seq=100 Destination Host Unreachable
From 10.0.0.3 icmp_seq=101 Destination Host Unreachable
From 10.0.0.3 icmp_seq=102 Destination Host Unreachable
```

```
root@mininet-vm:~# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
```

```
root@mininet-vm:~# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
```

```
 File "/home/mininet/pox/pox/boot.py", line 76, in do_import2
   except ModuleNotFoundError as exc:
NameError: name 'ModuleNotFoundError' is not defined
root@mininet-vm:/home/mininet# pox/pox.py log.level --DEBUG misc.ip_loadbalance
r --server=10.0.0.1,10.0.0.2
POX 0.7.0 (gar) / Copyright 2011-2020 James McCauley, et al.
Error executing pox.misc.ip_loadbalancer.launch:
TypeError: launch() got an unexpected keyword argument 'server'

Parameters for pox.misc.ip_loadbalancer:
 Name                     Default                  Active
 ---------------          ---------------          ---------------
 dpid                     None                     None
 servers                  <Unspecified>            <Unspecified>
 ip                       <Unspecified>            <Unspecified>
This component does not have a parameter named 'server'.
root@mininet-vm:/home/mininet# pox/pox.py
POX 0.7.0 (gar) / Copyright 2011-2020 James McCauley, et al.
WARNING:version:POX requires one of the following versions of Python: 3.6 3.7 3.
8 3.9
WARNING:version:You're running Python 3.5.
WARNING:version:If you run into problems, try using a supported version.
INFO:core:POX 0.7.0 (gar) is up.
```

```
From 10.0.0.4 icmp_seq=29 Destination Hos
From 10.0.0.4 icmp_seq=30 Destination Hos
From 10.0.0.4 icmp_seq=31 Destination Hos
From 10.0.0.4 icmp_seq=32 Destination Host Unreachable
From 10.0.0.4 icmp_seq=33 Destination Host Unreachable
From 10.0.0.4 icmp_seq=34 Destination Host Unreachable
From 10.0.0.4 icmp_seq=35 Destination Host Unreachable
From 10.0.0.4 icmp_seq=36 Destination Host Unreachable
From 10.0.0.4 icmp_seq=37 Destination Host Unreachable
From 10.0.0.4 icmp_seq=38 Destination Host Unreachable
From 10.0.0.4 icmp_seq=39 Destination Host Unreachable
From 10.0.0.4 icmp_seq=40 Destination Host Unreachable
From 10.0.0.4 icmp_seq=41 Destination Host Unreachable
From 10.0.0.4 icmp_seq=42 Destination Host Unreachable
From 10.0.0.4 icmp_seq=43 Destination Host Unreachable
From 10.0.0.4 icmp_seq=44 Destination Host Unreachable
From 10.0.0.4 icmp_seq=45 Destination Host Unreachable
From 10.0.0.4 icmp_seq=46 Destination Host Unreachable
From 10.0.0.4 icmp_seq=47 Destination Host Unreachable
From 10.0.0.4 icmp_seq=48 Destination Host Unreachable
From 10.0.0.4 icmp_seq=49 Destination Host Unreachable
From 10.0.0.4 icmp_seq=50 Destination Host Unreachable
From 10.0.0.4 icmp_seq=51 Destination Host Unreachable
```

- The codes are submitted separately.