

Computer Networks Assignment 2

Name- Rahul Chembakasseril

Roll Number- 20110158

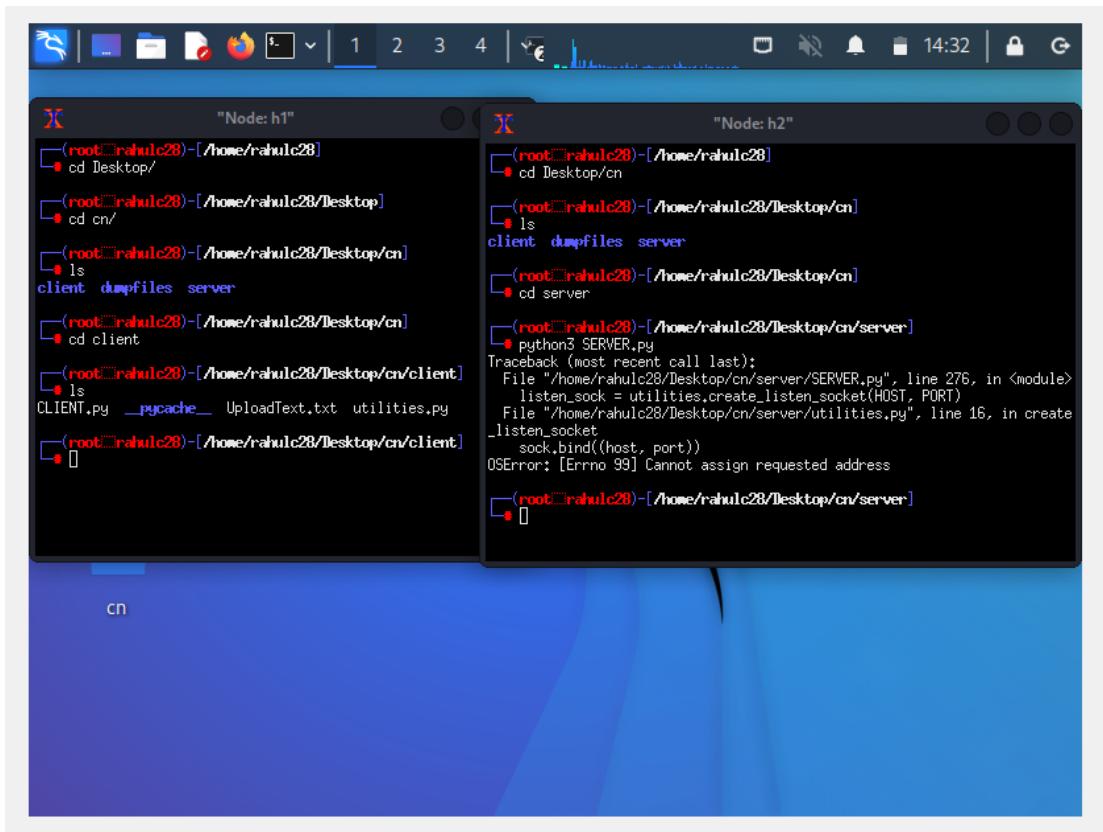
Question 1

- a) In assignment 1, I had hardcoded the IP address of the server as per the IP address on my Windows system, since I had run both client and server on my Windows system.

Now, I am running my server on h1 and client on h2. So, I have set the IP address of the server to 10.0.0.1. This change was made by setting the HOST value in the CLIENT.py file in the client directory.

10.0.0.1 was obtained by running the “h1 ifconfig” command in the mininet terminal.

-The errors obtained before making the edits in the code are as follows-



The screenshot shows two terminal windows side-by-side. The left window is titled "Node: h1" and the right window is titled "Node: h2". Both windows are root shells on the host 'rahulc28'.

In the Node h1 terminal, the user runs commands to navigate to the 'Desktop/cn' directory and lists files. It then attempts to run 'python3 SERVER.py' but fails with an OS error:

```
(root@rahulc28) [~/home/rahulc28/Desktop/cn]# python3 SERVER.py
OSError: [Errno 99] Cannot assign requested address
```

In the Node h2 terminal, the user runs commands to navigate to the 'Desktop/cn' directory and lists files. It then attempts to run 'python3 SERVER.py' but fails with a traceback indicating a problem with the listen socket:

```
(root@rahulc28) [~/home/rahulc28/Desktop/cn/server]# python3 SERVER.py
Traceback (most recent call last):
  File "/home/rahulc28/Desktop/cn/server/SERVER.py", line 276, in <module>
    listen_sock = utilities.create_listen_socket(HOST, PORT)
  File "/home/rahulc28/Desktop/cn/server/utilities.py", line 16, in create_listen_socket
    sock.bind((Host, port))
OSError: [Errno 99] Cannot assign requested address
```

-After making the changes, the Remote File System worked on the required commands in all the encryption modes-

```
"Node: h1"
[root@rahulc28 ~]# cd Desktop/server
bash: cd: Desktop/server: No such file or directory
[root@rahulc28 ~]# cd Desktop/cn/server
[root@rahulc28 ~]# python3 SERVER.py
Listening on ('0.0.0.0', 4040)
Connection from ('10.0.0.2', 48760)
('10.0.0.2', 48760); ls
Received message: ls
Closed connection to ('10.0.0.2', 48760)
Connection from ('10.0.0.2', 48352)
[]

"Node: h2"
[root@rahulc28 ~]# cd Desktop/cn/client
[root@rahulc28 ~]# python3 CLIENT.py
Connected to 10.0.0.1:4040
Choose your encryption mode - 0, 1 or 2: 0
Type your command, press enter to send, 'exit' to quit
ls
Sent message: ls
Received echo: ['__pycache__', 'Tests', 'SERVER.py', 'Test0.txt', 'Utilities.py']
]
Closed connection to server

Connected to 10.0.0.1:4040
Choose your encryption mode - 0, 1 or 2: ]
```

- b) To measure the performance, I used the time method available in Python.

Performance in Mininet- DOWNLOAD

The screenshot shows two terminal windows side-by-side. The left window is titled "Node: h1" and the right window is titled "Node: h2". Both windows are running under the root user on a Linux system (Ubuntu 14.04 LTS).

Node: h1 (Left Terminal):

- cd server
- python3 SERVER.py
- [STARTING] server is starting...
- [LISTENING] The server is listening
- [NEW CONNECTION] ('10.0.0.2', 57082) connected.
- Traceback (most recent call last):
- File "/home/rahulc28/Desktop/cn/server/SERVER.py", line 153, in <module>
 start_server()
File "/home/rahulc28/Desktop/cn/server/SERVER.py", line 149, in start_server
 handle_client(conn, addr)
File "/home/rahulc28/Desktop/cn/server/SERVER.py", line 104, in handle_client
 with open(filename, "rb") as f:
FileNotFoundError: [Errno 2] No such file or directory: 'Text0.txt'
- python3 SERVER.py
- [STARTING] server is starting...
- [LISTENING] The server is listening
- [NEW CONNECTION] ('10.0.0.2', 57778) connected.
- File Sent
- [NEW CONNECTION] ('10.0.0.2', 55608) connected.

Node: h2 (Right Terminal):

- ls
- CLIENT.py Text0.txt UploadText.txt
- python3 CLIENT.py
- Write your Message here: LS
- The current directory contains: ['SERVER.py', 'Test0.txt']
- Write your Message here: DWD[Test0.txt]
- Downloaded
- Download has taken: 15.51035s
- Write your Message here: ■

Time Taken - 15.51035s

UPLOAD

The image shows two terminal windows side-by-side. The left window, titled "Node: h1", contains the following text:

```
(root@rahulc28:~/Desktop/cn/server]
$ python3 SERVER.py
[STARTING] server is starting...
[LISTENING] The server is listening
[NEW CONNECTION] ('10.0.0.2', 57082) connected.
Traceback (most recent call last):
  File "/home/rahulc28/Desktop/cn/server/SERVER.py", line 153, in <module>
    start_server()
  File "/home/rahulc28/Desktop/cn/server/SERVER.py", line 149, in start_server
    handle_client(conn, addr)
  File "/home/rahulc28/Desktop/cn/server/SERVER.py", line 104, in handle_client
    with open(filename, "rb") as f:
FileNotFoundError: [Errno 2] No such file or directory: 'Text0.txt'

(root@rahulc28:~/Desktop/cn/server]
$ python3 SERVER.py
[STARTING] server is starting...
[LISTENING] The server is listening
[NEW CONNECTION] ('10.0.0.2', 57778) connected.
File Sent
[NEW CONNECTION] ('10.0.0.2', 55608) connected.
File Received
[NEW CONNECTION] ('10.0.0.2', 59306) connected.
```

The right window, titled "Node: h2", contains the following text:

```
(root@rahulc28:~/Desktop/cn/client]
$ ls
CLIENT.py Text0.txt UploadText.txt

(root@rahulc28:~/Desktop/cn/client]
$ python3 CLIENT.py
Write your Message here: LS
The current directory contains: ['SERVER.py', 'Test0.txt']
Write your Message here: DWD<Test0.txt>
Downloaded
Download has taken: 15.51035s
Write your Message here: UPK<UploadText.txt>
Uploaded
Upload has taken: 14.4169s
Write your Message here: ■
```

Time Taken - 14.4169s

Performance in Kali

DOWNLOAD

The image shows two terminal windows side-by-side. The left window, titled "rahulc28@rahulc28: ~/Desktop/cn/server", contains the following text:

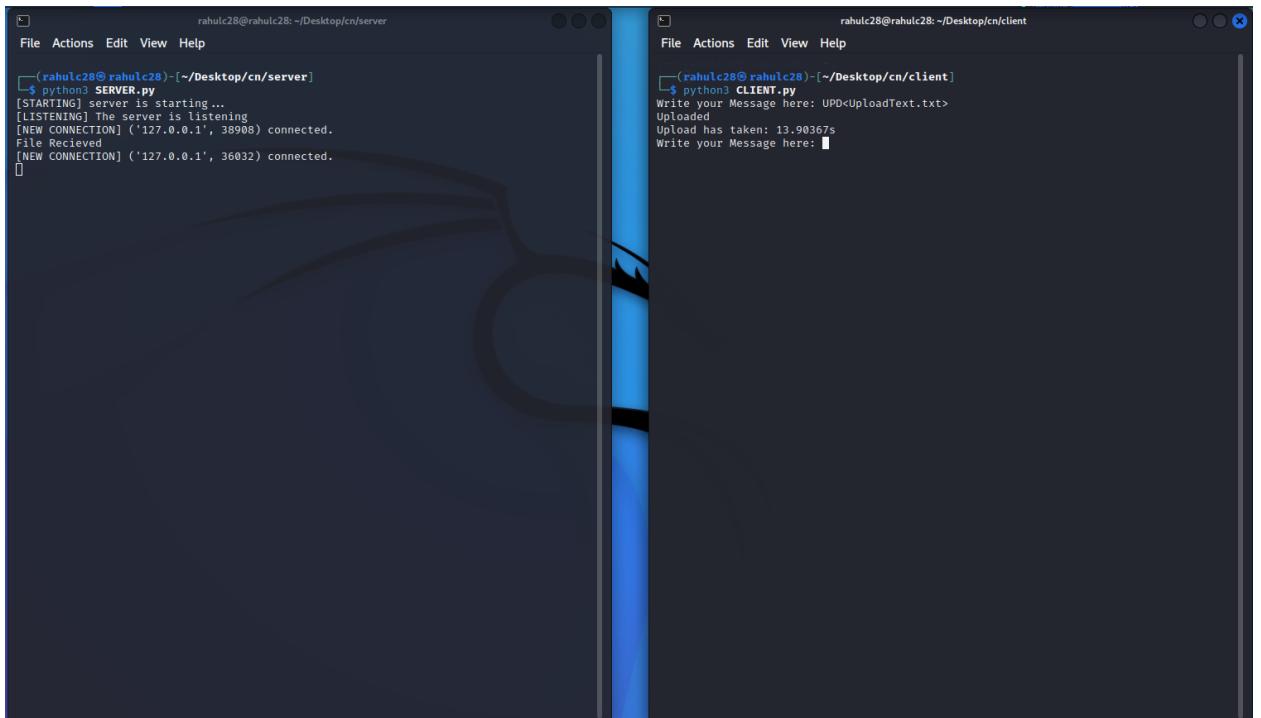
```
File Actions Edit View Help
(rahulc28@rahulc28: ~)
$ cd Desktop/cn/server
(rahulc28@rahulc28: ~/Desktop/cn/server]
$ python3 SERVER.py
[STARTING] server is starting...
[LISTENING] The server is listening
[NEW CONNECTION] ('127.0.0.1', 53854) connected.
File Sent
[NEW CONNECTION] ('127.0.0.1', 44722) connected.
```

The right window, titled "rahulc28@rahulc28: ~/Desktop/cn/client", contains the following text:

```
File Actions Edit View Help
(rahulc28@rahulc28: ~)
$ cd Desktop/cn/client
(rahulc28@rahulc28: ~/Desktop/cn/client]
$ python3 CLIENT.py
Write your Message here: DWD<Test0.txt>
Downloaded
Download has taken: 13.58164s
Write your Message here: ■
```

Time Taken = 13.58164s

UPLOAD



The image shows two terminal windows side-by-side. The left window is titled 'rahulc28@rahulc28: ~/Desktop/cn/server' and contains the following text:

```
[rahulc28@rahulc28:~/Desktop/cn/server]
$ python3 SERVER.py
[STARTING] server is starting...
[LISTENING] The server is listening
[NEW CONNECTION] ('127.0.0.1', 38908) connected.
File Received
[NEW CONNECTION] ('127.0.0.1', 36032) connected.
```

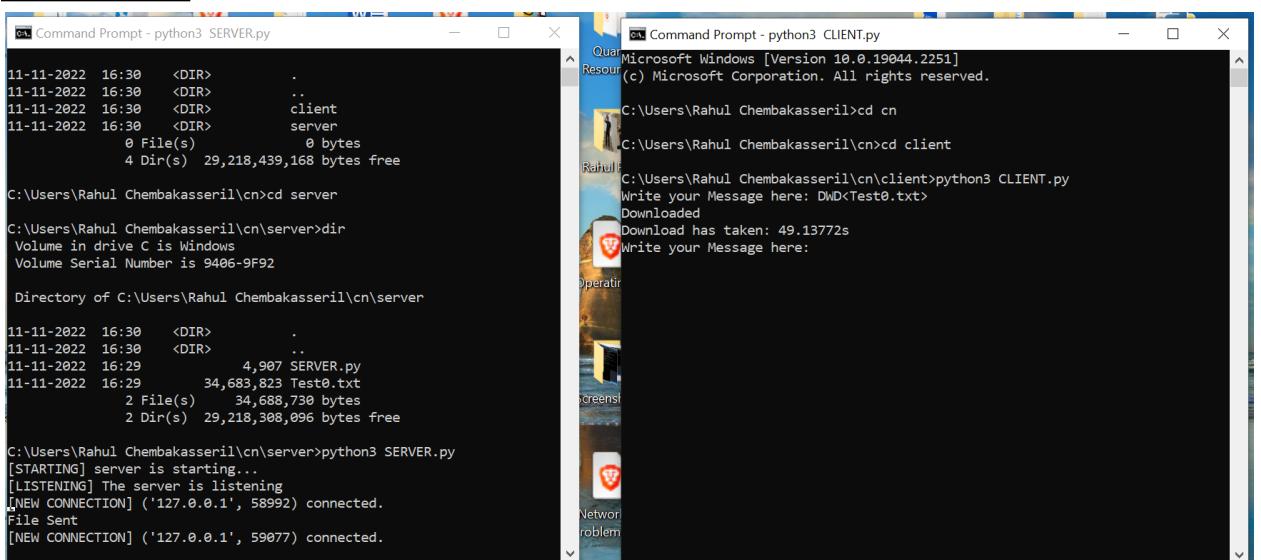
The right window is titled 'rahulc28@rahulc28: ~/Desktop/cn/client' and contains the following text:

```
[rahulc28@rahulc28:~/Desktop/cn/client]
$ python3 CLIENT.py
Write your Message here: UPD>UploadText.txt
Uploaded
Upload has taken: 13.90367s
Write your Message here: ■
```

Time Taken = 13.90367s

Performance in Windows

DOWNLOAD



The image shows two Command Prompt windows. The left window is titled 'Command Prompt - python3 SERVER.py' and contains the following text:

```
11-11-2022 16:30 <DIR> .
11-11-2022 16:30 <DIR> ..
11-11-2022 16:30 <DIR> client
11-11-2022 16:30 <DIR> server
    0 File(s)      0 bytes
    4 Dir(s)  29,218,439,168 bytes free

C:\Users\Rahul Chembakasseril\cn>cd server

C:\Users\Rahul Chembakasseril\cn\server>dir
Volume in drive C is Windows
Volume Serial Number is 9406-9F92

Directory of C:\Users\Rahul Chembakasseril\cn\server

11-11-2022 16:30 <DIR> .
11-11-2022 16:30 <DIR> ..
11-11-2022 16:29           4,907 SERVER.py
11-11-2022 16:29        34,683,823 Test0.txt
    2 File(s)   34,688,730 bytes
    2 Dir(s)  29,218,308,096 bytes free

C:\Users\Rahul Chembakasseril\cn\server>python3 SERVER.py
[STARTING] server is starting...
[LISTENING] The server is listening
[NEW CONNECTION] ('127.0.0.1', 58992) connected.
File Sent
[NEW CONNECTION] ('127.0.0.1', 59077) connected.
```

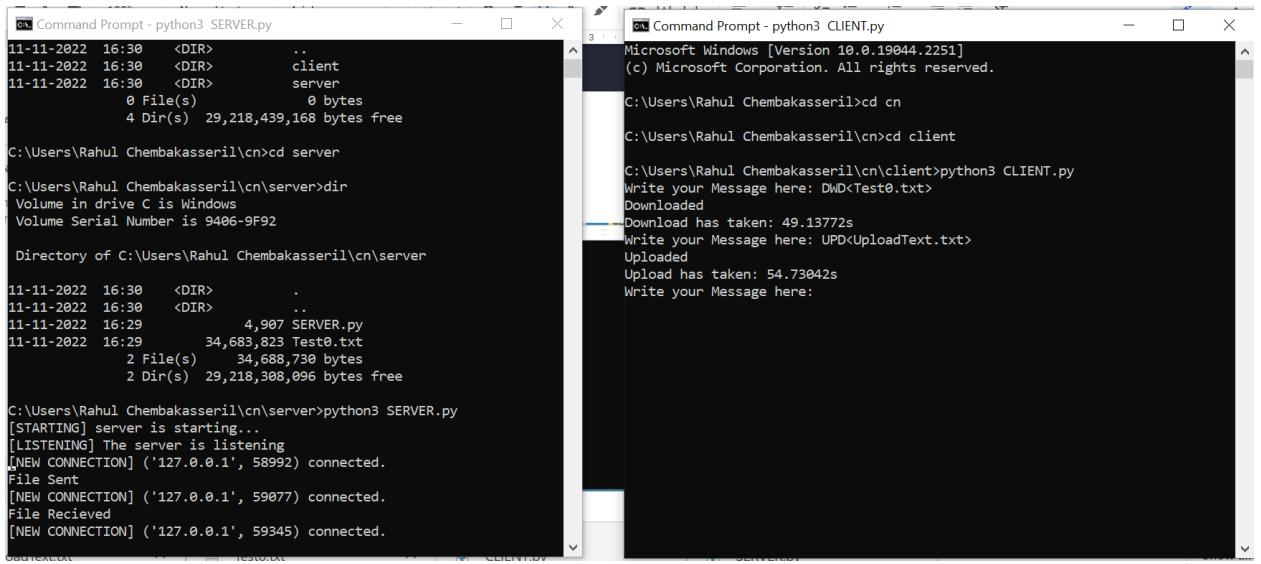
The right window is titled 'Command Prompt - python3 CLIENT.py' and contains the following text:

```
Microsoft Windows [Version 10.0.19044.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Rahul Chembakasseril>cd cn
C:\Users\Rahul Chembakasseril\cn>cd client
Rahul:
C:\Users\Rahul Chembakasseril\cn\client>python3 CLIENT.py
Write your Message here: DWD>Test0.txt
Downloaded
Download has taken: 49.13772s
Write your Message here: ■
```

Time Taken - 49.13772s

UPLOAD



The image shows two Command Prompt windows side-by-side. The left window, titled 'Command Prompt - python3 SERVER.py', displays the log output of a Python server running on a Windows machine. It shows directory listings and file transfers. The right window, titled 'Command Prompt - python3 CLIENT.py', shows the corresponding client logs on the same Windows machine. The client logs indicate the upload of a file named 'Test0.txt' and its download from the server. Both windows show the command prompt and the log output of the respective programs.

```
11-11-2022 16:30 <DIR> ..
11-11-2022 16:30 <DIR> client
11-11-2022 16:30 <DIR> server
    0 File(s)      0 bytes
    4 Dir(s) 29,218,439,168 bytes free

C:\Users\Rahul Chembakasseril\cn>cd server

C:\Users\Rahul Chembakasseril\cn>server>dir
Volume in drive C is Windows
Volume Serial Number is 9406-9F92

Directory of C:\Users\Rahul Chembakasseril\cn\server

11-11-2022 16:30 <DIR> .
11-11-2022 16:30 <DIR> ..
11-11-2022 16:29           4,907 SERVER.py
11-11-2022 16:29     34,683,823 Test0.txt
    2 File(s)   34,688,730 bytes
    2 Dir(s) 29,218,368,096 bytes free

C:\Users\Rahul Chembakasseril\cn>python3 SERVER.py
[STARTING] server is starting...
[LISTENING] The server is listening
[NEW CONNECTION] ('127.0.0.1', 58992) connected.
File Sent
[NEW CONNECTION] ('127.0.0.1', 59077) connected.
File Recieved
[NEW CONNECTION] ('127.0.0.1', 59345) connected.

C:\Users\Rahul Chembakasseril\cn>cd client
C:\Users\Rahul Chembakasseril\cn>cd client

C:\Users\Rahul Chembakasseril\cn>client>python3 CLIENT.py
Write your Message here: DWD<Test0.txt>
Downloaded
Download has taken: 49.13772s
Write your Message here: UPD<UploadText.txt>
Uploaded
Upload has taken: 54.73042s
Write your Message here:
```

Time Taken - 54.73042s

Observations-

We can observe that the time taken for File Transfer is the most in Windows which can be attributed to the fact that in Windows, many programs run in the background as compared to Kali Linux. Windows might have other protocols in place for packet transfer. This might lead to some extra overhead as compared to running code in Kali Linux.

As per my observations, my client-server File Transfer protocol runs slower in the Mininet terminal as compared to running the client-server in the Kali Linux terminal. Mininet is slower.

This result agrees with the fact that Mininet will be slower because it is an Emulator on top of a Virtual Box and it's default bandwidth value would have been set to a lower value than that of Kali Linux without Mininet. So the performance has decreased with Mininet.

Question 2

- a) To implement the custom topology, I used the template provided at
<http://mininet.org/walkthrough/>

The code has been uploaded to the GitHub repository available at :
https://github.com/RahulVC02/IITGN_Computer-Networks_Assignment-1

```
 1  from mininet.topo import Topo
 2  from mininet.link import TCLink
 3  from mininet.node import OVSController
 4
 5
 6  class MyTopo(Topo, TCLink, OVSController):
 7
 8      def __init__(self):
 9          Topo.build(self) #Initializing the Topology
10
11      #Adding the hosts
12
13      A = self.addHost('A')
14      B = self.addHost('B')
15      C = self.addHost('C')
16      D = self.addHost('D')
17
18      #Adding the switches
19
20      R1 = self.addSwitch('R1')
21      R2 = self.addSwitch('R2')
22
23      #Adding the links
24      L1 = self.addLink(A, R1, cls=TCLink, bw=1000, delay='1ms')
25      L5 = self.addLink(D, R1, cls=TCLink, bw=1000, delay='1ms')
26      L3 = self.addLink(B, R2, cls=TCLink, bw=1000, delay='1ms')
27      L4 = self.addLink(C, R2, cls=TCLink, bw=1000, delay='5ms')
28      L5 = self.addLink(R1, R2, cls=TCLink, bw=500, delay='10ms')
29
30
31  topos = {'mytopo': (lambda: MyTopo())}
```

- b) I used the ping command to measure the average RTT over transfer of 10 packets for the 5 pairs required.

1) AB

```
*** Starting CLI:  
mininet> A ping -c 10 B  
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.  
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=60.8 ms  
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=39.1 ms  
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=32.0 ms  
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=34.9 ms  
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=35.1 ms  
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=29.7 ms  
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=30.7 ms  
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=33.5 ms  
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=35.9 ms  
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=30.4 ms  
  
— 10.0.0.2 ping statistics —  
10 packets transmitted, 10 received, 0% packet loss, time 9016ms  
rtt min/avg/max/mdev = 29.697/36.201/60.778/8.646 ms  
mininet> █
```

Avg RTT = 36.201 ms

2) AC

```
mininet> A ping -c 10 C
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=73.9 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=37.9 ms
64 bytes from 10.0.0.3: icmp_seq=3 ttl=64 time=44.2 ms
64 bytes from 10.0.0.3: icmp_seq=4 ttl=64 time=41.9 ms
64 bytes from 10.0.0.3: icmp_seq=5 ttl=64 time=39.4 ms
64 bytes from 10.0.0.3: icmp_seq=6 ttl=64 time=36.5 ms
64 bytes from 10.0.0.3: icmp_seq=7 ttl=64 time=42.8 ms
64 bytes from 10.0.0.3: icmp_seq=8 ttl=64 time=36.9 ms
64 bytes from 10.0.0.3: icmp_seq=9 ttl=64 time=39.2 ms
64 bytes from 10.0.0.3: icmp_seq=10 ttl=64 time=41.3 ms

— 10.0.0.3 ping statistics —
10 packets transmitted, 10 received, 0% packet loss, time 9016ms
rtt min/avg/max/mdev = 36.491/43.403/73.931/10.454 ms
mininet> █
```

Avg RTT = 43.403 ms

3) AD

```
mininet> A ping -c 10 D
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=15.1 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=10.9 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=8.85 ms
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64 time=8.83 ms
64 bytes from 10.0.0.4: icmp_seq=5 ttl=64 time=8.52 ms
64 bytes from 10.0.0.4: icmp_seq=6 ttl=64 time=8.82 ms
64 bytes from 10.0.0.4: icmp_seq=7 ttl=64 time=10.4 ms
64 bytes from 10.0.0.4: icmp_seq=8 ttl=64 time=9.80 ms
64 bytes from 10.0.0.4: icmp_seq=9 ttl=64 time=9.91 ms
64 bytes from 10.0.0.4: icmp_seq=10 ttl=64 time=10.6 ms

— 10.0.0.4 ping statistics —
10 packets transmitted, 10 received, 0% packet loss, time 9015ms
rtt min/avg/max/mdev = 8.522/10.176/15.132/1.834 ms
mininet> █
```

Avg RTT = 10.176 ms

4) BC

```
mininet> B ping -c 10 C
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=29.7 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=14.1 ms
64 bytes from 10.0.0.3: icmp_seq=3 ttl=64 time=13.8 ms
64 bytes from 10.0.0.3: icmp_seq=4 ttl=64 time=14.5 ms
64 bytes from 10.0.0.3: icmp_seq=5 ttl=64 time=14.4 ms
64 bytes from 10.0.0.3: icmp_seq=6 ttl=64 time=12.6 ms
64 bytes from 10.0.0.3: icmp_seq=7 ttl=64 time=14.5 ms
64 bytes from 10.0.0.3: icmp_seq=8 ttl=64 time=14.1 ms
64 bytes from 10.0.0.3: icmp_seq=9 ttl=64 time=14.2 ms
64 bytes from 10.0.0.3: icmp_seq=10 ttl=64 time=14.5 ms

— 10.0.0.3 ping statistics —
10 packets transmitted, 10 received, 0% packet loss, time 9013ms
rtt min/avg/max/mdev = 12.619/15.636/29.734/4.728 ms
mininet> █
```

Avg RTT = 15.636 ms

5) BD

```
mininet> B ping -c 10 D
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=62.6 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=31.2 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=28.7 ms
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64 time=32.9 ms
64 bytes from 10.0.0.4: icmp_seq=5 ttl=64 time=30.8 ms
64 bytes from 10.0.0.4: icmp_seq=6 ttl=64 time=34.0 ms
64 bytes from 10.0.0.4: icmp_seq=7 ttl=64 time=33.1 ms
64 bytes from 10.0.0.4: icmp_seq=8 ttl=64 time=32.4 ms
64 bytes from 10.0.0.4: icmp_seq=9 ttl=64 time=31.8 ms
64 bytes from 10.0.0.4: icmp_seq=10 ttl=64 time=33.2 ms

— 10.0.0.4 ping statistics —
10 packets transmitted, 10 received, 0% packet loss, time 9019ms
rtt min/avg/max/mdev = 28.666/35.060/62.611/9.296 ms
mininet> █
```

Avg RTT = 35.060 ms

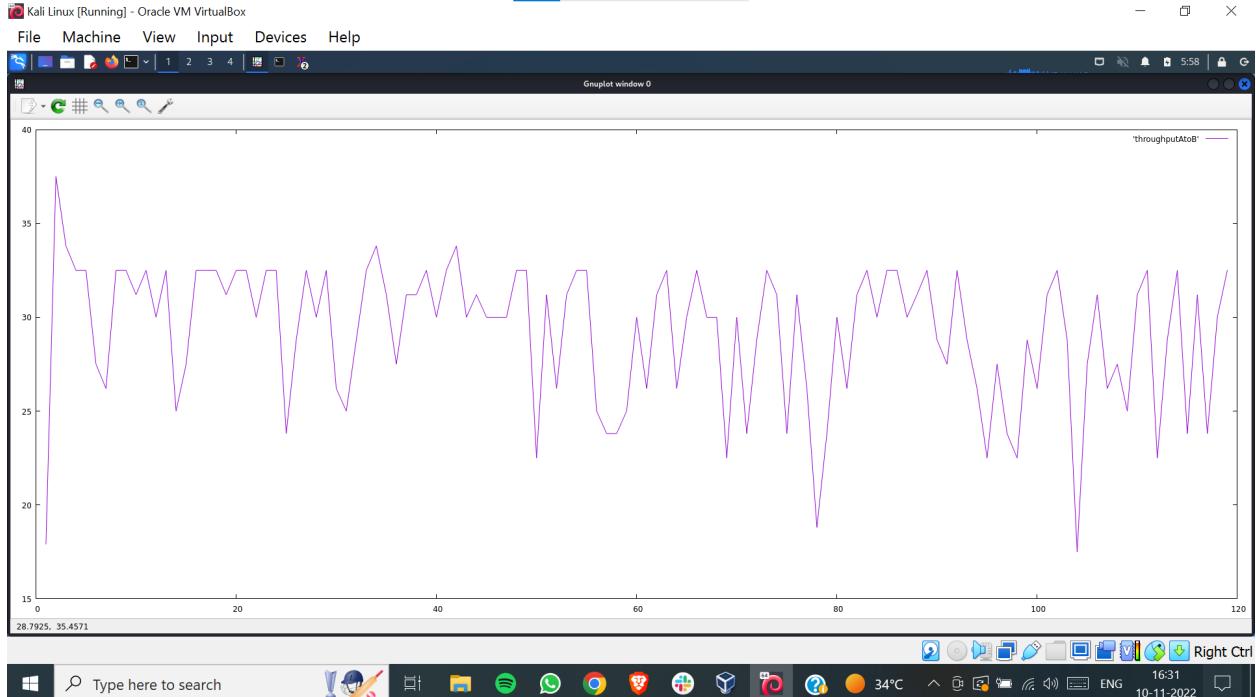
c) I ran the iperf client on B and the iperf server on A. The throughput is plotted using the gnuplot package. I measured the throughput over 3 iterations and 120 seconds for each iteration.

Iteration 1

```

Accepted connection from 10.0.0.2, port 47036
[ 7] local 10.0.0.1 port 5201 connected to 10.0.0.2 port 47038
[ ID] Interval Transfer Bitrate
[ 7] 0.00-1.00 sec 13.5 MBytes 113 Mbits/sec
[ 7] 1.00-2.00 sec 32.0 MBytes 263 Mbits/sec
[ 7] 2.00-3.00 sec 32.5 MBytes 276 Mbits/sec
[ 7] 3.00-4.00 sec 32.5 MBytes 274 Mbits/sec
[ 7] 4.00-5.00 sec 32.0 MBytes 268 Mbits/sec
[ 7] 5.00-6.00 sec 27.8 MBytes 233 Mbits/sec
[ 7] 6.00-7.00 sec 25.9 MBytes 217 Mbits/sec
[ 7] 7.00-8.00 sec 32.3 MBytes 271 Mbits/sec
[ 7] 8.00-9.00 sec 32.5 MBytes 273 Mbits/sec
[ 7] 9.00-10.00 sec 31.6 MBytes 265 Mbits/sec
[ 7] 10.00-11.00 sec 32.1 MBytes 269 Mbits/sec
[ 7] 11.00-12.00 sec 32.0 MBytes 266 Mbits/sec
[ 7] 12.00-13.00 sec 31.4 MBytes 264 Mbits/sec
[ 7] 13.00-14.00 sec 25.5 MBytes 214 Mbits/sec
[ 7] 14.00-15.00 sec 27.4 MBytes 230 Mbits/sec
[ 7] 15.00-16.00 sec 32.2 MBytes 270 Mbits/sec
[ 7] 16.00-17.00 sec 32.9 MBytes 276 Mbits/sec
[ 7] 17.00-18.00 sec 32.3 MBytes 271 Mbits/sec
[ 7] 18.00-19.00 sec 31.9 MBytes 268 Mbits/sec
[ 7] 19.00-20.00 sec 32.3 MBytes 271 Mbits/sec
[ 7] 20.00-21.00 sec 32.0 MBytes 273 Mbits/sec
[ 7] 21.00-22.00 sec 29.9 MBytes 251 Mbits/sec
[ 7] 22.00-23.00 sec 32.6 MBytes 274 Mbits/sec
[ 7] 23.00-24.00 sec 31.9 MBytes 267 Mbits/sec
[ 7] 24.00-25.00 sec 24.5 MBytes 206 Mbits/sec
[ 7] 25.00-26.00 sec 28.1 MBytes 235 Mbits/sec
[ 7] 26.00-27.00 sec 32.7 MBytes 274 Mbits/sec
[ 7] 27.00-28.00 sec 30.4 MBytes 255 Mbits/sec
[ 7] 28.00-29.00 sec 32.1 MBytes 263 Mbits/sec
[ 7] 29.00-30.00 sec 26.8 MBytes 226 Mbits/sec
[ 7] 30.00-31.00 sec 24.1 MBytes 208 Mbits/sec
[ 7] 31.00-32.00 sec 28.3 MBytes 238 Mbits/sec
[ 7] 32.00-33.00 sec 33.1 MBytes 278 Mbits/sec
[ 7] 33.00-34.00 sec 33.1 MBytes 277 Mbits/sec
[ 7] 34.00-35.00 sec 31.5 MBytes 264 Mbits/sec
[ 7] 35.00-36.00 sec 27.9 MBytes 234 Mbits/sec
[ 7] 36.00-37.00 sec 30.6 MBytes 257 Mbits/sec
[ 7] 37.00-38.00 sec 31.2 MBytes 262 Mbits/sec
[ 7] 38.00-39.00 sec 32.0 MBytes 270 Mbits/sec
[ 7] 39.00-40.00 sec 30.3 MBytes 254 Mbits/sec
[ 7] 40.00-41.00 sec 32.4 MBytes 272 Mbits/sec
[ 7] 41.00-42.00 sec 33.9 MBytes 284 Mbits/sec
[ 7] 42.00-43.00 sec 30.2 MBytes 253 Mbits/sec
[ 7] 43.00-44.00 sec 31.2 MBytes 261 Mbits/sec
[ 7] 44.00-45.00 sec 29.5 MBytes 248 Mbits/sec
[ 7] 45.00-46.00 sec 30.7 MBytes 257 Mbits/sec
[ 7] 46.00-47.00 sec 29.7 MBytes 250 Mbits/sec
[ 7] 47.00-48.00 sec 32.0 MBytes 271 Mbits/sec
[ 7] 48.00-49.00 sec 32.3 MBytes 274 Mbits/sec
[ 7] 49.00-50.00 sec 23.0 MBytes 193 Mbits/sec
[ 7] 50.00-51.00 sec 30.7 MBytes 257 Mbits/sec
[ 7] 51.00-52.00 sec 27.0 MBytes 227 Mbits/sec
[ 7] 52.00-53.00 sec 31.1 MBytes 261 Mbits/sec
[ 7] 53.00-54.00 sec 32.8 MBytes 275 Mbits/sec
[ 7] 54.00-55.00 sec 31.7 MBytes 266 Mbits/sec

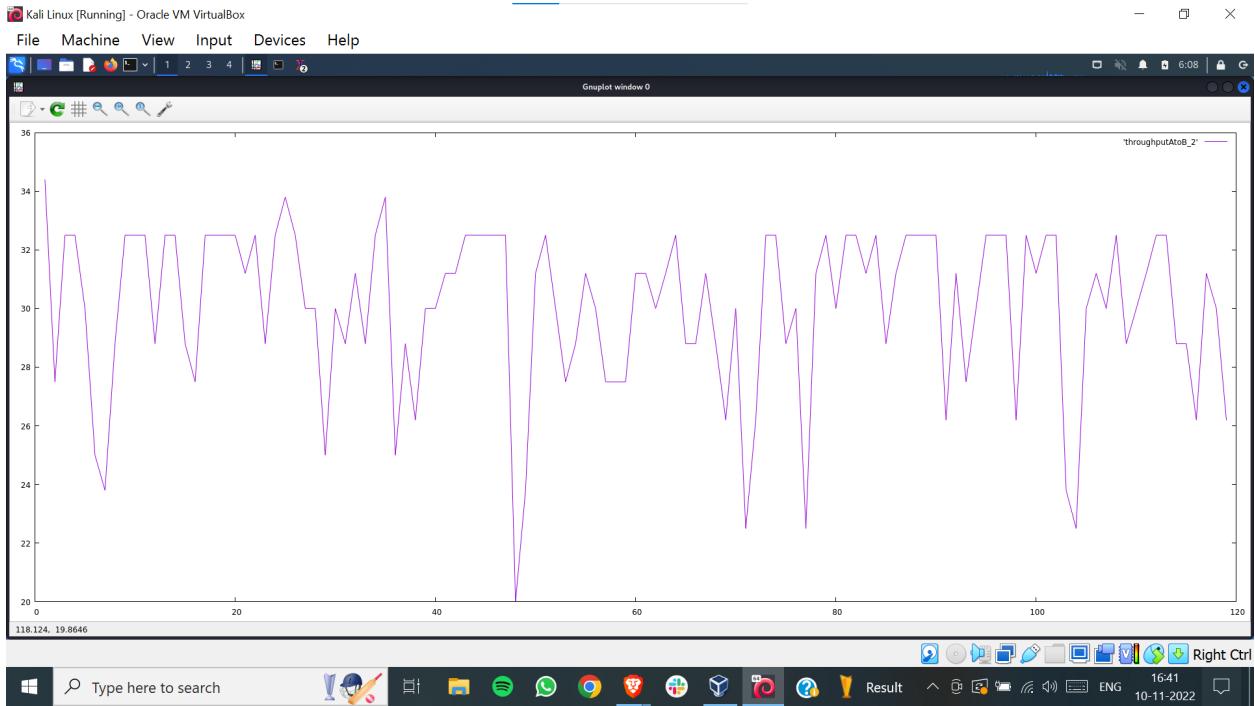
```



Avg Throughput = 0.0283 GBps

Iteration 2

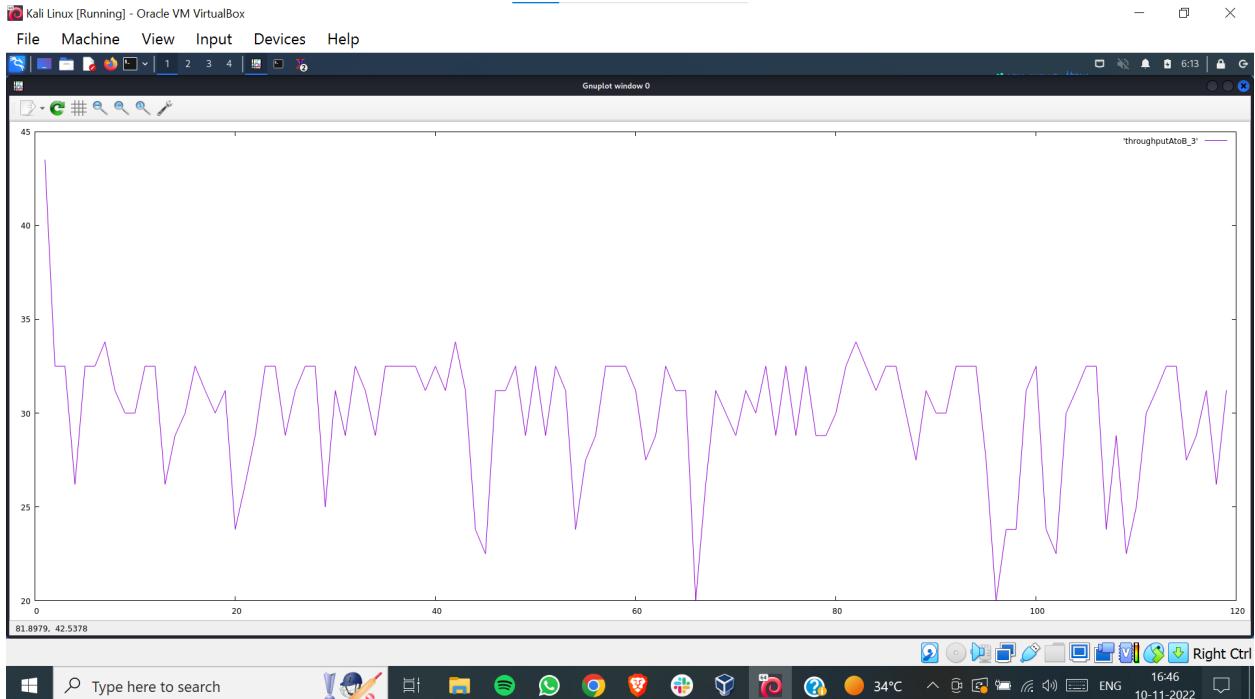
```
[ 7] local 10.0.0.1 port 5201 connected to 10.0.0.2 port 56894
[ 7] Interval Transfer Bitrate
[ 7] 0.00-1.00 sec 23.7 MBytes 199 Mbits/sec
[ 7] 1.00-2.00 sec 26.7 MBytes 224 Mbits/sec
[ 7] 2.00-3.00 sec 32.7 MBytes 275 Mbits/sec
[ 7] 3.00-4.00 sec 32.8 MBytes 275 Mbits/sec
[ 7] 4.00-5.00 sec 29.5 MBytes 248 Mbits/sec
[ 7] 5.00-6.00 sec 25.0 MBytes 210 Mbits/sec
[ 7] 6.00-7.00 sec 23.8 MBytes 200 Mbits/sec
[ 7] 7.00-8.00 sec 29.5 MBytes 247 Mbits/sec
[ 7] 8.00-9.00 sec 32.7 MBytes 274 Mbits/sec
[ 7] 9.00-10.00 sec 32.4 MBytes 272 Mbits/sec
[ 7] 10.00-11.00 sec 32.5 MBytes 273 Mbits/sec
[ 7] 11.00-12.00 sec 28.5 MBytes 239 Mbits/sec
[ 7] 12.00-13.00 sec 32.1 MBytes 270 Mbits/sec
[ 7] 13.00-14.00 sec 32.6 MBytes 273 Mbits/sec
[ 7] 14.00-15.00 sec 28.7 MBytes 240 Mbits/sec
[ 7] 15.00-16.00 sec 28.1 MBytes 235 Mbits/sec
[ 7] 16.00-17.00 sec 32.0 MBytes 268 Mbits/sec
[ 7] 17.00-18.00 sec 32.8 MBytes 275 Mbits/sec
[ 7] 18.00-19.00 sec 32.8 MBytes 275 Mbits/sec
[ 7] 19.00-20.00 sec 32.1 MBytes 270 Mbits/sec
[ 7] 20.00-21.00 sec 30.8 MBytes 258 Mbits/sec
[ 7] 21.00-22.00 sec 32.4 MBytes 272 Mbits/sec
[ 7] 22.00-23.00 sec 29.7 MBytes 249 Mbits/sec
[ 7] 23.00-24.00 sec 32.2 MBytes 270 Mbits/sec
[ 7] 24.00-25.00 sec 33.0 MBytes 277 Mbits/sec
[ 7] 25.00-26.00 sec 32.4 MBytes 272 Mbits/sec
[ 7] 26.00-27.00 sec 30.5 MBytes 256 Mbits/sec
[ 7] 27.00-28.00 sec 29.6 MBytes 248 Mbits/sec
[ 7] 28.00-29.00 sec 25.8 MBytes 216 Mbits/sec
[ 7] 29.00-30.00 sec 29.9 MBytes 251 Mbits/sec
[ 7] 30.00-31.00 sec 28.9 MBytes 242 Mbits/sec
[ 7] 31.00-32.00 sec 31.0 MBytes 261 Mbits/sec
[ 7] 32.00-33.00 sec 27.9 MBytes 234 Mbits/sec
[ 7] 33.00-34.00 sec 33.6 MBytes 281 Mbits/sec
[ 7] 34.00-35.00 sec 33.5 MBytes 282 Mbits/sec
[ 7] 35.00-36.00 sec 24.8 MBytes 208 Mbits/sec
[ 7] 36.00-37.00 sec 28.6 MBytes 240 Mbits/sec
[ 7] 37.00-38.00 sec 26.7 MBytes 224 Mbits/sec
[ 7] 38.00-39.00 sec 29.4 MBytes 247 Mbits/sec
[ 7] 39.00-40.00 sec 30.2 MBytes 253 Mbits/sec
[ 7] 40.00-41.00 sec 30.7 MBytes 258 Mbits/sec
[ 7] 41.00-42.00 sec 31.8 MBytes 266 Mbits/sec
[ 7] 42.00-43.00 sec 32.4 MBytes 271 Mbits/sec
```



Avg Throughput = 0.0302 GBps

Iteration 3

```
[ 7] local 10.0.0.1 port 5201 connected to 10.0.0.2 port 55446
[ ID] Interval Transfer Bitrate
[ 7] 0.00-1.00 sec 32.4 MBytes 272 Mbytes/sec
[ 7] 1.00-2.00 sec 32.7 MBytes 275 Mbytes/sec
[ 7] 2.00-3.00 sec 32.4 MBytes 271 Mbytes/sec
[ 7] 3.00-4.00 sec 26.8 MBytes 225 Mbytes/sec
[ 7] 4.00-5.00 sec 32.3 MBytes 271 Mbytes/sec
[ 7] 5.00-6.00 sec 32.7 MBytes 274 Mbytes/sec
[ 7] 6.00-7.00 sec 32.8 MBytes 275 Mbytes/sec
[ 7] 7.00-8.00 sec 32.4 MBytes 272 Mbytes/sec
[ 7] 8.00-9.00 sec 29.6 MBytes 249 Mbytes/sec
[ 7] 9.00-10.00 sec 29.6 MBytes 248 Mbytes/sec
[ 7] 10.00-11.00 sec 32.4 MBytes 272 Mbytes/sec
[ 7] 11.00-12.00 sec 32.7 MBytes 275 Mbytes/sec
[ 7] 12.00-13.00 sec 26.0 MBytes 218 Mbytes/sec
[ 7] 13.00-14.00 sec 29.2 MBytes 245 Mbytes/sec
[ 7] 14.00-15.00 sec 29.6 MBytes 248 Mbytes/sec
[ 7] 15.00-16.00 sec 32.9 MBytes 276 Mbytes/sec
[ 7] 16.00-17.00 sec 31.1 MBytes 261 Mbytes/sec
[ 7] 17.00-18.00 sec 30.2 MBytes 253 Mbytes/sec
[ 7] 18.00-19.00 sec 31.2 MBytes 262 Mbytes/sec
[ 7] 19.00-20.00 sec 23.6 MBytes 198 Mbytes/sec
[ 7] 20.00-21.00 sec 26.2 MBytes 219 Mbytes/sec
[ 7] 21.00-22.00 sec 29.3 MBytes 246 Mbytes/sec
[ 7] 22.00-23.00 sec 32.1 MBytes 269 Mbytes/sec
[ 7] 23.00-24.00 sec 32.2 MBytes 270 Mbytes/sec
[ 7] 24.00-25.00 sec 29.1 MBytes 244 Mbytes/sec
[ 7] 25.00-26.00 sec 30.4 MBytes 255 Mbytes/sec
[ 7] 26.00-27.00 sec 32.7 MBytes 274 Mbytes/sec
[ 7] 27.00-28.00 sec 32.6 MBytes 273 Mbytes/sec
[ 7] 28.00-29.00 sec 25.3 MBytes 212 Mbytes/sec
[ 7] 29.00-30.00 sec 31.0 MBytes 260 Mbytes/sec
[ 7] 30.00-31.00 sec 29.2 MBytes 245 Mbytes/sec
[ 7] 31.00-32.00 sec 32.2 MBytes 270 Mbytes/sec
[ 7] 32.00-33.00 sec 30.9 MBytes 259 Mbytes/sec
[ 7] 33.00-34.00 sec 29.1 MBytes 244 Mbytes/sec
[ 7] 34.00-35.00 sec 32.4 MBytes 272 Mbytes/sec
[ 7] 35.00-36.00 sec 32.7 MBytes 274 Mbytes/sec
[ 7] 36.00-37.00 sec 32.4 MBytes 272 Mbytes/sec
[ 7] 37.00-38.00 sec 32.1 MBytes 269 Mbytes/sec
[ 7] 38.00-39.00 sec 31.6 MBytes 265 Mbytes/sec
```



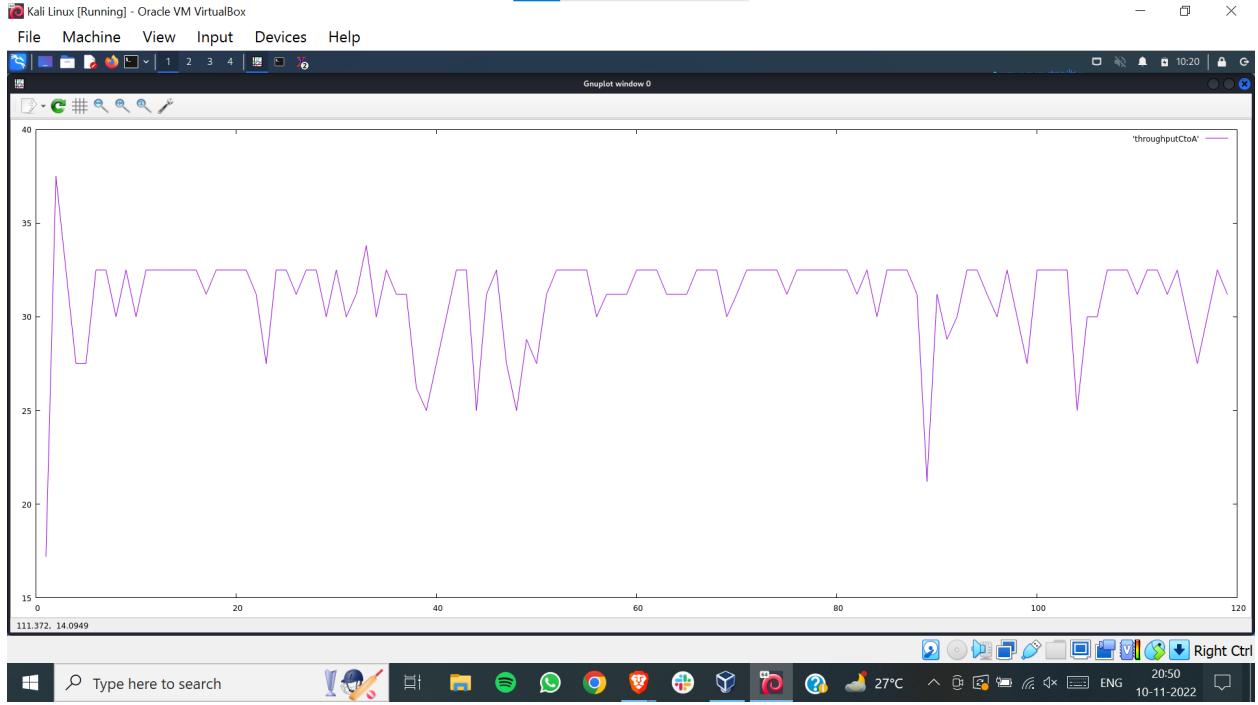
Avg Throughput = 0.031 GBps

So, the average overall throughput over the 3 iterations is 0.029833 GBps

d) In this case, I ran the iperf server on host A and the iperf client on host C. The throughput is plotted using the gnuplot package. I measured the throughput over 3 iterations and 120 seconds for each iteration.

Iteration 1

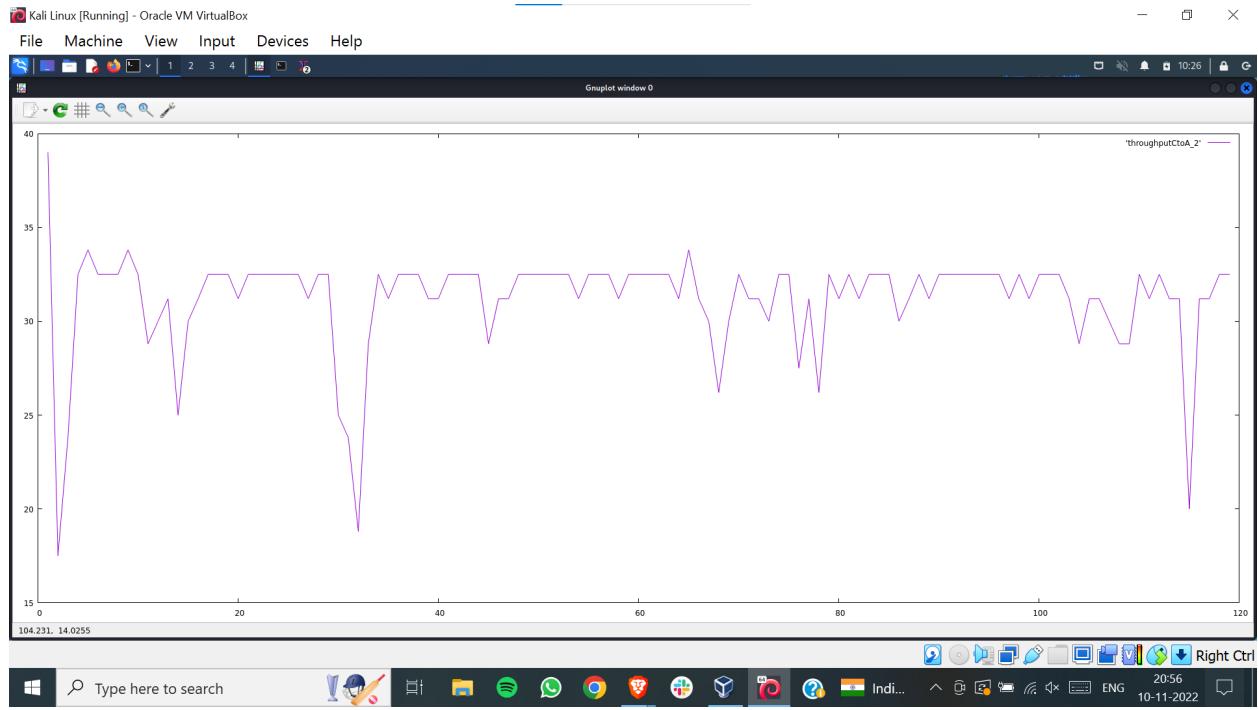
```
[ 7] local 10.0.0.1 port 5201 connected to 10.0.0.3 port 36822
[ ID] Interval      Transfer     Bitrate
[ 7]  0.00-1.00   sec  12.9 MBytes  108 Mbits/sec
[ 7]  1.00-2.00   sec  32.2 MBytes  270 Mbits/sec
[ 7]  2.00-3.00   sec  33.0 MBytes  277 Mbits/sec
[ 7]  3.00-4.00   sec  27.8 MBytes  233 Mbits/sec
[ 7]  4.00-5.00   sec  27.6 MBytes  231 Mbits/sec
[ 7]  5.00-6.00   sec  32.0 MBytes  269 Mbits/sec
[ 7]  6.00-7.00   sec  32.2 MBytes  270 Mbits/sec
[ 7]  7.00-8.00   sec  29.7 MBytes  249 Mbits/sec
[ 7]  8.00-9.00   sec  32.3 MBytes  270 Mbits/sec
[ 7]  9.00-10.00  sec  29.9 MBytes  250 Mbits/sec
[ 7] 10.00-11.00  sec  32.2 MBytes  270 Mbits/sec
[ 7] 11.00-12.00  sec  32.4 MBytes  271 Mbits/sec
[ 7] 12.00-13.00  sec  32.4 MBytes  272 Mbits/sec
[ 7] 13.00-14.00  sec  32.1 MBytes  269 Mbits/sec
[ 7] 14.00-15.00  sec  32.3 MBytes  271 Mbits/sec
[ 7] 15.00-16.00  sec  32.2 MBytes  270 Mbits/sec
[ 7] 16.00-17.00  sec  32.3 MBytes  270 Mbits/sec
[ 7] 17.00-18.00  sec  32.4 MBytes  272 Mbits/sec
[ 7] 18.00-19.00  sec  32.3 MBytes  271 Mbits/sec
[ 7] 19.00-20.00  sec  32.2 MBytes  270 Mbits/sec
[ 7] 20.00-21.00  sec  32.3 MBytes  271 Mbits/sec
[ 7] 21.00-22.00  sec  31.9 MBytes  268 Mbits/sec
[ 7] 22.00-23.00  sec  27.4 MBytes  230 Mbits/sec
[ 7] 23.00-24.00  sec  32.1 MBytes  269 Mbits/sec
[ 7] 24.00-25.00  sec  32.4 MBytes  272 Mbits/sec
[ 7] 25.00-26.00  sec  32.1 MBytes  269 Mbits/sec
[ 7] 26.00-27.00  sec  32.4 MBytes  272 Mbits/sec
[ 7] 27.00-28.00  sec  32.2 MBytes  270 Mbits/sec
[ 7] 28.00-29.00  sec  30.0 MBytes  251 Mbits/sec
[ 7] 29.00-30.00  sec  32.2 MBytes  270 Mbits/sec
[ 7] 30.00-31.00  sec  29.7 MBytes  249 Mbits/sec
[ 7] 31.00-32.00  sec  32.3 MBytes  271 Mbits/sec
[ 7] 32.00-33.00  sec  32.8 MBytes  275 Mbits/sec
[ 7] 33.00-34.00  sec  30.4 MBytes  255 Mbits/sec
```



Avg Throughput = 0.0327 GBps

Iteration 2

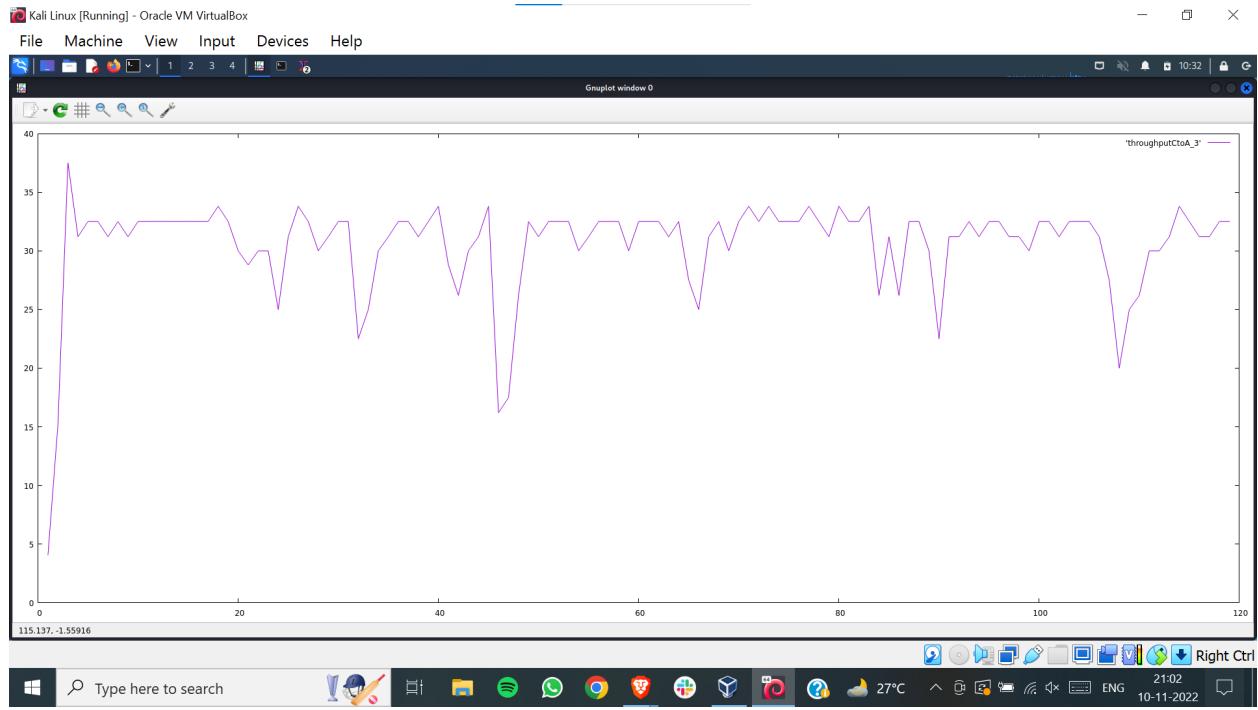
```
[ 7] local 10.0.0.1 port 5201 connected to 10.0.0.3 port 57986
[ ID] Interval Transfer Bitrate
[ 7] 0.00-1.00 sec 27.5 MBytes 231 Mbits/sec
[ 7] 1.00-2.00 sec 18.4 MBytes 154 Mbits/sec
[ 7] 2.00-3.00 sec 23.6 MBytes 198 Mbits/sec
[ 7] 3.00-4.00 sec 32.7 MBytes 275 Mbits/sec
[ 7] 4.00-5.00 sec 32.8 MBytes 276 Mbits/sec
[ 7] 5.00-6.00 sec 32.8 MBytes 275 Mbits/sec
[ 7] 6.00-7.00 sec 32.8 MBytes 275 Mbits/sec
[ 7] 7.00-8.00 sec 33.1 MBytes 277 Mbits/sec
[ 7] 8.00-9.00 sec 32.5 MBytes 274 Mbits/sec
[ 7] 9.00-10.00 sec 32.4 MBytes 272 Mbits/sec
[ 7] 10.00-11.00 sec 30.1 MBytes 252 Mbits/sec
[ 7] 11.00-12.00 sec 29.8 MBytes 250 Mbits/sec
[ 7] 12.00-13.00 sec 30.3 MBytes 254 Mbits/sec
[ 7] 13.00-14.00 sec 25.3 MBytes 212 Mbits/sec
[ 7] 14.00-15.00 sec 30.4 MBytes 255 Mbits/sec
[ 7] 15.00-16.00 sec 31.0 MBytes 261 Mbits/sec
[ 7] 16.00-17.00 sec 32.3 MBytes 271 Mbits/sec
[ 7] 17.00-18.00 sec 32.5 MBytes 272 Mbits/sec
[ 7] 18.00-19.00 sec 32.2 MBytes 270 Mbits/sec
[ 7] 19.00-20.00 sec 32.3 MBytes 271 Mbits/sec
[ 7] 20.00-21.00 sec 32.4 MBytes 272 Mbits/sec
[ 7] 21.00-22.00 sec 32.5 MBytes 273 Mbits/sec
[ 7] 22.00-23.00 sec 32.3 MBytes 270 Mbits/sec
[ 7] 23.00-24.00 sec 32.2 MBytes 270 Mbits/sec
[ 7] 24.00-25.00 sec 32.4 MBytes 272 Mbits/sec
[ 7] 25.00-26.00 sec 32.1 MBytes 269 Mbits/sec
[ 7] 26.00-27.00 sec 32.3 MBytes 271 Mbits/sec
[ 7] 27.00-28.00 sec 32.2 MBytes 270 Mbits/sec
[ 7] 28.00-29.00 sec 32.1 MBytes 269 Mbits/sec
[ 7] 29.00-30.00 sec 25.4 MBytes 213 Mbits/sec
[ 7] 30.00-31.00 sec 23.6 MBytes 198 Mbits/sec
[ 7] 31.00-32.00 sec 18.5 MBytes 155 Mbits/sec
[ 7] 32.00-33.00 sec 28.8 MBytes 242 Mbits/sec
[ 7] 33.00-34.00 sec 32.2 MBytes 270 Mbits/sec
[ 7] 34.00-35.00 sec 32.1 MBytes 270 Mbits/sec
[ 7] 35.00-36.00 sec 32.4 MBytes 271 Mbits/sec
[ 7] 36.00-37.00 sec 32.4 MBytes 272 Mbits/sec
```



Avg Throughput = 0.0317 GBps

Iteration 3

```
Accepted connection from 10.0.0.3, port 35580
[ 7] local 10.0.0.1 port 5201 connected to 10.0.0.3 port 35596
[ 7] Interval           Transfer     Bitrate
[ 7]  0.00-1.00   sec  2.98 MBytes  25.0 Mbits/sec
[ 7]  1.00-2.00   sec 12.8 MBytes 108 Mbits/sec
[ 7]  2.00-3.00   sec 31.5 MBytes 264 Mbits/sec
[ 7]  3.00-4.00   sec 30.2 MBytes 253 Mbits/sec
[ 7]  4.00-5.00   sec 32.0 MBytes 268 Mbits/sec
[ 7]  5.00-6.00   sec 32.3 MBytes 271 Mbits/sec
[ 7]  6.00-7.00   sec 32.0 MBytes 269 Mbits/sec
[ 7]  7.00-8.00   sec 32.4 MBytes 272 Mbits/sec
[ 7]  8.00-9.00   sec 30.9 MBytes 260 Mbits/sec
[ 7]  9.00-10.00  sec 33.0 MBytes 277 Mbits/sec
[ 7] 10.00-11.00  sec 32.4 MBytes 271 Mbits/sec
[ 7] 11.00-12.00  sec 32.5 MBytes 273 Mbits/sec
[ 7] 12.00-13.00  sec 31.8 MBytes 266 Mbits/sec
[ 7] 13.00-14.00  sec 32.8 MBytes 276 Mbits/sec
[ 7] 14.00-15.00  sec 32.4 MBytes 272 Mbits/sec
[ 7] 15.00-16.00  sec 32.5 MBytes 273 Mbits/sec
[ 7] 16.00-17.00  sec 32.8 MBytes 275 Mbits/sec
[ 7] 17.00-18.00  sec 33.6 MBytes 282 Mbits/sec
[ 7] 18.00-19.00  sec 32.6 MBytes 274 Mbits/sec
[ 7] 19.00-20.00  sec 29.6 MBytes 249 Mbits/sec
[ 7] 20.00-21.00  sec 29.4 MBytes 247 Mbits/sec
[ 7] 21.00-22.00  sec 29.7 MBytes 250 Mbits/sec
[ 7] 22.00-23.00  sec 30.0 MBytes 251 Mbits/sec
[ 7] 23.00-24.00  sec 25.2 MBytes 212 Mbits/sec
[ 7] 24.00-25.00  sec 31.3 MBytes 263 Mbits/sec
[ 7] 25.00-26.00  sec 33.8 MBytes 284 Mbits/sec
[ 7] 26.00-27.00  sec 31.7 MBytes 265 Mbits/sec
[ 7] 27.00-28.00  sec 29.7 MBytes 250 Mbits/sec
[ 7] 28.00-29.00  sec 32.0 MBytes 263 Mbits/sec
[ 7] 29.00-30.00  sec 32.8 MBytes 275 Mbits/sec
[ 7] 30.00-31.00  sec 32.4 MBytes 271 Mbits/sec
[ 7] 31.00-32.00  sec 22.4 MBytes 188 Mbits/sec
[ 7] 32.00-33.00  sec 24.5 MBytes 205 Mbits/sec
[ 7] 33.00-34.00  sec 30.8 MBytes 258 Mbits/sec
```



So, the average throughput over the 3 iterations is 0.032233 GBps

Observations-

We would expect that the average throughput should decrease when there is a delay in the link, however the difference is hardly noticeable and in fact, to the contrary, the throughput has increased slightly.

This may be because the current testing conditions don't capture the effect of the increased delay as well. If we set the bandwidth between R1 and R2 sufficiently lower, then we might observe a noticeable difference in the throughput values. If the iteration interval is reduced to lesser than the current value (1second), then we may observe lower throughput in initial iterations after which it may stabilize.

e) In this scenario, I ran the iperf client on hosts B and C, the iperf server on host A. I measured the throughput over 3 iterations and 120 seconds for each iteration.

I ran the server on Node A on 2 different ports servicing 2 different clients B and C.

Iteration 1

At Node A

```
[root@rahulc28 ~]# iperf3 -s -p 4040 & iperf3 -s -p 5050
[1] 70462
-----
Server listening on 4040 (test #1)
-----
Server listening on 5050 (test #1)
Accepted connection from 10.0.0.3, port 52564
[ 7] local 10.0.0.1 port 5050 connected to 10.0.0.3 port 52572
[ 10] Interval Transfer Bitrate
[ 7] 0.00-1.00 sec 2.33 MBytes 19.6 Mbits/sec
[ 7] 1.00-2.00 sec 9.23 MBytes 77.5 Mbits/sec
Accepted connection from 10.0.0.2, port 39730
[ 7] local 10.0.0.1 port 4040 connected to 10.0.0.2 port 39740
[ 7] 2.00-3.00 sec 20.0 MBytes 168 Mbits/sec
[ 10] Interval Transfer Bitrate
[ 7] 0.00-1.00 sec 2.04 MBytes 17.1 Mbits/sec
[ 7] 3.00-4.00 sec 27.7 MBytes 232 Mbits/sec
[ 7] 1.00-2.00 sec 2.34 MBytes 19.8 Mbits/sec
[ 7] 4.00-5.00 sec 26.0 MBytes 218 Mbits/sec
[ 7] 2.00-3.00 sec 1.47 MBytes 12.4 Mbits/sec
[ 7] 5.00-6.00 sec 24.4 MBytes 205 Mbits/sec
[ 7] 3.00-4.00 sec 1.38 MBytes 11.5 Mbits/sec
[ 7] 6.00-7.00 sec 27.9 MBytes 234 Mbits/sec
[ 7] 4.00-5.00 sec 998 KBytes 8.18 Mbits/sec
[ 7] 7.00-8.00 sec 32.3 MBytes 271 Mbits/sec
[ 7] 5.00-6.00 sec 1.01 MBytes 8.48 Mbits/sec
[ 7] 8.00-9.00 sec 22.6 MBytes 190 Mbits/sec
[ 7] 6.00-7.00 sec 885 KBytes 7.27 Mbits/sec
[ 7] 9.00-10.00 sec 25.5 MBytes 214 Mbits/sec
[ 7] 7.00-8.00 sec 1.06 MBytes 8.93 Mbits/sec
[ 7] 10.00-11.00 sec 23.8 MBytes 200 Mbits/sec
[ 7] 8.00-9.00 sec 1.18 MBytes 9.88 Mbits/sec
[ 7] 11.00-12.00 sec 22.0 MBytes 185 Mbits/sec
[ 7] 9.00-10.00 sec 1.58 MBytes 11.6 Mbits/sec
```

At Node B

```
[root@rahulc28 ~]# iperf3 -t120 -p 4040 -c 10.0.0.1 > result_ba_conc
[root@rahulc28 ~]# cat result_ba_conc | grep sec | head -119 | tr " " " | awk '{print $4, $6}' > throughput_ba_conc
[root@rahulc28 ~]# gnuplot
G N U P L O T
Version 5.4 patchlevel 4    last modified 2022-07-10
Copyright (C) 1986-1993, 1998, 2004, 2007-2022
Thomas Williams, Colin Kelley and many others
gnuplot home:   http://www.gnuplot.info
faq, bugs, etc: type "help FAQ"
immediate help: type "help" (plot window: hit 'h')
Terminal type is now 'qt'
gnuplot> plot 'throughput_ba_conc' w l
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
gnuplot> █
```

At Node C

```

└─(root@rahulc28)─[~/home/rahulc28]
  * iperf3 -t120 -p 5050 -c 10.0.0.1 > result_ca_conc

└─(root@rahulc28)─[~/home/rahulc28]
  * cat result_ca_conc | grep sec | head -119 | tr - " " | awk '{print $4, $6}' > throughput_ca_conc

└─(root@rahulc28)─[~/home/rahulc28]
  * gnuplot

    G N U P L O T
    Version 5.4 patchlevel 4      last modified 2022-07-10

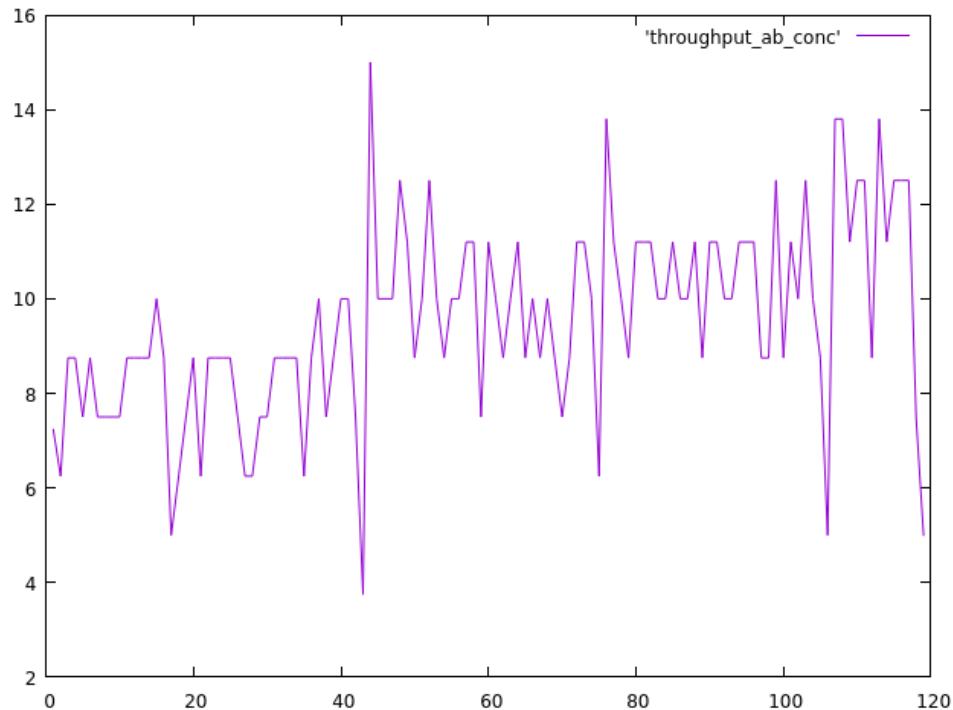
    Copyright (C) 1986-1993, 1998, 2004, 2007-2022
    Thomas Williams, Colin Kelley and many others

    gnuplot home:      http://www.gnuplot.info
    faq, bugs, etc:   type "help FAQ"
    immediate help:   type "help" (plot window; hit 'h')

Terminal type is now 'qt'
gnuplot> plot 'throughput_ca_conc' w l
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
gnuplot> █

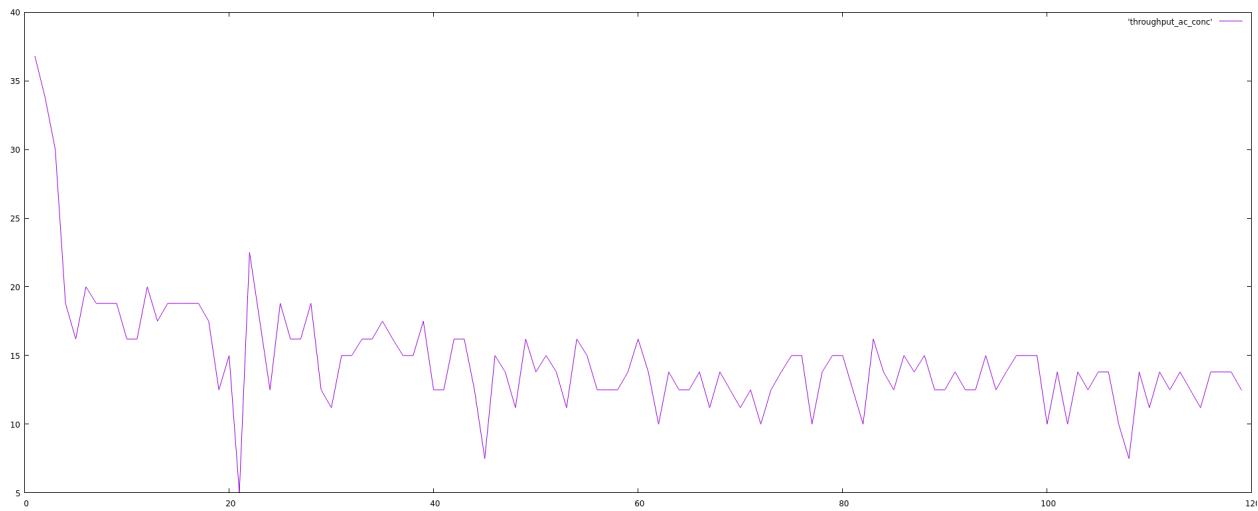
```

Throughput Plot for Node B -



Avg Throughput for B = 0.0105 GBps

Throughput Plot for Node C -

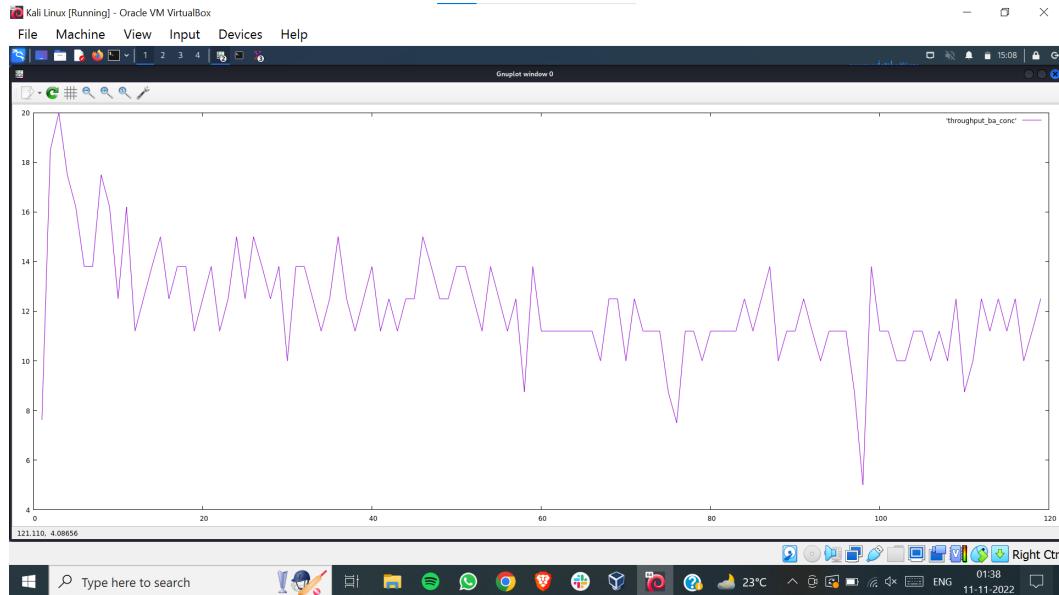


Avg Throughput for C = 0.0134 GBps

Iteration 2

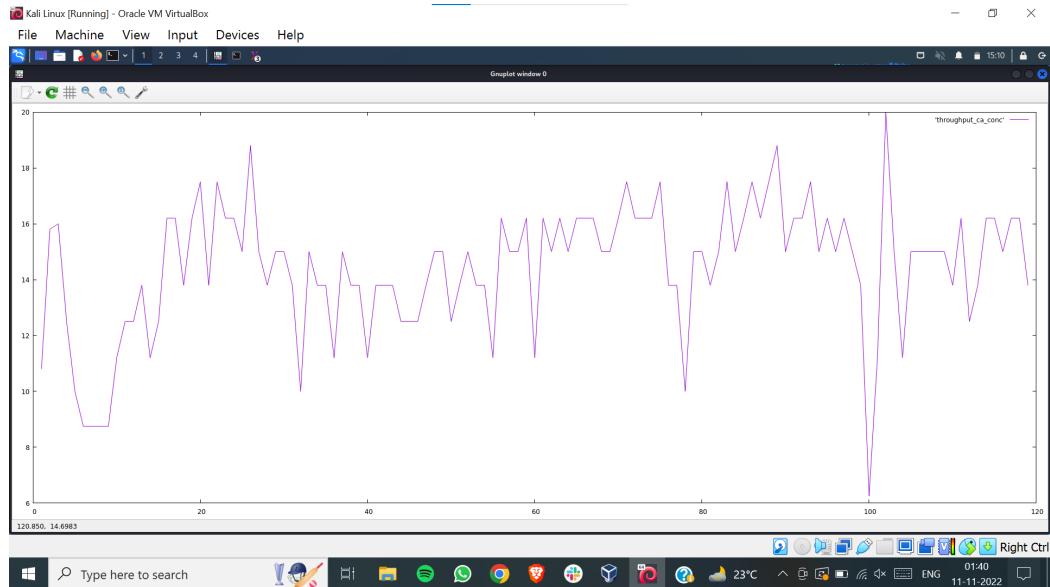
The commands run in iteration 2 were same as iteration 1.

Throughput Plot for Node B-



Avg Throughput for B = 0.0124 GBps

Throughput Plot for Node C-

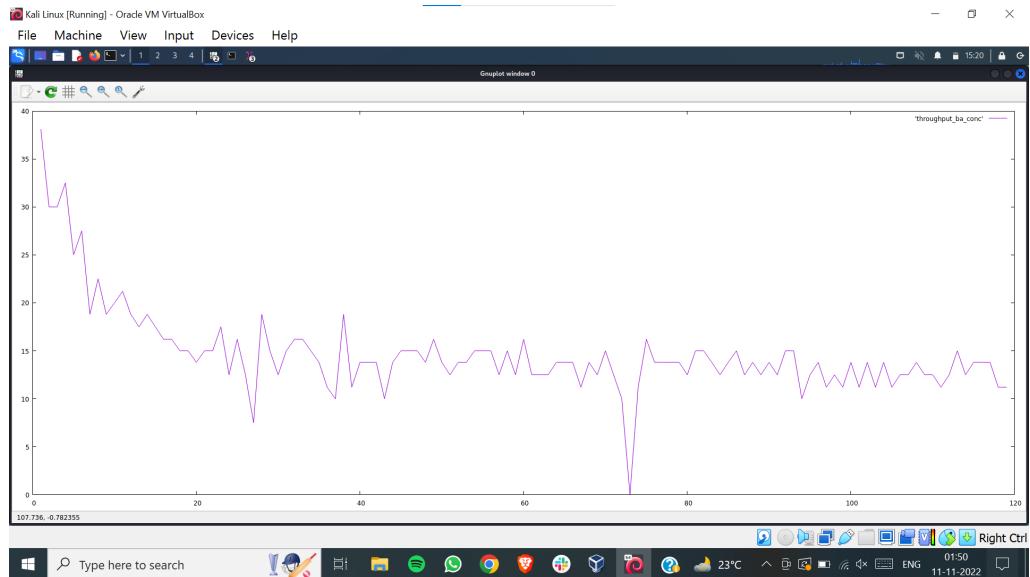


Avg Throughput for C = 0.0135 GBps

Iteration 3

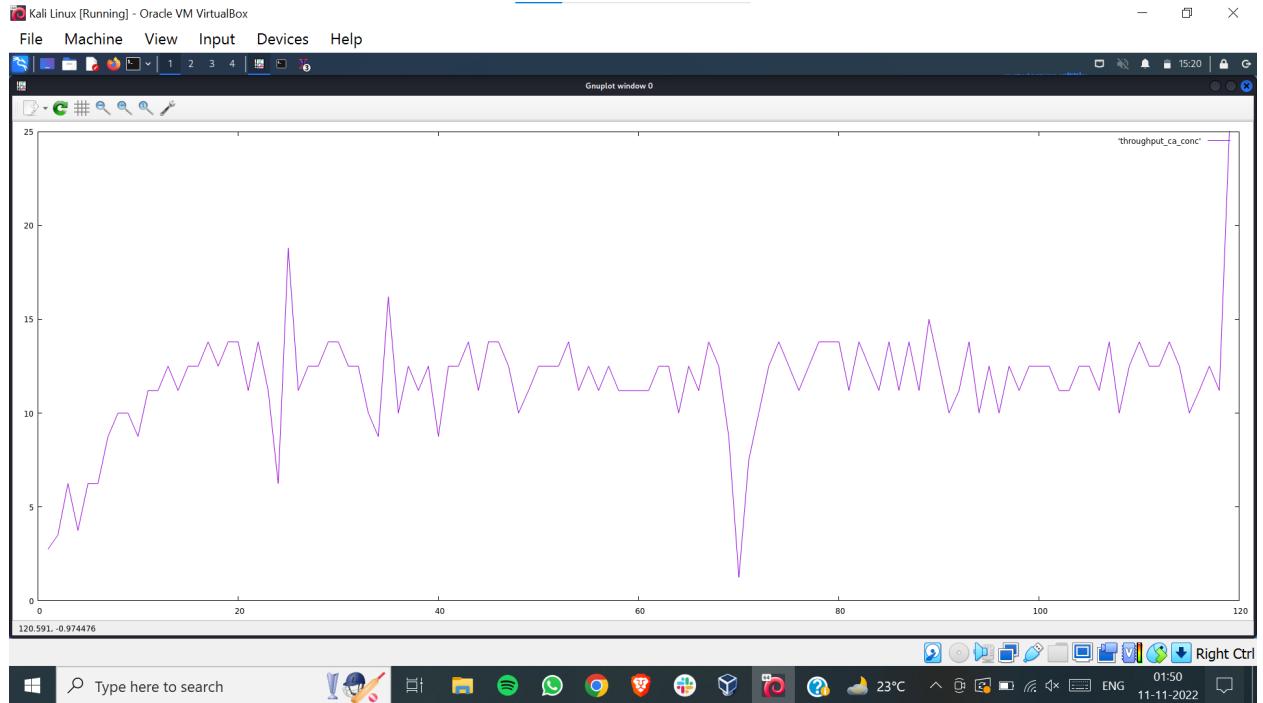
The commands run were the same again.

Throughput Plot for Node B-



Avg Throughput for B = 0.0127 GBps

Throughput Plot for Node C-



Avg Throughput for C = 0.0125 GBps

Avg Throughput for B over 3 iterations = 0.01186 GBps

Avg Throughput for B over 3 iterations = 0.01313 GBps

Observation- As expected, the individual throughputs for both the clients B and C have reduced as compared to the previous cases, when both of them were the sole clients for server A. Their throughput while running together take roughly the same value.

f) In this case, I have changed the link4 delay to 1ms. I have then run the server on A and clients on B and C concurrently using a similar procedure to the one in part e.

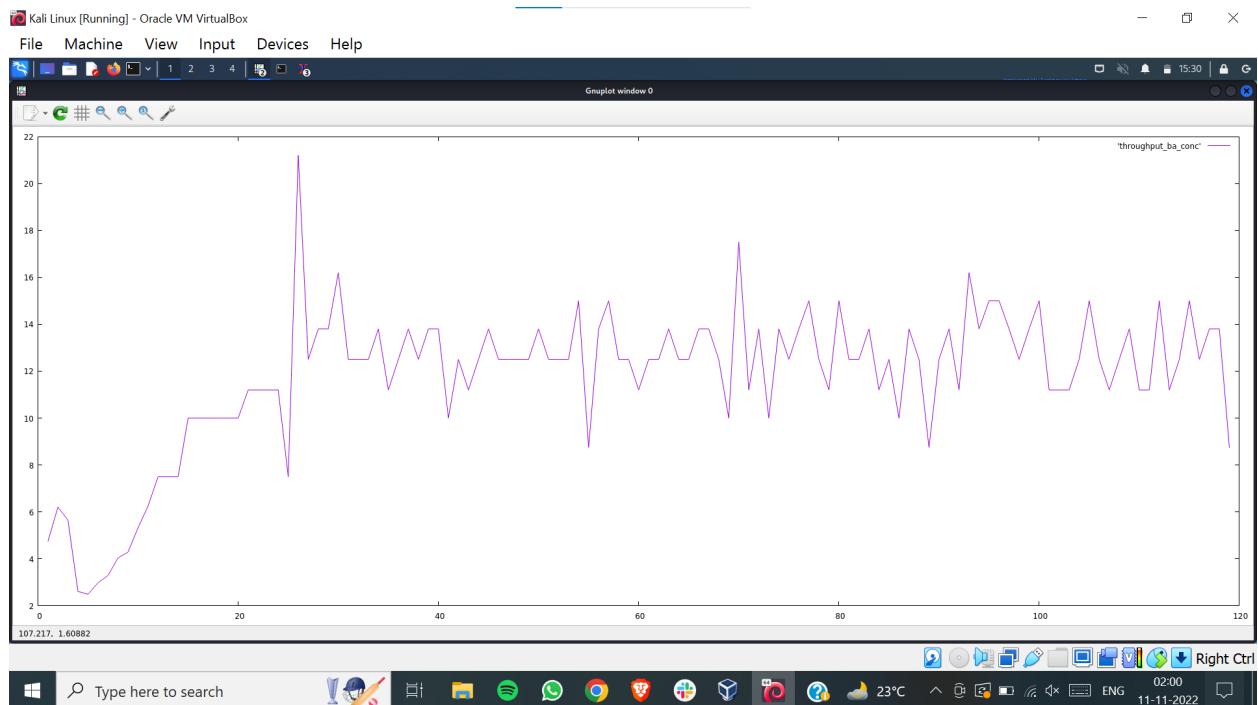
```

1 from mininet.topo import Topo
2 from mininet.link import TCLink
3 from mininet.node import OVSKernelController
4
5
6 class MyTopo( Topo ):
7     "Simple topology example."
8
9     def build( self ):
10        "Create custom topo."
11
12        # Initialize topology
13        Topo.build(self)
14
15        # Add hosts and switches
16        A = self.addHost('A')
17        B = self.addHost('B')
18        C = self.addHost('C')
19        D = self.addHost('D')
20        R1 = self.addSwitch('R1')
21        R2 = self.addSwitch('R2')
22
23        # Add links
24        L1 = self.addLink(A, R1, cls=TCLink, bw=1000, delay='1ms')
25        L5 = self.addLink(D, R1, cls=TCLink, bw=1000, delay='1ms')
26        L3 = self.addLink(B, R2, cls=TCLink, bw=1000, delay='1ms')
27        L4 = self.addLink(C, R2, cls=TCLink, bw=1000, delay='1ms')
28        L5 = self.addLink(R1, R2, cls=TCLink, bw=500, delay='10ms ')
29
30
31 topos = {'mytopo': (lambda: MyTopo())}
32

```

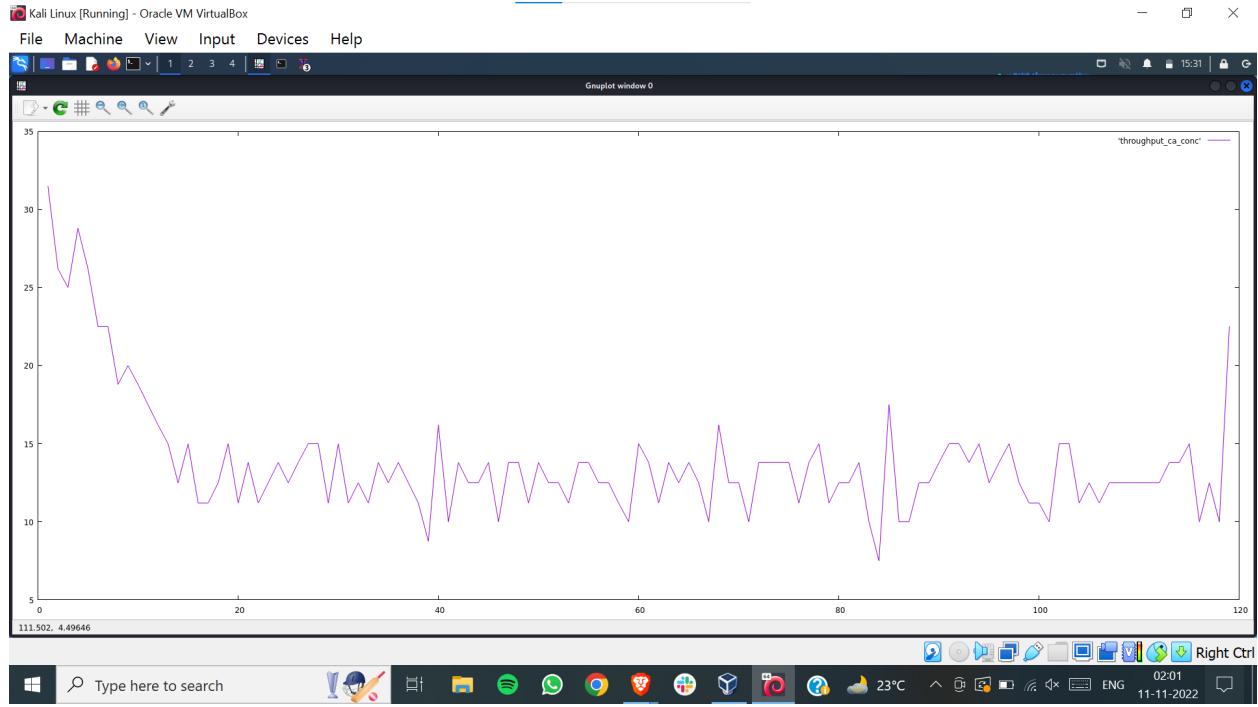
Iteration 1

Throughput Plot for Node B-



Avg Throughput for B = 0.0125 Gbps

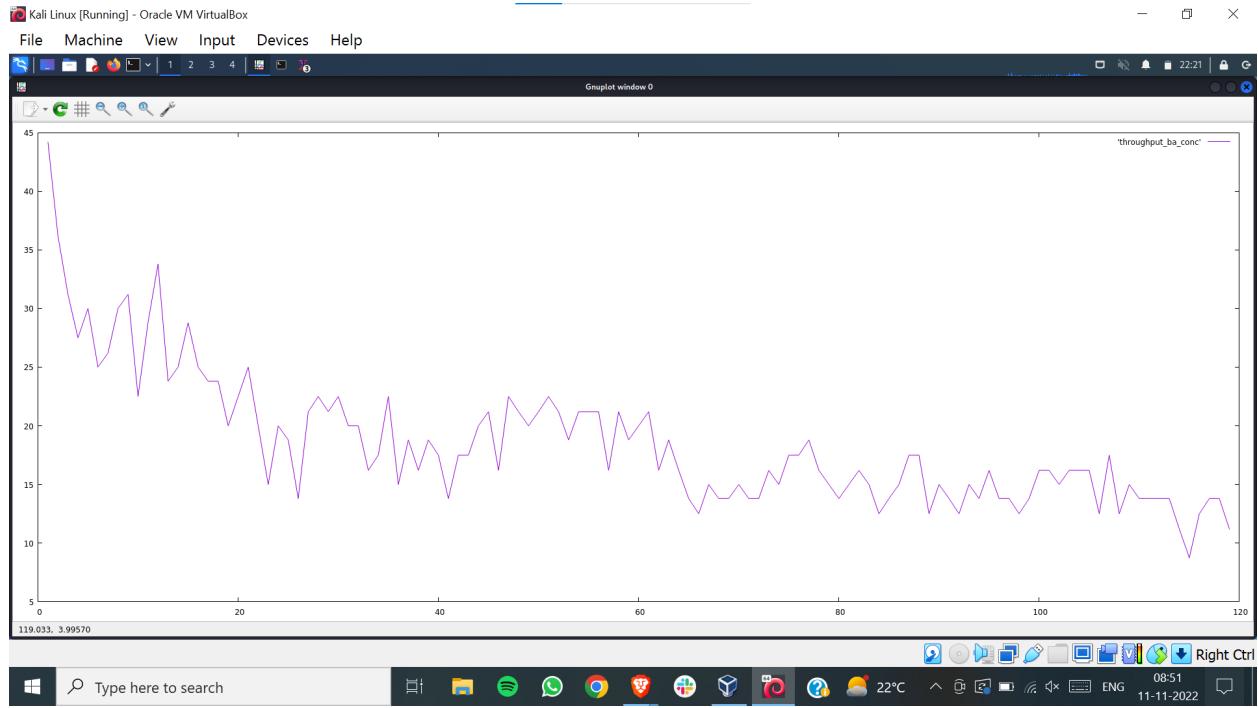
Throughput Plot for Node C-



Avg Throughput for C = 0.0142 GBps

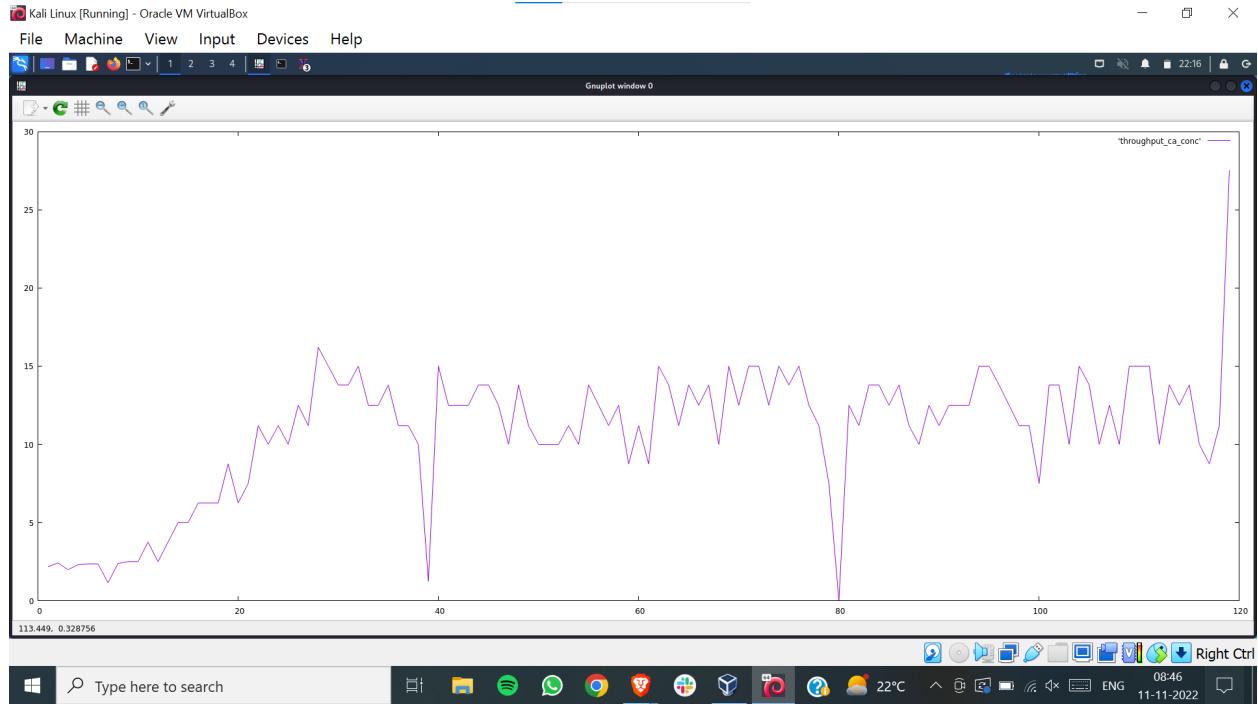
Iteration 2

Throughput Plot for Node B-



Avg Throughput for B = 0.0165 GBps

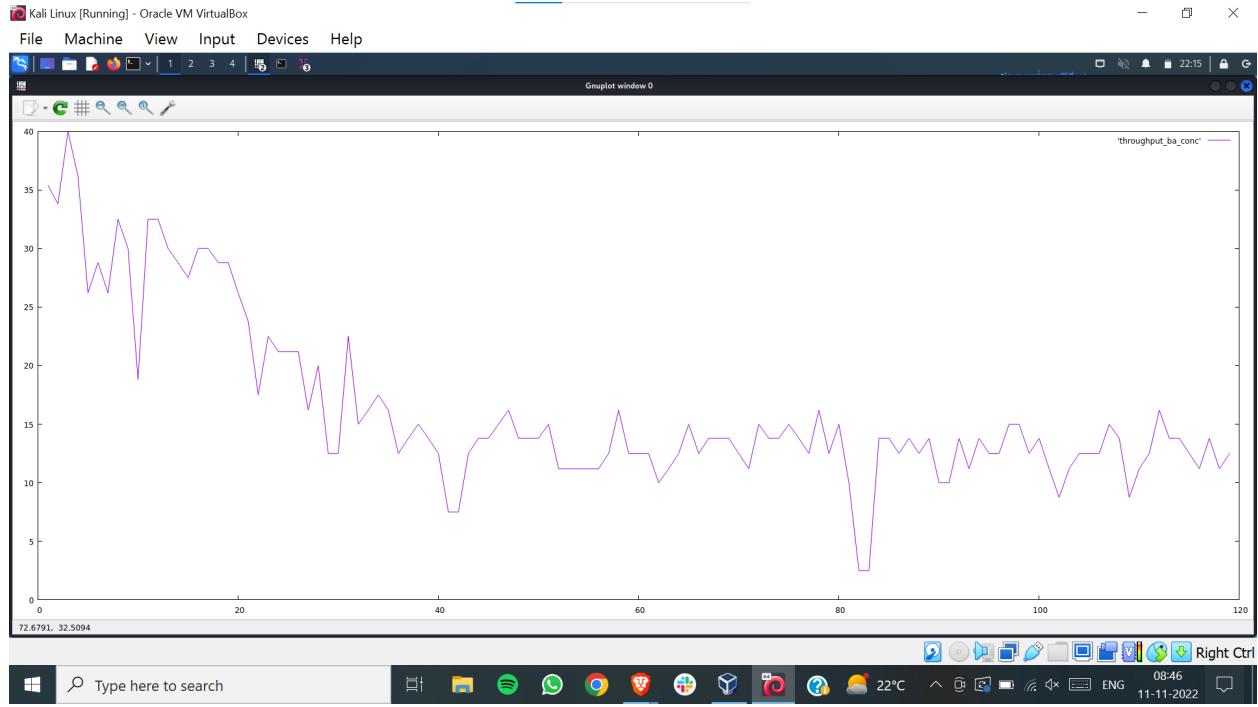
Throughput Plot for Node C-



Avg Throughput for C = 0.0142 GBps

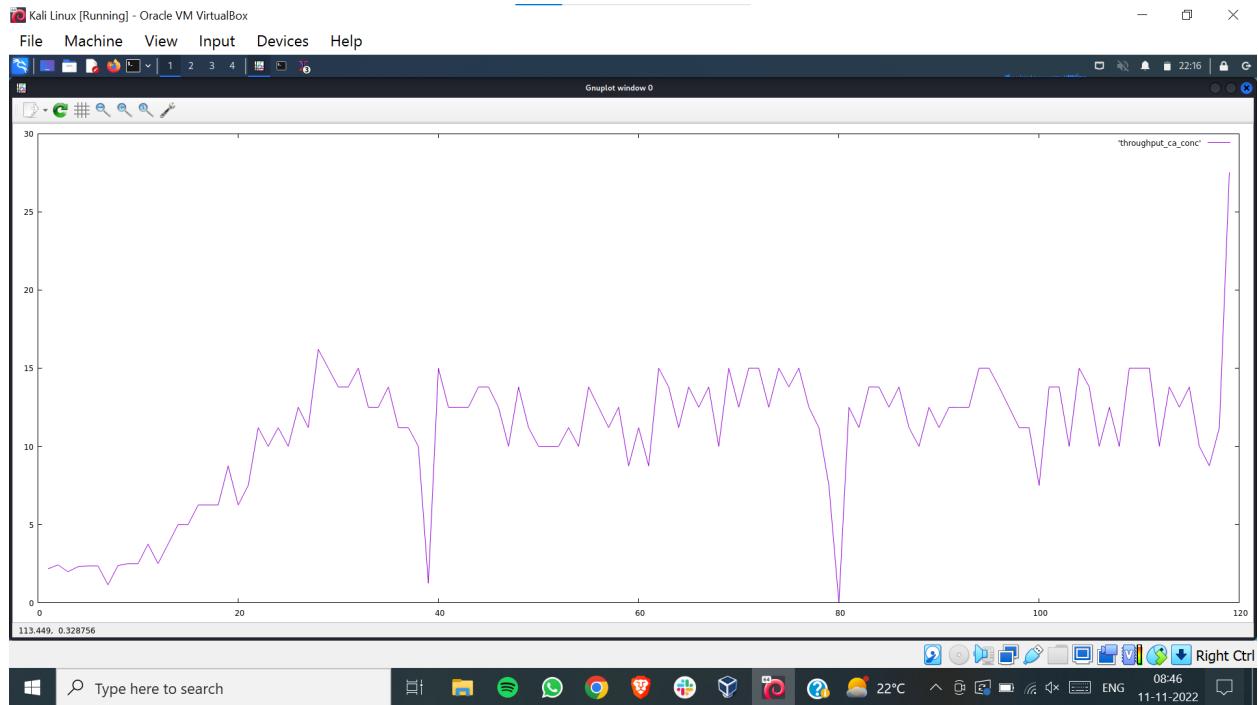
Iteration 3

Throughput Plot for Node B-



Avg Throughput for B = 0.0141 GBps

Throughput Plot for Node C-



Avg Throughput for C = 0.0147 GBps

Avg Throughput for B over 3 iterations= 0.014433 GBps

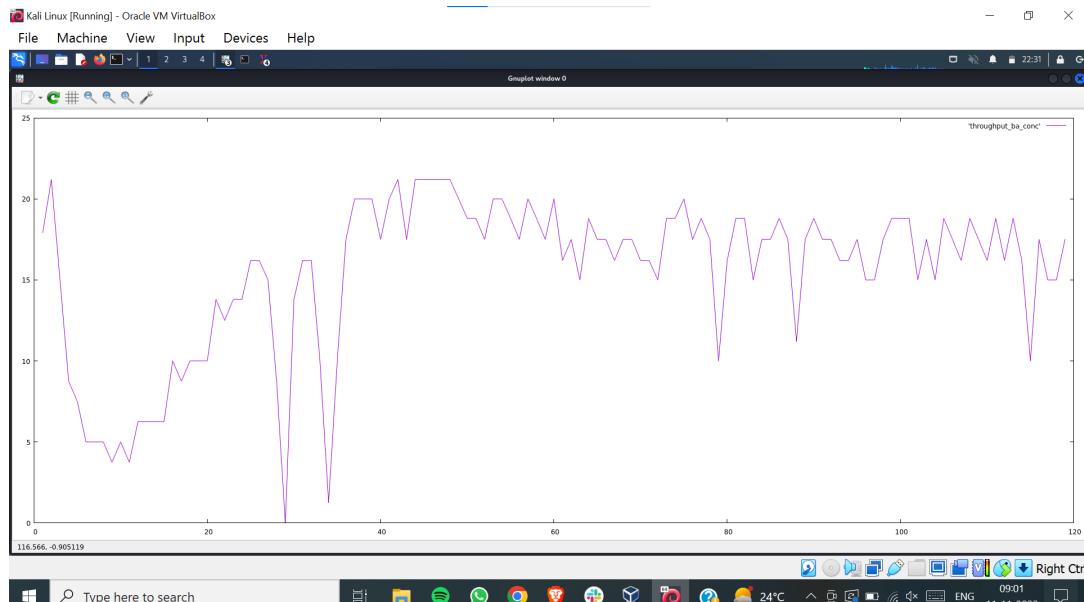
Avg Throughput for C over 3 iterations= 0.014367 GBps

Observation - We can observe that the throughput for client C has increased as compared to the previous case as the delay value is smaller in this case. This was as expected.

g) In this part, I let the link 4 delay be the modified value of 1ms. I ran the iperf serve on Node A and the iperf clients on Node B, C and D.

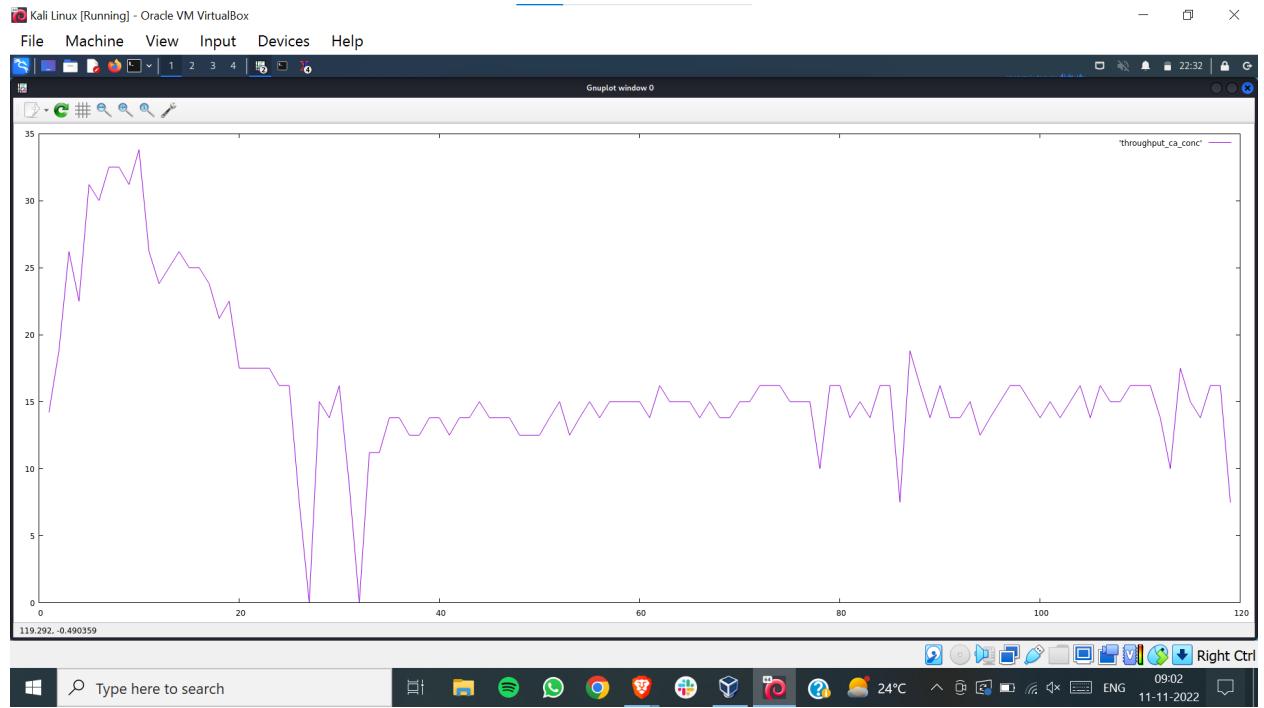
Iteration 1

Throughput for Node B-



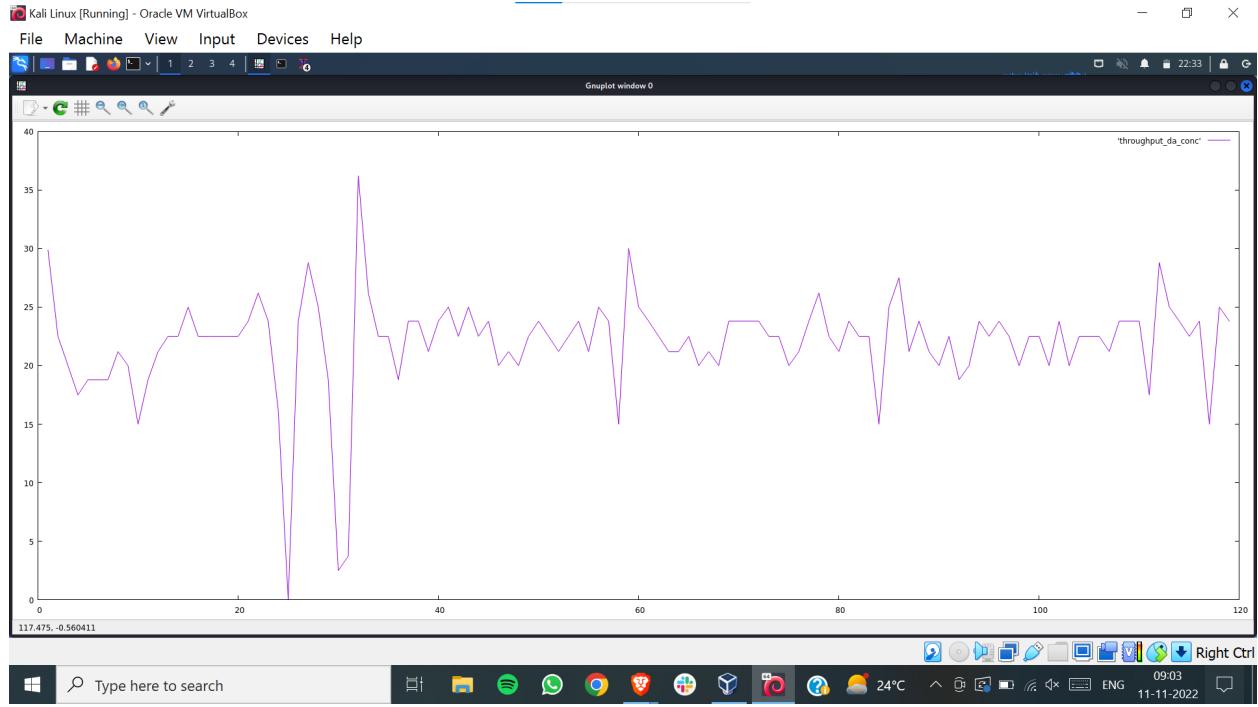
Avg Throughput for B = 16.5672 Mbps

Throughput for Node C-



Avg Throughput for C = 14.8756 MBps

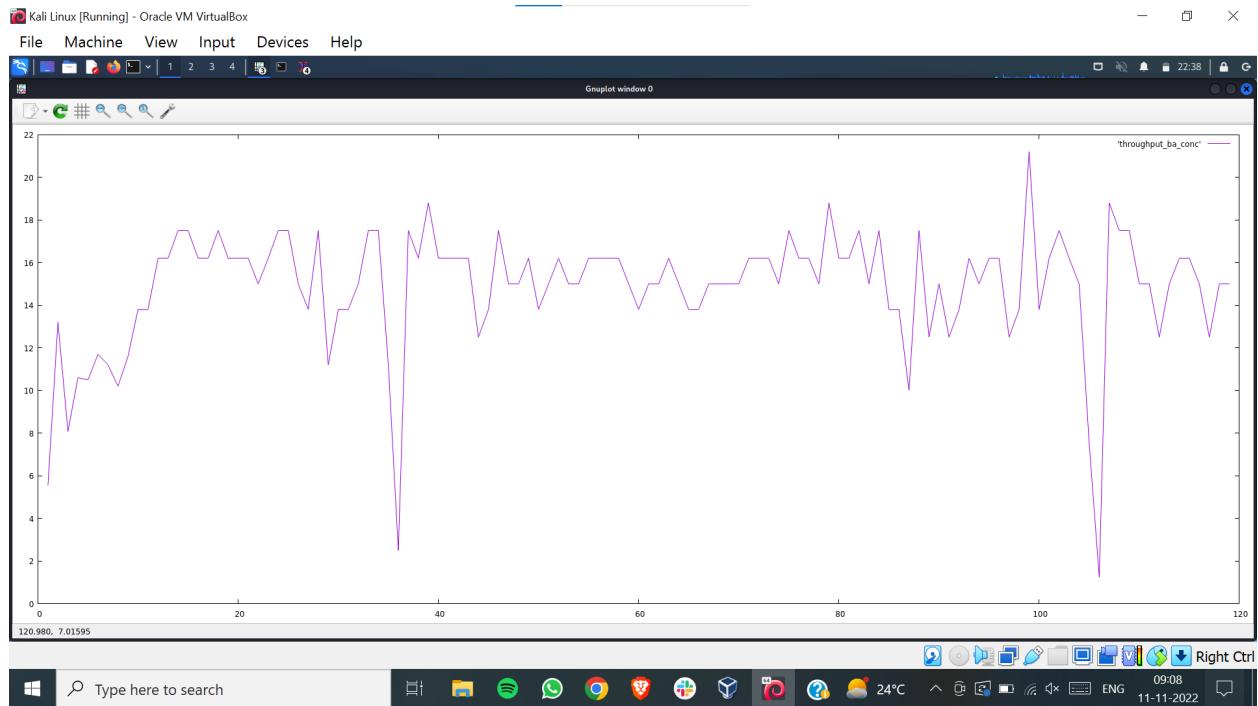
Throughput for Node D-



Avg Throughput for D = 22.5673 MBps

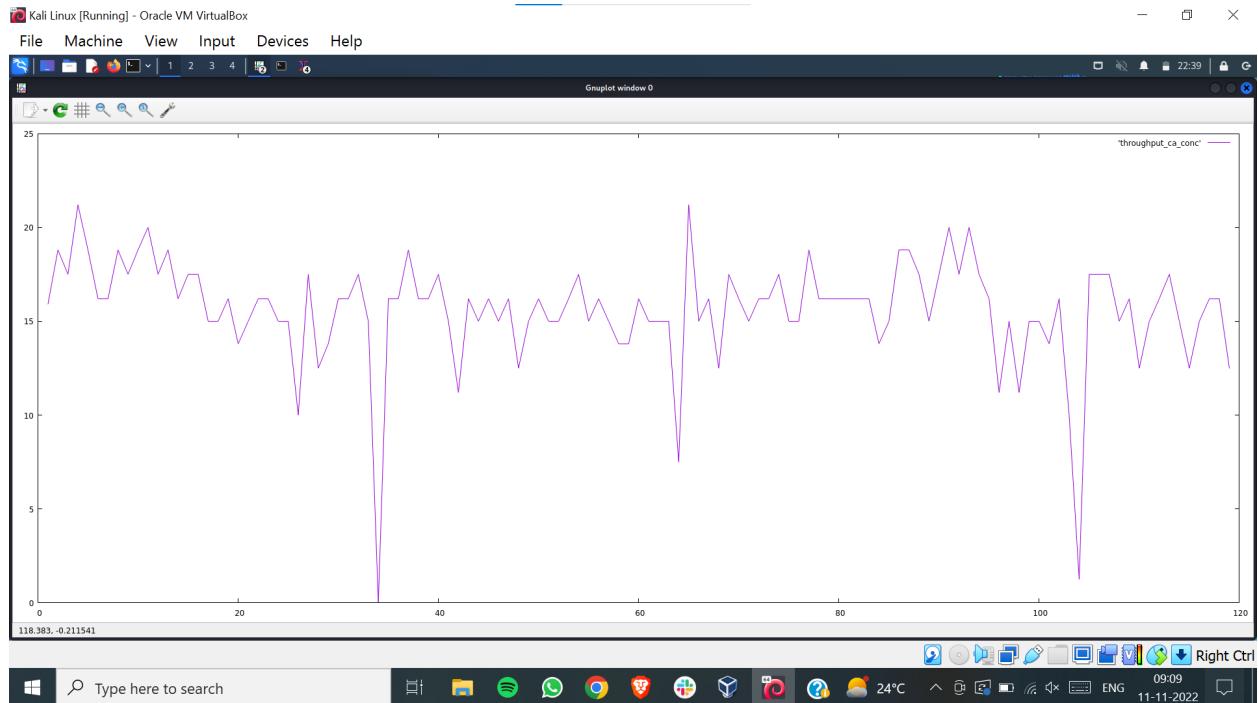
Iteration 2

Throughput for Node B-



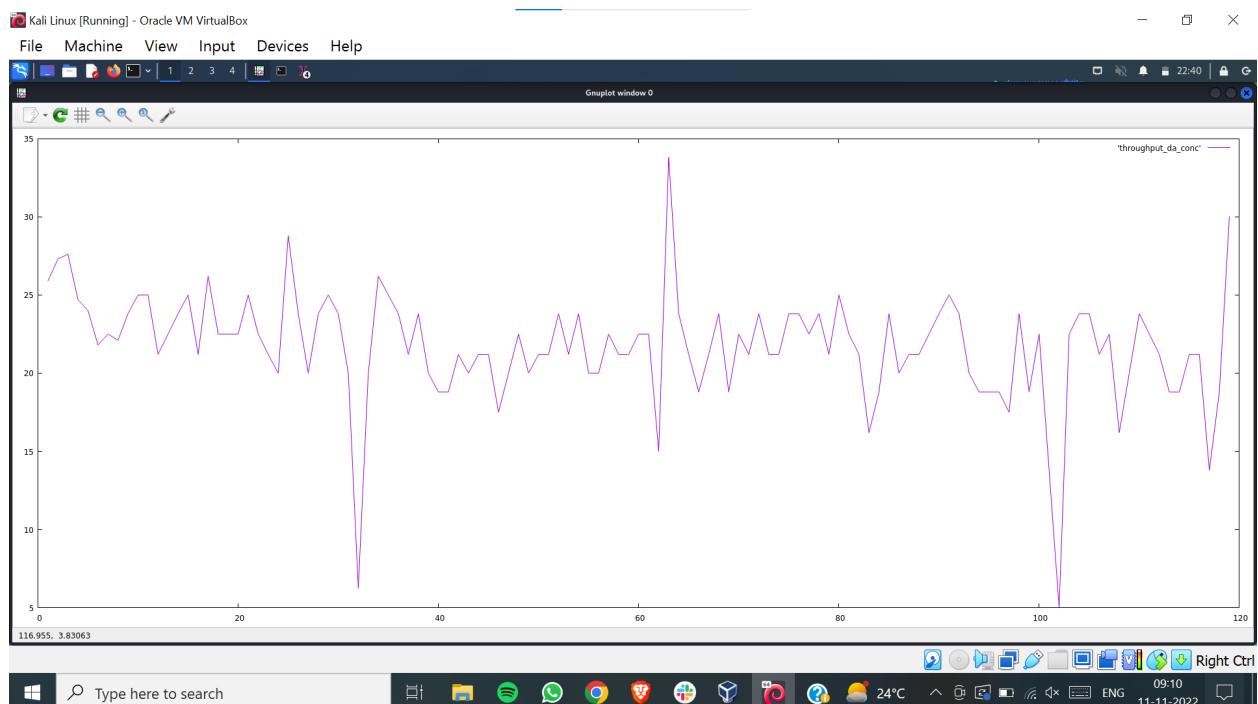
Avg Throughput for B = 16.2316 MBps

Throughput for Node C-



Avg Throughput for C = 16.1324 MBps

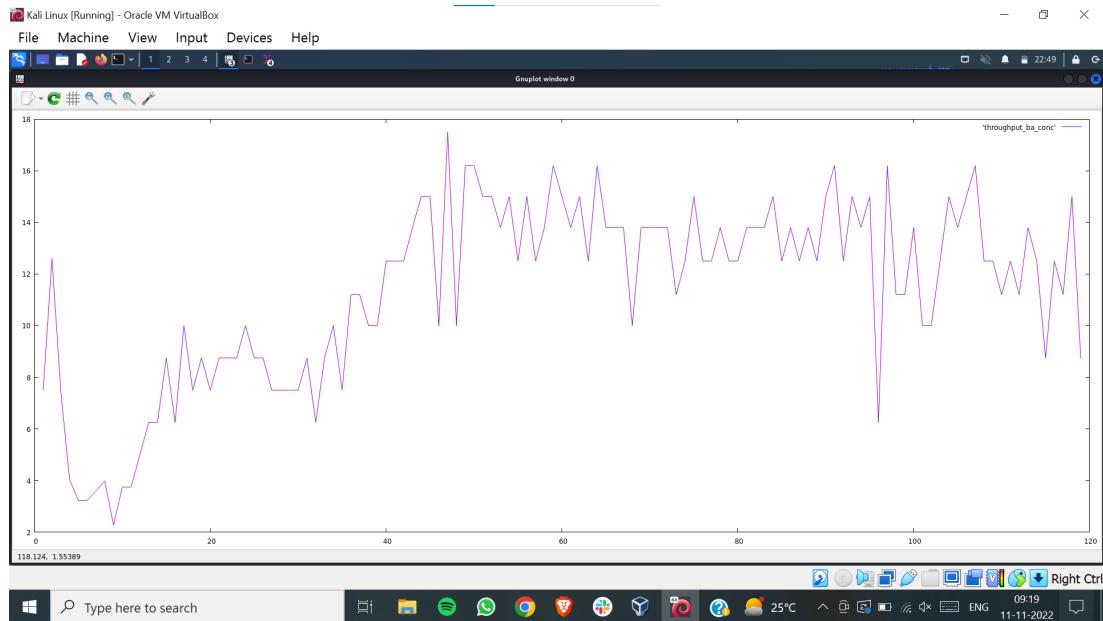
Throughput for Node D-



Avg Throughput for D = 21.8923 MBps

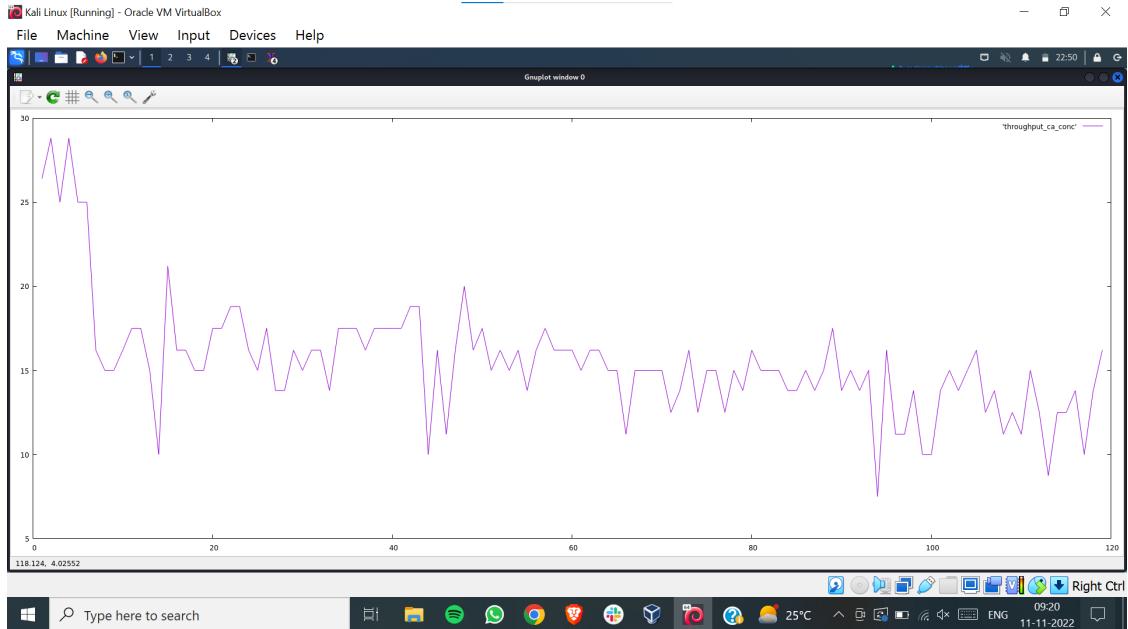
Iteration 3

Throughput for Node B-



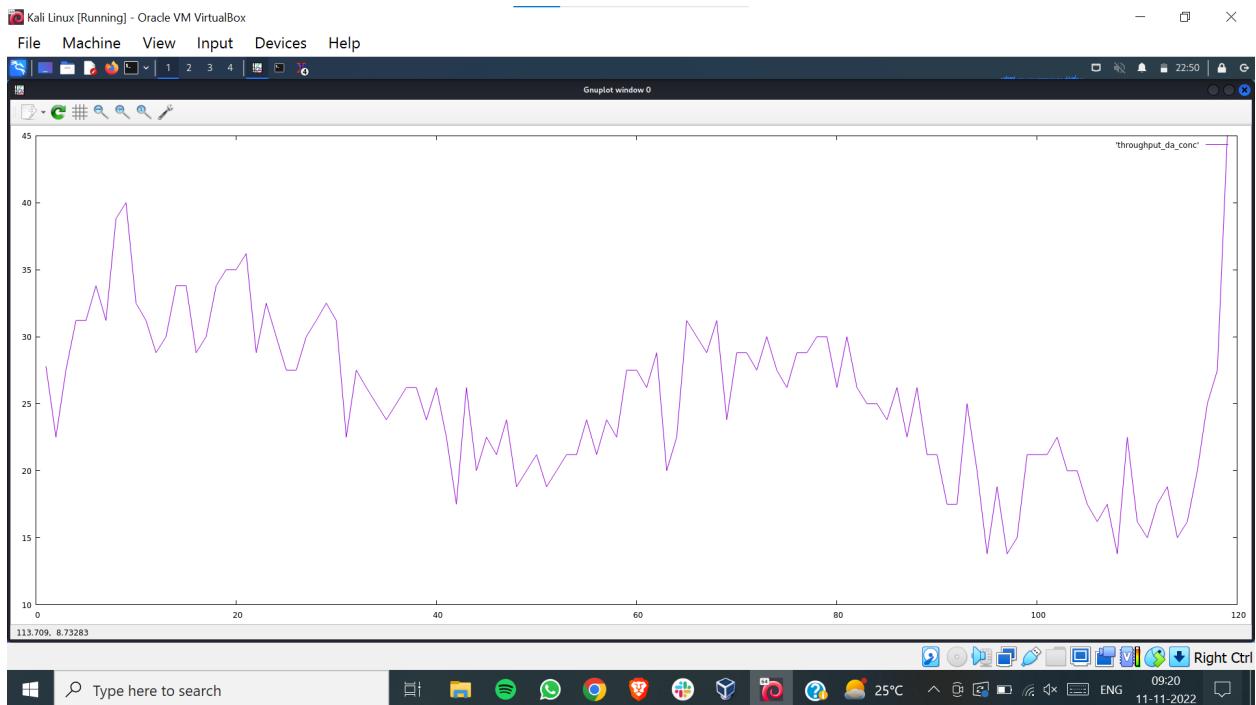
Avg Throughput for B = 13.5623 MBps

Throughput for Node C-



Avg Throughput for C = 14.8745 MBps

Throughput for Node D-



Avg Throughput for D = 25.1254 MBps

Observations - We can observe that the throughput has gone down. This is expected because we are running multiple clients at the same time. The throughput of B, C, D all is lesser than in the cases when they were the only client being serviced by A. The throughput values for B and C are similar because their bandwidth is same. Throughput of D is more due to D having higher bandwidth.

Question 3

Case 1 - Loss =1%

This command runs a linear topology with Loss = 1% and Bandwidth = 50 Mbps

```
(rahulc28@rahulc28) [~]
$ sudo mn --link tc,loss=1,bw=50
[sudo] password for rahulc28:
** No default OpenFlow controller found for default switch!
** Falling back to OVS Bridge
** Creating network
** Adding controller
** Adding hosts:
h1 h2
** Adding switches:
s1
** Adding links:
(50.00Mbit 1.00000% loss) (50.00Mbit 1.00000% loss) (h1, s1) (50.00Mbit 1.00000% loss) (50.00Mbit 1.00000% loss)
** Configuring hosts
h1 h2
** Starting controller

** Starting 1 switches
s1 ... (50.00Mbit 1.00000% loss) (50.00Mbit 1.00000% loss)
** Starting CLI:
mininet> xterm h1 h2
mininet> 
```

"Node h1"

[root@rahulc28 ~]# iperf3 -s

Server listening on 5201 (test #1)

Accepted connection from 10.0.0.2, port 59366

[7] local 10.0.0.1 port 5201 connected to 10.0.0.2 port 59366

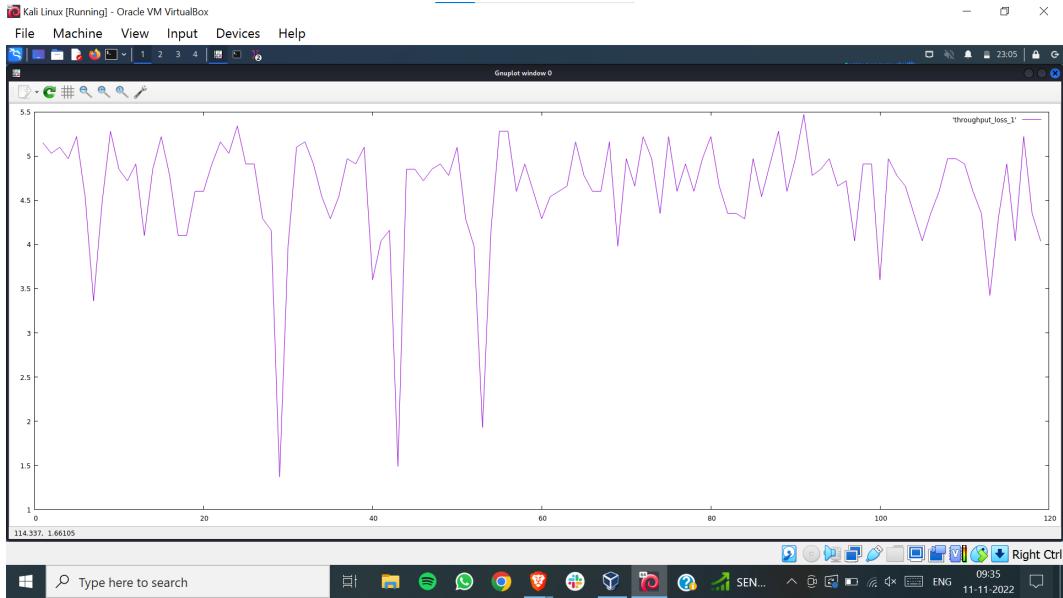
Interval	Transfer	Bitrate
[7] 0.00-1.00	sec 4.84 Mbytes	40.6 Mbits/sec
[7] 1.00-2.00	sec 4.89 Mbytes	41.8 Mbits/sec
[7] 2.00-3.00	sec 5.07 Mbytes	42.6 Mbits/sec
[7] 3.00-4.00	sec 4.93 Mbytes	41.3 Mbits/sec
[7] 4.00-5.00	sec 5.18 Mbytes	43.4 Mbits/sec
[7] 5.00-6.00	sec 4.62 Mbytes	38.8 Mbits/sec
[7] 6.00-7.00	sec 3.34 Mbytes	28.0 Mbits/sec
[7] 7.00-8.00	sec 4.53 Mbytes	38.0 Mbits/sec
[7] 8.00-9.00	sec 5.14 Mbytes	43.1 Mbits/sec
[7] 9.00-10.00	sec 4.85 Mbytes	40.7 Mbits/sec
[7] 10.00-11.00	sec 4.70 Mbytes	39.4 Mbits/sec
[7] 11.00-12.00	sec 4.97 Mbytes	41.7 Mbits/sec
[7] 12.00-13.00	sec 4.16 Mbytes	34.0 Mbits/sec
[7] 13.00-14.00	sec 4.63 Mbytes	40.6 Mbits/sec
[7] 14.00-15.00	sec 5.22 Mbytes	43.8 Mbits/sec
[7] 15.00-16.00	sec 4.88 Mbytes	40.8 Mbits/sec

"Node h2"

[root@rahulc28 ~]# iperf3 -t 120 -l 1 > result_loss_1

[root@rahulc28 ~]#

Throughput Plot



Avg Throughput = 0.00465 GBps
Case 2 - Loss = 2%

This command runs a linear topology with Loss = 2% and Bandwidth = 50 Mbps

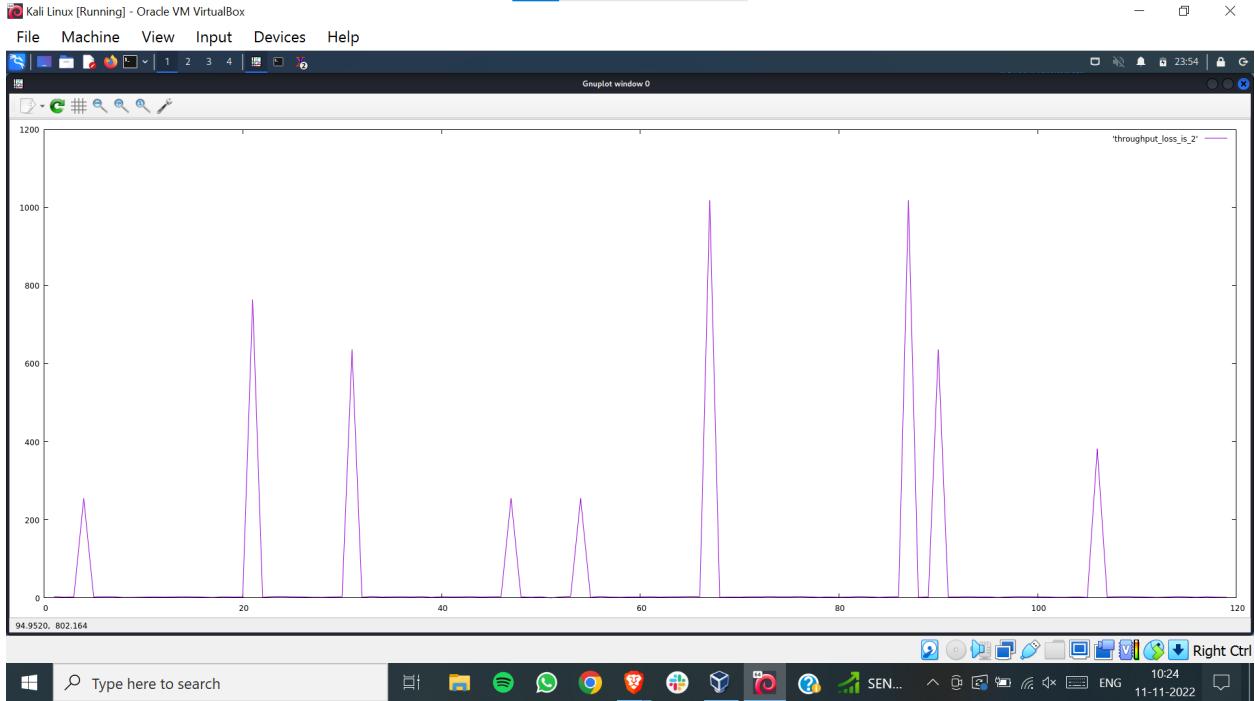
```

[rahulc28@rahulc28:~]
$ sudo mn --link tc,loss=2,bw=50
[sudo] password for rahulc28:
** No default OpenFlow controller found for default switch!
** Falling back to OVS Bridge
** Creating network
** Adding controller
** Adding hosts:
h1 h2
** Adding switches:
s1
** Adding links:
(50.00Mbit 2.00000% loss) (50.00Mbit 2.00000% loss) (h1, s1)
** Configuring hosts
h1 h2
** Starting controller

** Starting 1 switches
s1 ... (50.00Mbit 2.00000% loss) (50.00Mbit 2.00000% loss)
** Starting CLI:
mininet> xterm h1 h2
mininet> []

```

Throughput Plot



Avg Throughput = 0.000475 GBps

Case 3 - Loss = 5%

This command runs a linear topology with Loss = 2% and Bandwidth = 50 Mbps

```
(rahulc28@rahulc28) [~] $ sudo mn --link tc,loss=5,bw=50
*** No default OpenFlow controller
*** Falling back to OVS Bridge
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(50.00Mbit 5.00000% loss) (50.00Mbps)
*** Configuring hosts
h1 h2
*** Starting controller

*** Starting 1 switches
s1 ... (50.00Mbit 5.00000% loss) (50.00Mbps)
*** Starting CLI:
mininet> xterm h1 h2
mininet> 
```

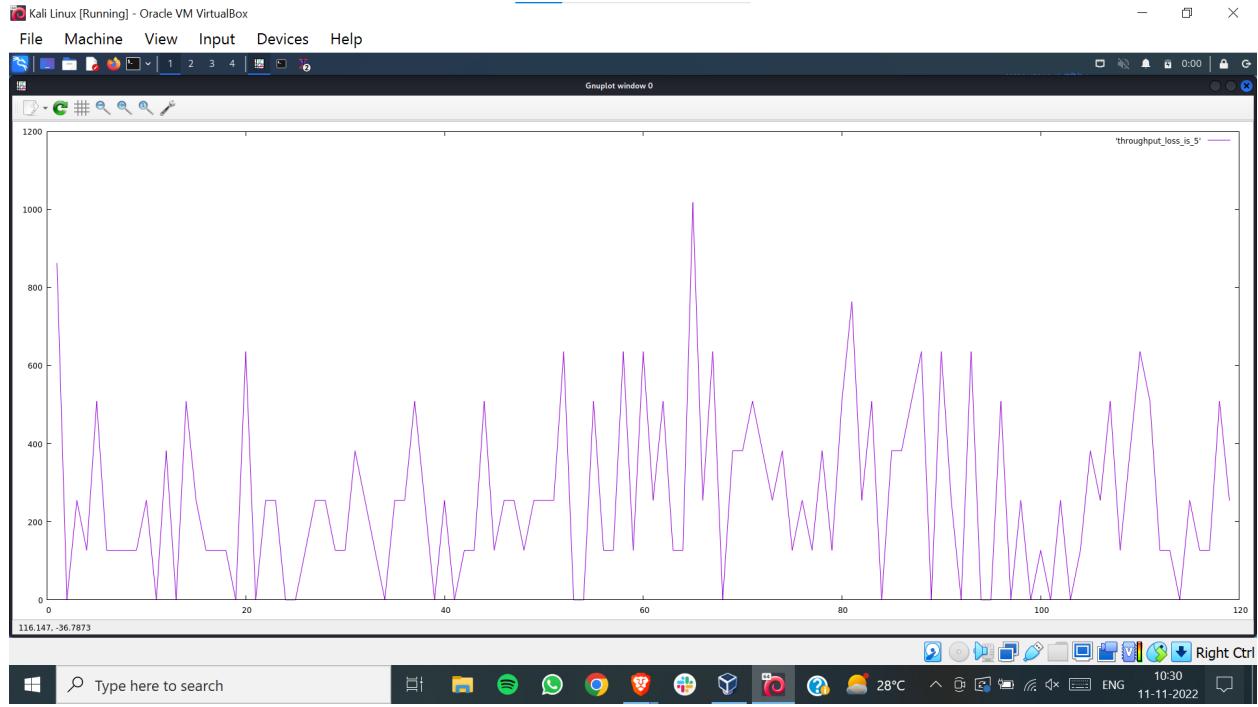
"Node: h1"

```
[ 7] 64.00-65.00 sec 970 KBytes 7.95 Mbytes
[ 7] 65.00-66.00 sec 307 KBytes 2.51 Mbytes/sec
[ 7] 66.00-67.00 sec 588 KBytes 4.81 Mbytes/sec
[ 7] 67.00-68.00 sec 63.8 KBytes 522 kbytes/sec
[ 7] 68.00-69.00 sec 375 KBytes 3.07 Mbytes/sec
[ 7] 69.00-70.00 sec 334 KBytes 2.73 Mbytes/sec
[ 7] 70.00-71.00 sec 568 KBytes 4.16 Mbytes/sec
[ 7] 71.00-72.00 sec 568 KBytes 4.16 Mbytes/sec
[ 7] 72.00-73.00 sec 247 KBytes 2.03 Mbytes/sec
[ 7] 73.00-74.00 sec 313 KBytes 2.56 Mbytes/sec
[ 7] 74.00-75.00 sec 211 KBytes 1.73 Mbytes/sec
[ 7] 75.00-76.00 sec 229 KBytes 1.88 Mbytes/sec
[ 7] 76.00-77.00 sec 107 KBytes 880 kbytes/sec
[ 7] 77.00-78.00 sec 322 KBytes 2.64 Mbytes/sec
[ 7] 78.00-79.00 sec 171 KBytes 1.40 Mbytes/sec
[ 7] 79.00-80.00 sec 489 KBytes 3.84 Mbytes/sec
[ 7] 80.00-81.00 sec 529 KBytes 4.21 Mbytes/sec
[ 7] 81.00-82.00 sec 209 KBytes 1.71 Mbytes/sec
[ 7] 82.00-83.00 sec 451 KBytes 3.70 Mbytes/sec
[ 7] 83.00-84.00 sec 1.41 KBytes 11.6 kbytes/sec
[ 7] 84.00-85.00 sec 489 KBytes 4.01 Mbytes/sec
[ 7] 85.00-86.00 sec 267 KBytes 2.19 Mbytes/sec
[ 7] 86.00-87.00 sec 530 KBytes 4.35 Mbytes/sec
```

"Node: h2"

```
[root@rahulc28 ~]# iperf3 -t120 -n1 -c 10.0.0.1 > result_loss_is_5
```

Throughput Plot



Avg Throughput = 0.000295 GBps

Observations- As the loss parameter associated with the link increases, the throughput decreases. This is expected because if loss increases, retransmissions increase and therefore, the actual throughput decreases.

Question 4

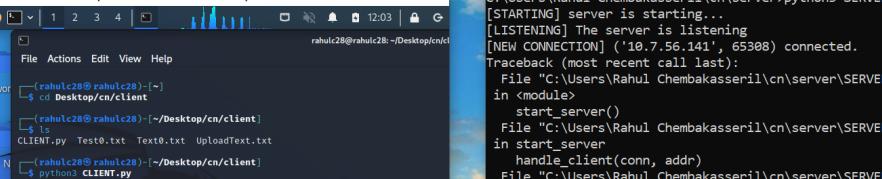
-The networking problem has been solved on paper and a separate pdf file is submitted on the Google Classroom.

Question 5

-Error that occurred when I set the server IP address to the IPv4 Address of the Kali VM on eth0 interface - 10.0.2.15

- Here, I tried to run the server on Kali Linux and the client on Windows.

If I do the opposite, and run the server on Windows and the client on Kali Linux by setting the HOST IP value as the Windows IP, then this works smoothly as shown below.



```
Kali Linux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

File Actions Edit View Help
[rahulc2@rahulc2:~] $ cd Desktop/cn/client
[rahulc2@rahulc2:~/Desktop/cn/client] $ ls
CLIENT.py Test0.txt Text0.txt UploadText.txt
[rahulc2@rahulc2:~/Desktop/cn/client] $ python3 CLIENT.py
Write your Message here: UDP-Test0.txt
Traceback (most recent call last):
  File "/home/rahulc2/Desktop/cn/client/CLIENT.py", line 137, in <module>
    start()
  File "/home/rahulc2/Desktop/cn/client/CLIENT.py", line 132, in start
    send_msg(message, input, client)
  File "/home/rahulc2/Desktop/cn/client/CLIENT.py", line 88, in send_msg
    with open(filename, "wb") as f:
PermissionError: [Errno 13] Permission denied: 'Test0.txt'

[rahulc2@rahulc2:~/Desktop/cn/client] $ python3 CLIENT.py
Write your Message here: UDP-UploadText.txt
UploadText.txt
Upload has taken: 42.22834s
Write your Message here: Right Ctrl

C:\Users\Rahul Chembakasseril\cn\server>python3 SERVER.py
[STARTING] server is starting...
[LISTENING] The server is listening
[NEW CONNECTION] ('10.7.56.141', 65508) connected.
Traceback (most recent call last):
  File "C:\Users\Rahul Chembakasseril\cn\server\SERVER.py", line 153,
in <module>
    start_server()
  File "C:\Users\Rahul Chembakasseril\cn\server\SERVER.py", line 149,
in start_server
    handle_client(conn, addr)
  File "C:\Users\Rahul Chembakasseril\cn\server\SERVER.py", line 116,
in handle_client
    conn.sendall(bytes_read)
ConnectionAbortedError: [WinError 10053] An established connection was
aborted by the software in your host machine

C:\Users\Rahul Chembakasseril\cn\server>python3 SERVER.py
[STARTING] server is starting...
[LISTENING] The server is listening
[NEW CONNECTION] ('10.7.56.141', 65438) connected.
File Received
[NEW CONNECTION] ('10.7.56.141', 65530) connected.
```

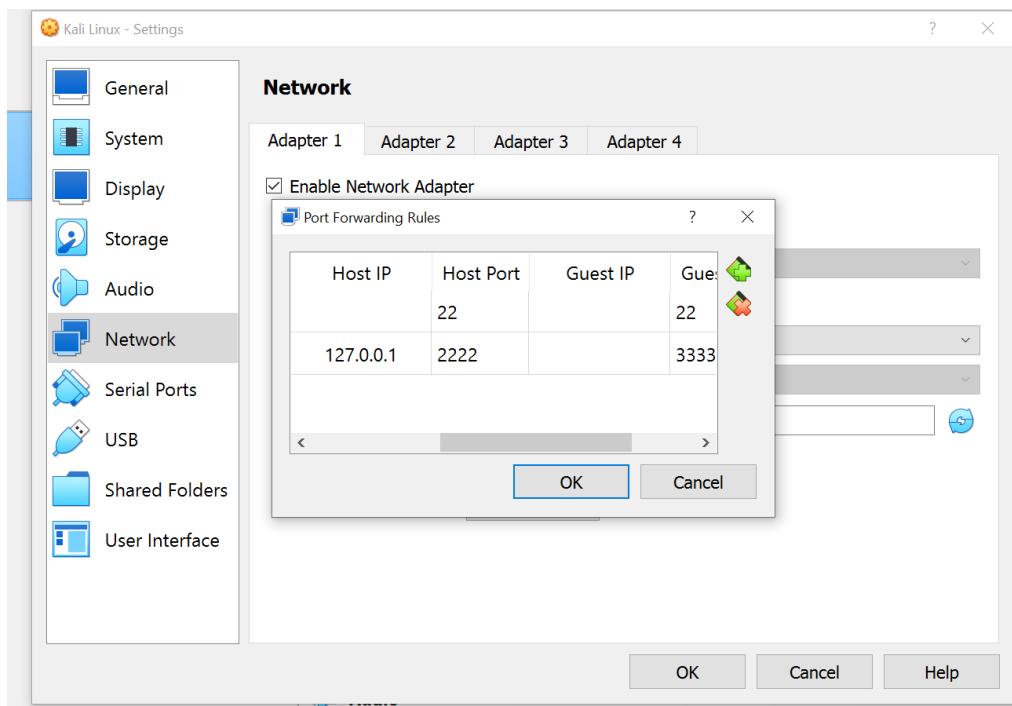
Running the server on Kali Linux and the client on Windows doesn't work. The problem seems to be due to a an error related to NAT - Network Address Translation.

Currently, Kali Linux is being run on my system using Oracle's Virtual Box. The problem is with how the NAT has been configured between my Windows machine and Kali Linux using Oracle's Virtual Box.

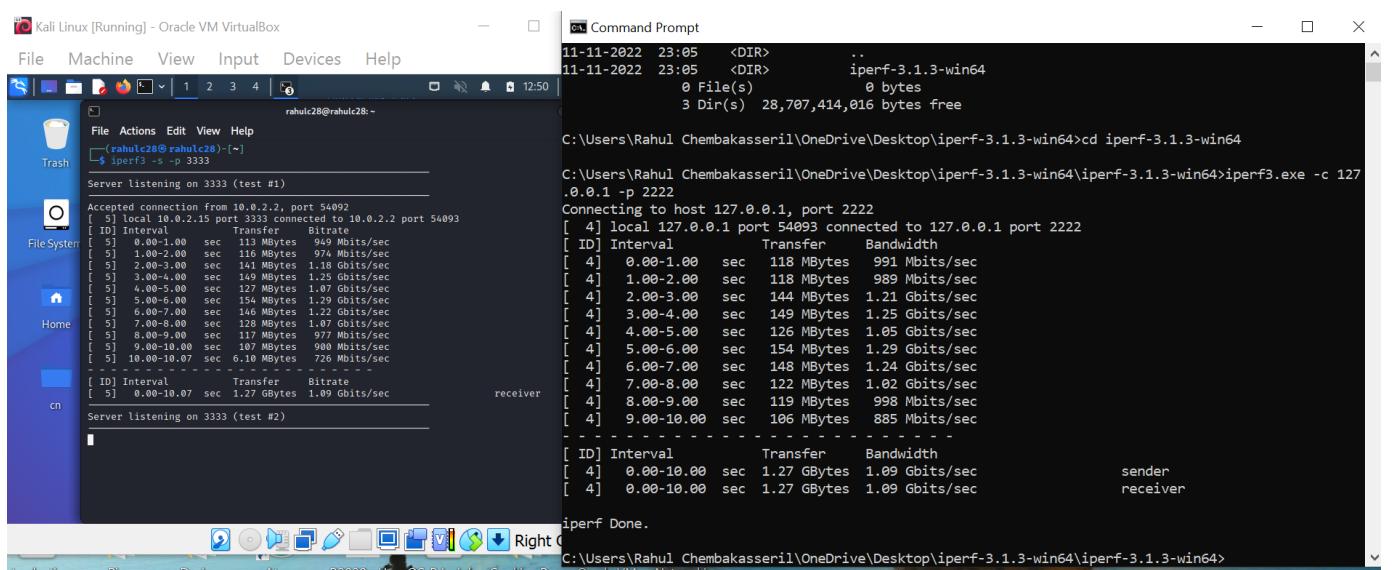
Currently, I'm trying to visualize the VM behind the NAT and hence the requests from the client on Windows won't be able to connect to the server on Kali. The serve on Kali Linux won't be able to listen to any requests.

Solution-

I set up port forwarding by forwarding all the traffic coming to on our host (Kali VM) to a different port by changing the Oracle Virtual Box Kali Linux network settings. The Host IP - 127.0.0.1 - means that any traffic intended towards this address on host on Host Port 2222 will get forwarded to the Guest Port 3333. Now we can run the iperf3 server and it works fine by forwarding the traffic to port 3333.



- Now, when I run the iperf server on Kali Linux and the iperf client on Windows, it works.



Hence, the iperf server works on Kali Linux and the iperf client on Windows after enabling the port forwarding.