

Part C

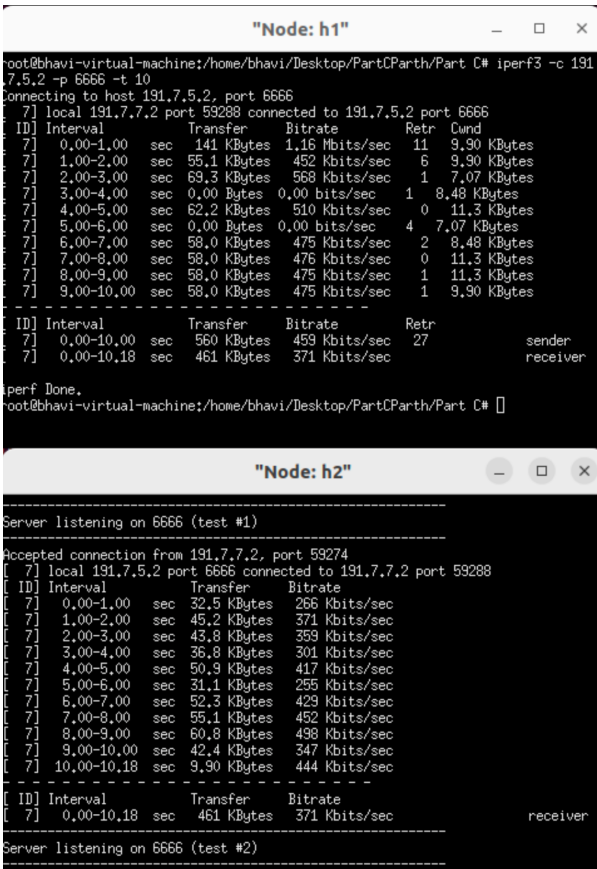
C. IPerf test

a. MyIperf.py added

b. Screenshots

The following results show the output of the IPerf test:

i) 10 kb



The screenshot displays two terminal windows. The top window, titled "Node: h1", shows the output of the command `iperf3 -c 191.7.5.2 -p 6666 -t 10`. It reports a connection to host 191.7.5.2 on port 6666 and provides a detailed table of performance metrics over 10-second intervals. The bottom window, titled "Node: h2", shows the server-side output, indicating it is listening on port 6666 and has accepted a connection from 191.7.7.2 on port 59274. It also provides a table of performance metrics.

Interval	Transfer	Bitrate	Retr	Cwnd
0.00-1.00 sec	141 KBytes	1.16 Mbits/sec	11	9.90 KBytes
1.00-2.00 sec	55.1 KBytes	452 Kbits/sec	6	9.90 KBytes
2.00-3.00 sec	59.3 KBytes	568 Kbits/sec	1	7.07 KBytes
3.00-4.00 sec	0.00 Bytes	0.00 bits/sec	1	8.48 KBytes
4.00-5.00 sec	52.2 KBytes	510 Kbits/sec	0	11.3 KBytes
5.00-6.00 sec	0.00 Bytes	0.00 bits/sec	4	7.07 KBytes
6.00-7.00 sec	58.0 KBytes	475 Kbits/sec	2	8.48 KBytes
7.00-8.00 sec	58.0 KBytes	476 Kbits/sec	0	11.3 KBytes
8.00-9.00 sec	58.0 KBytes	475 Kbits/sec	1	11.3 KBytes
9.00-10.00 sec	58.0 KBytes	475 Kbits/sec	1	9.90 KBytes

Interval	Transfer	Bitrate	Retr	sender
0.00-10.00 sec	560 KBytes	459 Kbits/sec	27	sender
0.00-10.18 sec	461 KBytes	371 Kbits/sec		receiver

Interval	Transfer	Bitrate	receiver
0.00-1.00 sec	32.5 KBytes	266 Kbits/sec	
1.00-2.00 sec	45.2 KBytes	371 Kbits/sec	
2.00-3.00 sec	43.8 KBytes	359 Kbits/sec	
3.00-4.00 sec	36.8 KBytes	301 Kbits/sec	
4.00-5.00 sec	50.9 KBytes	417 Kbits/sec	
5.00-6.00 sec	31.1 KBytes	255 Kbits/sec	
6.00-7.00 sec	52.3 KBytes	429 Kbits/sec	
7.00-8.00 sec	55.1 KBytes	452 Kbits/sec	
8.00-9.00 sec	60.8 KBytes	498 Kbits/sec	
9.00-10.00 sec	42.4 KBytes	347 Kbits/sec	
10.00-10.18 sec	9.90 KBytes	444 Kbits/sec	

Average Bandwidth = 459 kbps

```
bhavi@bhavi-virtual-machine: ~/Desktop/PartCParth/Part C
h1 h2 r1 r2 r3 r4
*** Done
bhavi@bhavi-virtual-machine:~/Desktop/PartCParth/Part C$ sudo python3 MyIperf10.py
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 r1 r2 r3 r4
*** Adding switches:

*** Adding links:
(h1, r1) (h2, r4) (r1, r2) (r1, r3) (r2, r4) (r3, r4)
*** Configuring hosts
h1 h2 r1 r2 r3 r4
*** Starting controller
c0
*** Starting 0 switches

*** Starting CLI:
mininet> h1 ping h2
PING 191.7.5.2 (191.7.5.2) 56(84) bytes of data.
64 bytes from 191.7.5.2: icmp_seq=1 ttl=61 time=244 ms
64 bytes from 191.7.5.2: icmp_seq=2 ttl=61 time=184 ms
64 bytes from 191.7.5.2: icmp_seq=3 ttl=61 time=183 ms
64 bytes from 191.7.5.2: icmp_seq=4 ttl=61 time=182 ms
64 bytes from 191.7.5.2: icmp_seq=5 ttl=61 time=184 ms
64 bytes from 191.7.5.2: icmp_seq=6 ttl=61 time=183 ms
64 bytes from 191.7.5.2: icmp_seq=7 ttl=61 time=183 ms
64 bytes from 191.7.5.2: icmp_seq=8 ttl=61 time=184 ms
64 bytes from 191.7.5.2: icmp_seq=9 ttl=61 time=183 ms
64 bytes from 191.7.5.2: icmp_seq=10 ttl=61 time=184 ms
64 bytes from 191.7.5.2: icmp_seq=11 ttl=61 time=184 ms
64 bytes from 191.7.5.2: icmp_seq=12 ttl=61 time=184 ms
64 bytes from 191.7.5.2: icmp_seq=13 ttl=61 time=183 ms
^C
--- 191.7.5.2 ping statistics ---
13 packets transmitted, 13 received, 0% packet loss, time 1201ms
rtt min/avg/max/mdev = 181.968/188.078/244.050/16.167 ms
mininet>
```

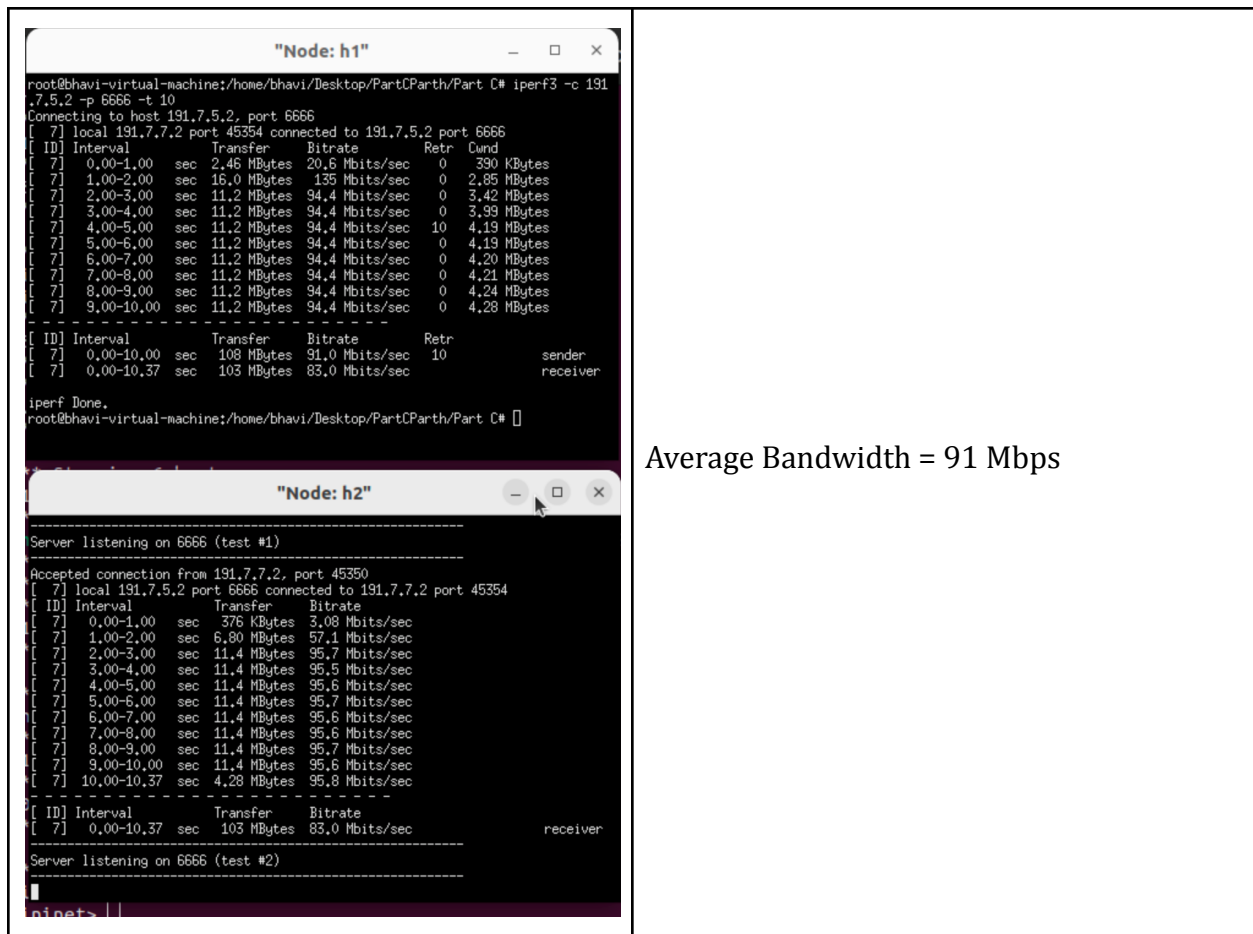
Average RTT = 188.078 ms

BDP in this case = Average Bandwidth x Average RTT
= 459 kbps x 188.078 ms = 86.33 kb

BDP is 86.33 kb while the buffer size is 10 kb, this causes the buffer to get full and packets start dropping. This causes a large number of retransmissions (Retransmissions = 27).

Our average bandwidth in this case is 459 kbps, which is much less than 100 Mbps. Since the buffer is only 10 kb, it gets full much before the entire bandwidth is used.

ii) 5 Mb



```
bhavi@bhavi-virtual-machine: ~/Desktop/PartCParth/Part C
*** Stopping 6 hosts
h1 h2 r1 r2 r3 r4
*** Done
bhavi@bhavi-virtual-machine:~/Desktop/PartCParth/Part C$ sudo python3 MyIperf5.py
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 r1 r2 r3 r4
*** Adding switches:

*** Adding links:
(h1, r1) (h2, r4) (r1, r2) (r1, r3) (r2, r4) (r3, r4)
*** Configuring hosts
h1 h2 r1 r2 r3 r4
*** Starting controller
c0
*** Starting 0 switches

*** Starting CLI:
mininet> h1 ping h2
PING 191.7.5.2 (191.7.5.2) 56(84) bytes of data.
64 bytes from 191.7.5.2: icmp_seq=1 ttl=61 time=244 ms
64 bytes from 191.7.5.2: icmp_seq=2 ttl=61 time=185 ms
64 bytes from 191.7.5.2: icmp_seq=3 ttl=61 time=183 ms
64 bytes from 191.7.5.2: icmp_seq=4 ttl=61 time=185 ms
64 bytes from 191.7.5.2: icmp_seq=5 ttl=61 time=184 ms
64 bytes from 191.7.5.2: icmp_seq=6 ttl=61 time=183 ms
64 bytes from 191.7.5.2: icmp_seq=7 ttl=61 time=185 ms
64 bytes from 191.7.5.2: icmp_seq=8 ttl=61 time=185 ms
64 bytes from 191.7.5.2: icmp_seq=9 ttl=61 time=183 ms
64 bytes from 191.7.5.2: icmp_seq=10 ttl=61 time=183 ms
64 bytes from 191.7.5.2: icmp_seq=11 ttl=61 time=185 ms
64 bytes from 191.7.5.2: icmp_seq=12 ttl=61 time=183 ms
^C
--- 191.7.5.2 ping statistics ---
13 packets transmitted, 12 received, 7.69231% packet loss, time 12022ms
rtt min/avg/max/mdev = 182.652/189.012/244.414/16.730 ms
mininet>
```

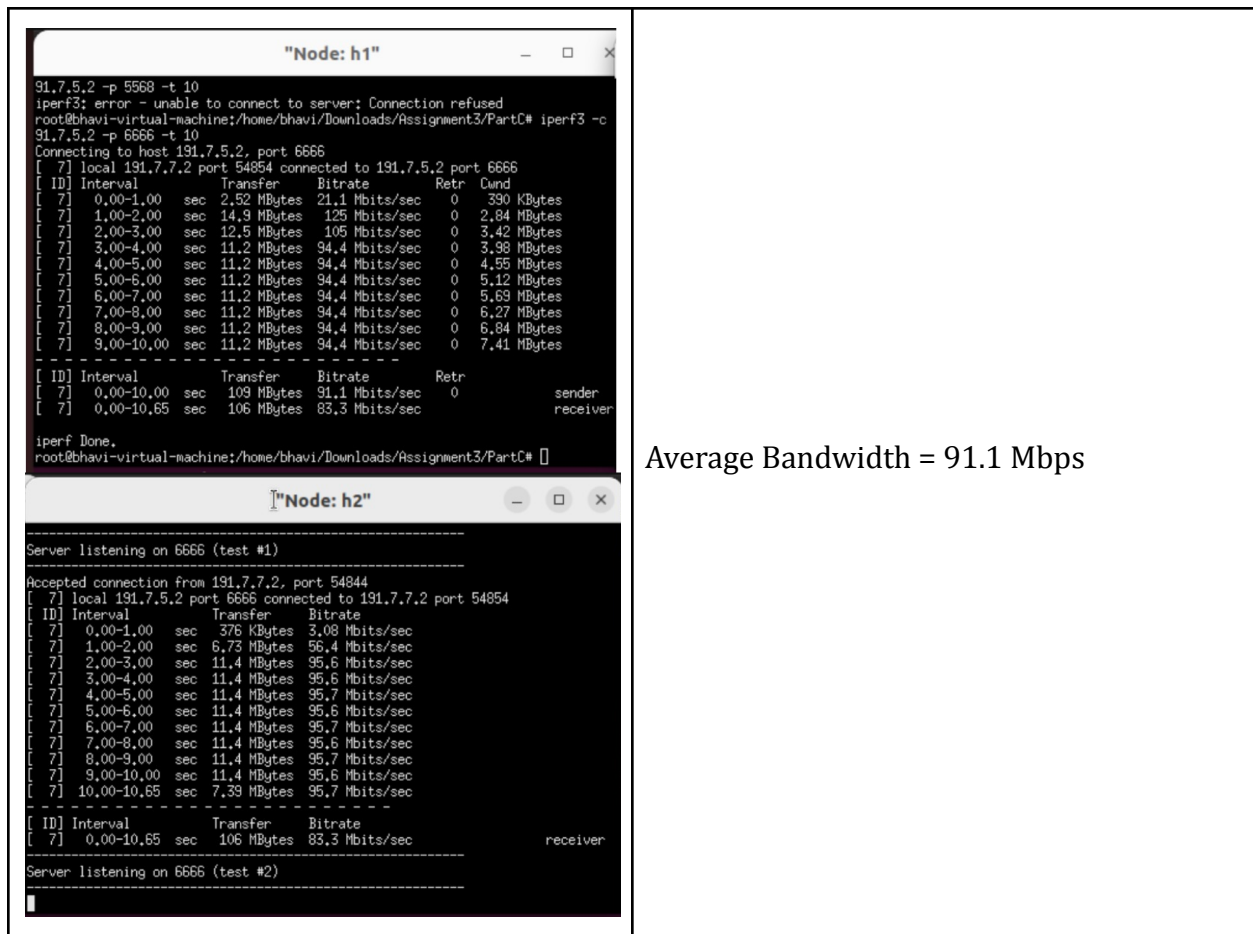
Average RTT = 189.012 ms

BDP in this case = Average Bandwidth x Average RTT
= 91 Mbps x 189.012 ms = 17.2 Mb

BDP is 17.2 Mb while the buffer size is 5 Mb, this causes the buffer to get full and packets start dropping. This causes retransmission of packets (Retransmissions = 10). The number of retransmissions is lesser than in the previous case as the buffer size in this case is larger.

Our average bandwidth in this case is 91 Mbps, which indicates that most of the bandwidth out of the max bandwidth of 100 Mbps, is used.

iii) 25 Mb



Average Bandwidth = 91.1 Mbps

```

bhavi@bhavi-virtual-machine:~/Downloads/Assignment3/PartC$ sudo python3 MyIperf25.py
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 r1 r2 r3 r4
*** Adding switches:

*** Adding links:
(h1, r1) (h2, r4) (r1, r2) (r1, r3) (r2, r4) (r3, r4)
*** Configuring hosts
h1 h2 r1 r2 r3 r4
*** Starting controller
c0
*** Starting 0 switches

*** Starting CLI:
mininet> h1 ping h2 -c 10
PING 191.7.5.2 (191.7.5.2) 56(84) bytes of data.
64 bytes from 191.7.5.2: icmp_seq=1 ttl=61 time=245 ms
64 bytes from 191.7.5.2: icmp_seq=2 ttl=61 time=184 ms
64 bytes from 191.7.5.2: icmp_seq=3 ttl=61 time=185 ms
64 bytes from 191.7.5.2: icmp_seq=4 ttl=61 time=183 ms
64 bytes from 191.7.5.2: icmp_seq=5 ttl=61 time=185 ms
64 bytes from 191.7.5.2: icmp_seq=6 ttl=61 time=186 ms
64 bytes from 191.7.5.2: icmp_seq=7 ttl=61 time=184 ms
64 bytes from 191.7.5.2: icmp_seq=8 ttl=61 time=185 ms
64 bytes from 191.7.5.2: icmp_seq=9 ttl=61 time=185 ms
64 bytes from 191.7.5.2: icmp_seq=10 ttl=61 time=184 ms

--- 191.7.5.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9009ms
rtt min/avg/max/mdev = 182.667/190.519/245.410/18.317 ms
mininet>

```

Average RTT = 190.519 ms

BDP in this case = Average Bandwidth x Average RTT
= 91.1 Mbps x 190.519 ms = 17.356 Mb

BDP is 17.356 Mb while the buffer size is 25 Mb. Since the buffer size is larger than the BDP, no packets are lost and retransmissions are 0.

Our average bandwidth in this case is 91.1 Mbps, which indicates that most of the bandwidth out of the max bandwidth of 100 Mbps, is used. We observe that the bandwidth in this case is similar to the bandwidth in the previous case.