

Figure 1: Network Topology

## C1:

I created the above topology on mininet using the mininet Python API. It consists of 2 hosts, H1 and H2, connected through a network of 4 routers R1-R4.

In this part, we use these below commands to limit router bandwidth, buffer size, and introduce delays in every interface of all routers with the tc tbfb and tc netem options respectively.

```
tc qdisc add dev <interface> root handle 1: tbf rate <bandwidth> burst <burst_rate> limit <buffer_size>
tc qdisc add dev <interface> parent 1:1 handle 10: netem delay <delay_in_ms>
```

BIRD is used to run the RIP protocol for dynamic routing. Once the routers have been reconfigured with the above commands, we then spawn xterm instances of hosts H1 and H2 to run the network performance tool, lperf, which will simulate a TCP performance test between the two hosts, with H2 acting as the lperf server and H1 the client. We do 3 measurements, with the router buffer size varying as 10Kb, 5Mb and 25Mb across them. Delay = 30ms and Bandwidth = 100Mbps are kept constant in all 3 runs.

To run H2 as the lperf server, we use the below command on its xterm terminal

```
lperf3 -s
```

To run H1 as the lperf client, we use the below command on its xterm terminal

```
lperf3 -c <H2_IP>
```

To create a json output file we can use -J option and pipe the result of both server and client.

We first run the Mylperf.py file in the partC directory that automates the configuration of the routers with the tc commands. To verify that the routers have been configured correctly we can use these two command that is shown in Figure 2.

```

mininet> R1 tc qdisc show
qdisc noqueue 0: dev lo root refcnt 2
qdisc tbf 1: dev r1-eth0 root refcnt 2 rate 100Mbit burst 10Kb lat 0us
qdisc netem 10: dev r1-eth0 parent 1:1 limit 1000 delay 30.0ms
qdisc tbf 1: dev r1-eth1 root refcnt 2 rate 100Mbit burst 10Kb lat 0us
qdisc netem 10: dev r1-eth1 parent 1:1 limit 1000 delay 30.0ms
qdisc tbf 1: dev r1-eth2 root refcnt 2 rate 100Mbit burst 10Kb lat 0us
qdisc netem 10: dev r1-eth2 parent 1:1 limit 1000 delay 30.0ms
mininet> H1 ping -c 10 H2
PING 172.10.6.2 (172.10.6.2) 56(84) bytes of data.
64 bytes from 172.10.6.2: icmp_seq=1 ttl=61 time=181 ms
64 bytes from 172.10.6.2: icmp_seq=2 ttl=61 time=181 ms
64 bytes from 172.10.6.2: icmp_seq=3 ttl=61 time=181 ms
64 bytes from 172.10.6.2: icmp_seq=4 ttl=61 time=181 ms
64 bytes from 172.10.6.2: icmp_seq=5 ttl=61 time=181 ms
64 bytes from 172.10.6.2: icmp_seq=6 ttl=61 time=181 ms
64 bytes from 172.10.6.2: icmp_seq=7 ttl=61 time=181 ms
64 bytes from 172.10.6.2: icmp_seq=8 ttl=61 time=181 ms
64 bytes from 172.10.6.2: icmp_seq=9 ttl=61 time=181 ms
64 bytes from 172.10.6.2: icmp_seq=10 ttl=61 time=181 ms

--- 172.10.6.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9007ms
rtt min/avg/max/mdev = 181.493/181.632/181.978/0.445 ms

```

Figure 2

In the Figure 2, we can see that all interfaces of router R1 have been correctly configured by the tc command to update its bandwidth, buffer\_size and delay parameters. Similarly, we can verify the same for the other routers as well.

Also from the ping command, we see that the RTT delay is now 180ms. Every packet flow through 6 interfaces from H1 to H2 (refer topology diagram in Figure 1, the interfaces are H1 -> r1-eth0 -> r1-eth1 -> r2-eth0 -> r2-eth1 -> r4-eth1 -> r4-eth0 -> H2), and hence total delay is  $6 * 30 = 180\text{ms}$  per packet.

Now that we have setup the routers, we will run iperf between H1 and H2 through their xterms, as shown below. We will then describe the results for buffer\_size = 10Kb, 5Mb and 25Mb.

```

R2 -> H1 H2 R1 R3 R4
R3 -> H1 H2 R1 R2 R4
R4 -> H1 H2 R1 R2 R3
*** Results: 0% dropped (30/30 received)
0.0*** Starting CLI:
mininet> R1 tc qdisc show
qdisc noqueue 0: dev lo root refcnt 2
qdisc tbf 1: dev r1-eth0 root refcnt 2 rate 100Mb
qdisc netem 10: dev r1-eth0 parent 1:1 limit 1000
qdisc tbf 1: dev r1-eth1 root refcnt 2 rate 100Mb
qdisc netem 10: dev r1-eth1 parent 1:1 limit 1000
qdisc tbf 1: dev r1-eth2 root refcnt 2 rate 100Mb
qdisc netem 10: dev r1-eth2 parent 1:1 limit 1000
mininet> H1 ping -c 10 H2
PING 172.10.6.2 (172.10.6.2) 56(84) bytes of data.
64 bytes from 172.10.6.2: icmp_seq=1 ttl=61 time=181 ms
64 bytes from 172.10.6.2: icmp_seq=2 ttl=61 time=181 ms
64 bytes from 172.10.6.2: icmp_seq=3 ttl=61 time=181 ms
64 bytes from 172.10.6.2: icmp_seq=4 ttl=61 time=181 ms
64 bytes from 172.10.6.2: icmp_seq=5 ttl=61 time=181 ms
64 bytes from 172.10.6.2: icmp_seq=6 ttl=61 time=181 ms
64 bytes from 172.10.6.2: icmp_seq=7 ttl=61 time=181 ms
64 bytes from 172.10.6.2: icmp_seq=8 ttl=61 time=181 ms
64 bytes from 172.10.6.2: icmp_seq=9 ttl=61 time=181 ms
64 bytes from 172.10.6.2: icmp_seq=10 ttl=61 time=181 ms

--- 172.10.6.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9015ms
rtt min/avg/max/mdev = 181.553/181.778/182.253/0.223 ms
mininet> xterm H1 H2
mininet>

```

Figure 3

We also capture the client/server result json files for each buffer\_size configuration and store them in the *partC/iperf\_results* directory.

```

mininet> R1 tc qdisc show
qdisc noqueue 0: dev lo root
qdisc tbf 1: dev r1-eth0 root
qdisc netem 10: dev r1-eth0 root
qdisc tbf 1: dev r1-eth1 root
qdisc netem 10: dev r1-eth1 root
qdisc tbf 1: dev r1-eth2 root
qdisc netem 10: dev r1-eth2 root
mininet> H1 ping -c 10 H2
PING 172.10.6.2 (172.10.6.2): 64 bytes of data:
64 bytes from 172.10.6.2: icmp_seq=1 ttl=64 time=0.153 ms
64 bytes from 172.10.6.2: icmp_seq=2 ttl=64 time=0.223 ms
64 bytes from 172.10.6.2: icmp_seq=3 ttl=64 time=0.178 ms
64 bytes from 172.10.6.2: icmp_seq=4 ttl=64 time=0.253 ms
64 bytes from 172.10.6.2: icmp_seq=5 ttl=64 time=0.155 ms
64 bytes from 172.10.6.2: icmp_seq=6 ttl=64 time=0.223 ms
64 bytes from 172.10.6.2: icmp_seq=7 ttl=64 time=0.178 ms
64 bytes from 172.10.6.2: icmp_seq=8 ttl=64 time=0.253 ms
64 bytes from 172.10.6.2: icmp_seq=9 ttl=64 time=0.155 ms
64 bytes from 172.10.6.2: icmp_seq=10 ttl=64 time=0.223 ms
--- 172.10.6.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9015ms
rtt min/avg/max/mdev = 181.553/181.778/182.253/0.223 ms
mininet> xterm H1 H2
mininet>

```

```

Node: H2@mininet-vm
root@mininet-vm:~/hw35# iperf3 -s -J > 10kb_server_output.json
Warning: this system does not seem to support IPv6 - trying IPv4

Node: H1@mininet-vm
root@mininet-vm:~/hw35# ls
10kb_client_output.json  MyIperf.py  R2  R4
10kb_server_output.json  R1          R3  readme.txt
root@mininet-vm:~/hw35#

```

Figure 4

To estimate the actual bandwidth, number of retransmissions and the round-trip times of the entire flow from the 'bits\_per\_second', 'retransmits' and 'mean\_rtt' attributes (in *\$.end.streams[0].sender.\** JSONPath) in the *\*\_client\_output.json* files, as shown in Figure 5. The Bandwidth Delay Product (BDP) can then be calculated as  $BDP = \text{bits\_per\_second} * \text{mean\_rtt}$

```

4      "end": {
5        "streams": [{
6          "sender": {
7            "socket": 6,
8            "start": 0,
9            "end": 10.000387,
10           "seconds": 10.000387,
11           "bytes": 111537784,
12           "bits_per_second": 89226774.540034,
13           "retransmits": 130,
14           "max_snd_cwnd": 3420176,
15           "max_rtt": 281473,
16           "min_rtt": 183816,
17           "mean_rtt": 248779
18         }
19       ]
20     }

```

Figure 5

## Test1. Buffer Size 10Kb

```

mininet> R1 tc qdisc show
qdisc noqueue 0: dev lo root
qdisc tbf 1: dev r1-eth0 root
qdisc netem 10: dev r1-eth0 root
qdisc tbf 1: dev r1-eth1 root
qdisc netem 10: dev r1-eth1 root
qdisc tbf 1: dev r1-eth2 root
qdisc netem 10: dev r1-eth2 root
mininet> H1 ping -c 10 H2
PING 172.10.6.2 (172.10.6.2): 64 bytes of data:
64 bytes from 172.10.6.2: icmp_seq=1 ttl=64 time=0.153 ms
64 bytes from 172.10.6.2: icmp_seq=2 ttl=64 time=0.223 ms
64 bytes from 172.10.6.2: icmp_seq=3 ttl=64 time=0.178 ms
64 bytes from 172.10.6.2: icmp_seq=4 ttl=64 time=0.253 ms
64 bytes from 172.10.6.2: icmp_seq=5 ttl=64 time=0.155 ms
64 bytes from 172.10.6.2: icmp_seq=6 ttl=64 time=0.223 ms
64 bytes from 172.10.6.2: icmp_seq=7 ttl=64 time=0.178 ms
64 bytes from 172.10.6.2: icmp_seq=8 ttl=64 time=0.253 ms
64 bytes from 172.10.6.2: icmp_seq=9 ttl=64 time=0.155 ms
64 bytes from 172.10.6.2: icmp_seq=10 ttl=64 time=0.223 ms
--- 172.10.6.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9015ms
rtt min/avg/max/mdev = 181.553/181.778/182.253/0.223 ms
mininet> xterm H1 H2
mininet>

```

```

Node: H2@mininet-vm
Server listening on 5201
Accepted connection from 172.10.1.2, port 58866
[ ID] Interval      Transfer      Bandwidth
[ 6] 0.00-1.00 sec  181 KBytes  1.48 Mbits/sec
[ 7] 1.00-2.00 sec  4.77 MBytes  40.1 Mbits/sec
[ 6] 2.00-3.00 sec  11.4 MBytes  95.6 Mbits/sec
[ 7] 3.00-4.00 sec  11.4 MBytes  95.5 Mbits/sec
[ 6] 4.00-5.00 sec  11.4 MBytes  95.6 Mbits/sec
[ 7] 5.00-6.00 sec  11.4 MBytes  95.5 Mbits/sec
[ 6] 6.00-7.00 sec  11.4 MBytes  95.6 Mbits/sec
[ 7] 7.00-8.00 sec  11.4 MBytes  95.5 Mbits/sec
[ 6] 8.00-9.00 sec  11.4 MBytes  95.6 Mbits/sec
[ 7] 9.00-10.00 sec 11.4 MBytes  95.5 Mbits/sec
[ 6] 10.00-10.22 sec 2.32 MBytes  86.7 Mbits/sec
[ ID] Interval      Transfer      Bandwidth
[ 7] 0.00-10.22 sec  96.4 MBytes  80.7 Mbits/sec
[ 6] 0.00-10.22 sec  96.4 MBytes  80.7 Mbits/sec
Server listening on 5201

```

```

Node: H1@mininet-vm
root@mininet-vm:~/hw35# iperf3 -c 172.10.6.2
Connecting to host 172.10.6.2, port 5201
[ 6] local 172.10.1.2 port 58866 connected to 172.10.6.2 port 5201
[ ID] Interval      Transfer      Bandwidth      Retr      Cwnd
[ 6] 0.00-1.00 sec  1.26 MBytes  10.6 Mbits/sec    0      196 KBytes
[ 6] 1.00-2.00 sec  16.4 MBytes  137 Mbits/sec    0      2.67 MBytes
[ 6] 2.00-3.00 sec  11.2 MBytes  94.4 Mbits/sec    0      3.24 MBytes
[ 6] 3.00-4.00 sec  11.2 MBytes  94.4 Mbits/sec   119      2.43 MBytes
[ 6] 4.00-5.00 sec  11.2 MBytes  94.4 Mbits/sec    0      2.64 MBytes
[ 6] 5.00-6.00 sec  11.2 MBytes  94.4 Mbits/sec    0      2.82 MBytes
[ 6] 6.00-7.00 sec  11.2 MBytes  94.4 Mbits/sec    0      2.37 MBytes
[ 6] 7.00-8.00 sec  11.2 MBytes  94.4 Mbits/sec    0      3.08 MBytes
[ 6] 8.00-9.00 sec  11.2 MBytes  94.4 Mbits/sec    0      3.19 MBytes
[ 6] 9.00-10.00 sec  11.2 MBytes  94.4 Mbits/sec   16      2.28 MBytes
[ ID] Interval      Transfer      Bandwidth      Retr      Cwnd
[ 6] 0.00-10.00 sec  108 MBytes  90.3 Mbits/sec  135      sender
[ 6] 0.00-10.00 sec  96.4 MBytes  82.6 Mbits/sec      receiver

iperf Done.
root@mininet-vm:~/hw35#

```

Figure 6

$$BDP = 89226774.540034 * 248779 * 10^{-6} = 2.10 \text{ Mbytes}$$

## Test2. Buffer Size 5mb

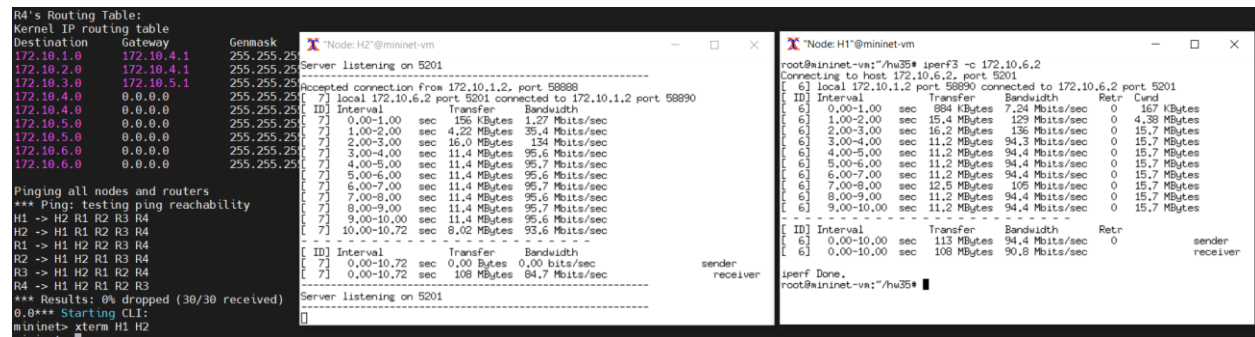


Figure 7

$$\text{BDP} = 94794209.244125 * 595674 * 10^{-6} = 53.85 \text{ Mbytes}$$

## Test3. Buffer Size 25mb

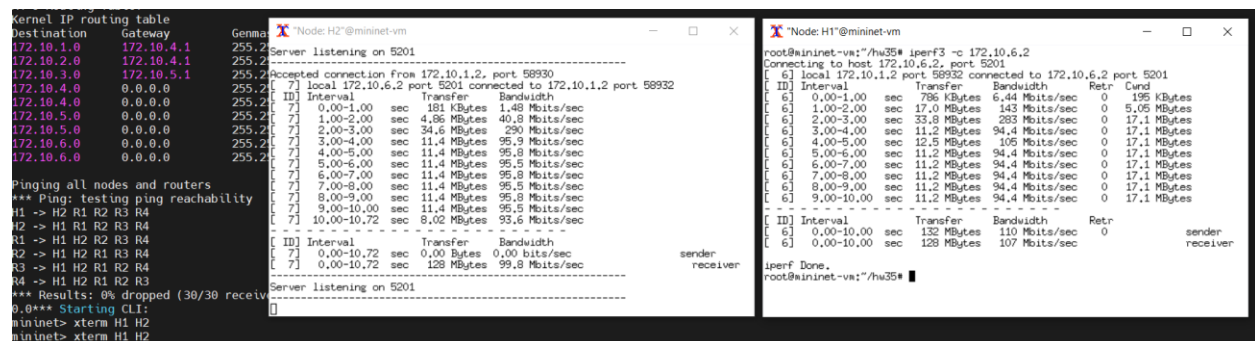


Figure 8

$$\text{BDP} = 110724500.982998 * 544719 * 10^{-6} = 57.51 \text{ Mbytes}$$

## OBSERVATIONS:

1. We see some retransmissions in the first measurement with 10Kb buffer size. This may be due to packets being dropped from one of the router interfaces due to the small-size buffer queue filling up. For the other 2 measurements, we see that there are no retransmissions after the router buffer size was increased.
2. The BDP is increasing with increase in buffer\_size. This is because of increase in the actual average bandwidth (refer Bitrate in H1) with increase in buffer\_size.
3. We see a major multiplicative increase of the cwnd in the first 2 seconds, followed by a sharp drop in the bandwidth (bitrate) which then leads to the cwnd stabilizing across successive intervals.
4. Since retransmission does not occur for buffer\_sizes 5Mb and 25Mb, the retransmission time is 0. For 10Kb, the retransmission time may be around 0.24s (10.24s – 10s)