

Member 1: Aman Kumar Gupta (23210122)

Member 2: Rugved Patil (23210085)

Part I:

Solution to QN 1(a)

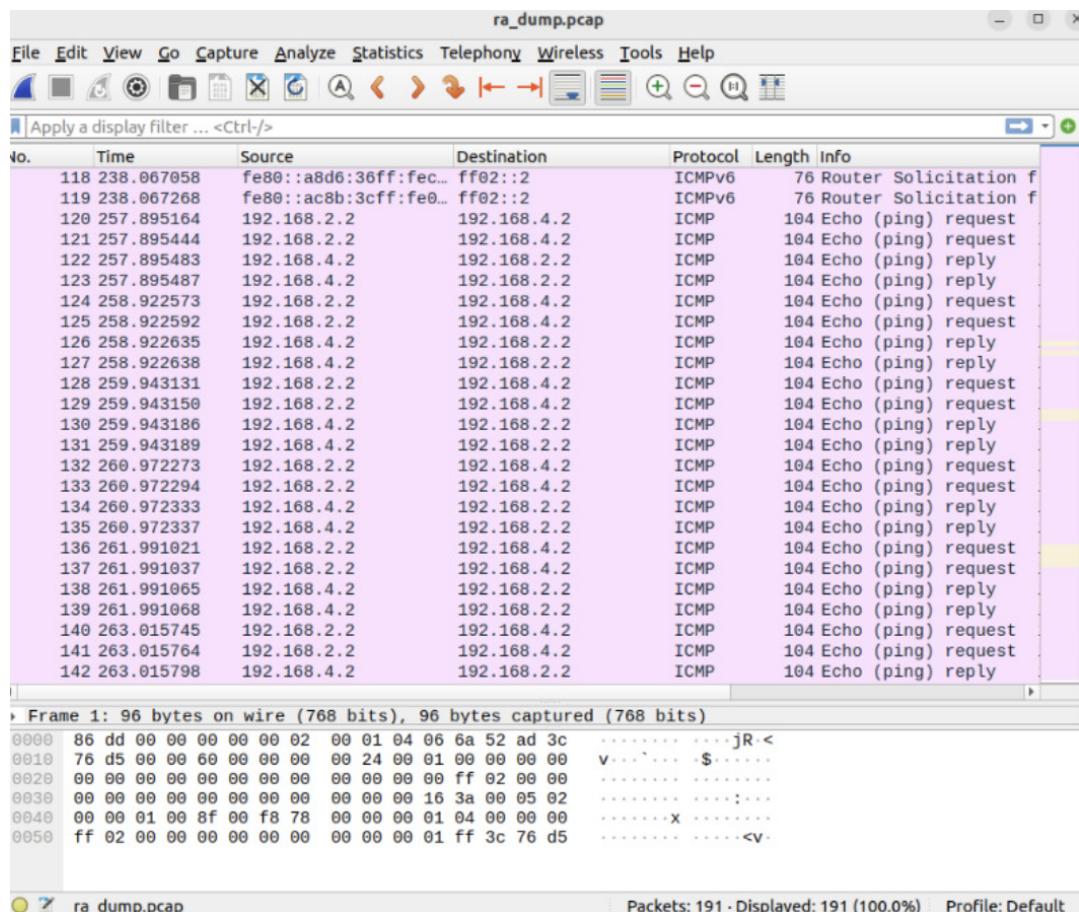
The proof of working of the given topology can be verified with the pingall command in mininet. The screenshot for hosts and routers reachable from all the hosts is presented below:

```
[mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6 rA rB rC
h2 -> h1 h3 h4 h5 h6 rA rB rC
h3 -> h1 h2 h4 h5 h6 rA rB rC
h4 -> h1 h2 h3 h5 h6 rA rB rC
h5 -> h1 h2 h3 h4 h6 rA rB rC
h6 -> h1 h2 h3 h4 h5 rA rB rC
rA -> h1 h2 h3 h4 h5 h6 rB rC
rB -> h1 h2 h3 h4 h5 h6 rA rC
rC -> h1 h2 h3 h4 h5 h6 rA rB
*** Results: 0% dropped (72/72 received)
```

Solution to QN 1(b)

The tcp dump for the ping from h1 to h2 at router-A (rA) has been given below:

```
mininet> h1 ping h3
PING 192.168.4.2 (192.168.4.2) 56(84) bytes of data.
64 bytes from 192.168.4.2: icmp_seq=1 ttl=62 time=0.425 ms
64 bytes from 192.168.4.2: icmp_seq=2 ttl=62 time=0.134 ms
64 bytes from 192.168.4.2: icmp_seq=3 ttl=62 time=0.105 ms
64 bytes from 192.168.4.2: icmp_seq=4 ttl=62 time=0.105 ms
64 bytes from 192.168.4.2: icmp_seq=5 ttl=62 time=0.127 ms
64 bytes from 192.168.4.2: icmp_seq=6 ttl=62 time=0.090 ms
64 bytes from 192.168.4.2: icmp_seq=7 ttl=62 time=0.102 ms
64 bytes from 192.168.4.2: icmp_seq=8 ttl=62 time=0.072 ms
64 bytes from 192.168.4.2: icmp_seq=9 ttl=62 time=0.099 ms
^C
--- 192.168.4.2 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8175ms
rtt min/avg/max/mdev = 0.072/0.139/0.425/0.102 ms
```



Solution to QN 1(c)

Comparing the ‘ping’ and ‘iperf’ that happened for the two different configurations, we can see a significant difference in the latency and bandwidth. For three hops, the average round trip time(rtt) was noted to be **0.208 ms** whereas in the case of four hops, the average round trip time(rtt) was **0.229 ms**. In terms of bandwidth, it dropped from **45.6 Gb/s** to **41.6 Gb/s**.

Iperf and ping for h1->rA->rC->h5

```
[mininet> h5 iperf -s &
[mininet> h1 iperf -c h5
-----
Client connecting to 192.168.6.2, TCP port 5001
TCP window size: 85.3 KByte (default)
-----
[ 1] local 192.168.2.2 port 44540 connected with 192.168.6.2 port 5001
[ ID] Interval Transfer Bandwidth
[ 1] 0.0000-10.0205 sec 53.2 GBytes 45.6 Gbits/sec
[mininet> h1 ping h5
PING 192.168.6.2 (192.168.6.2) 56(84) bytes of data.
64 bytes from 192.168.6.2: icmp_seq=1 ttl=62 time=0.134 ms

64 bytes from 192.168.6.2: icmp_seq=2 ttl=62 time=0.238 ms
64 bytes from 192.168.6.2: icmp_seq=3 ttl=62 time=0.235 ms
64 bytes from 192.168.6.2: icmp_seq=4 ttl=62 time=0.202 ms
64 bytes from 192.168.6.2: icmp_seq=5 ttl=62 time=0.231 ms
^C
--- 192.168.6.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4204ms
rtt min/avg/max/mdev = 0.134/0.208/0.238/0.039 ms
```

Iperf and ping for **h1->rA->rB->rC->h5**

```
[mininet> h5 iperf -s &
[mininet> h1 iperf -c h5
-----
Client connecting to 192.168.6.2, TCP port 5001
TCP window size: 85.3 KByte (default)
-----
[ 1] local 192.168.2.2 port 52508 connected with 192.168.6.2 port 5001
[ ID] Interval      Transfer     Bandwidth
[ 1] 0.0000-10.0425 sec   48.7 GBytes   41.6 Gbits/sec
[mininet> h1 ping h5
PING 192.168.6.2 (192.168.6.2) 56(84) bytes of data.
64 bytes from 192.168.6.2: icmp_seq=1 ttl=62 time=0.214 ms
64 bytes from 192.168.6.2: icmp_seq=2 ttl=62 time=0.218 ms
64 bytes from 192.168.6.2: icmp_seq=3 ttl=62 time=0.246 ms
64 bytes from 192.168.6.2: icmp_seq=4 ttl=62 time=0.213 ms
64 bytes from 192.168.6.2: icmp_seq=5 ttl=62 time=0.256 ms
64 bytes from 192.168.6.2: icmp_seq=6 ttl=62 time=0.277 ms
64 bytes from 192.168.6.2: icmp_seq=7 ttl=62 time=0.185 ms
^C
--- 192.168.6.2 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6247ms
rtt min/avg/max/mdev = 0.185/0.229/0.277/0.028 ms
```

Solution to QN 1(d)

The routing table for two different configurations have been presented below.

Routing table for default case

```
Kernel IP routing table
Destination    Gateway        Genmask       Flags Metric Ref  Use Iface
192.168.2.0    0.0.0.0       255.255.255.0 U     0      0      0 rA-eth1
192.168.3.0    0.0.0.0       255.255.255.0 U     0      0      0 rA-eth2
192.168.4.0    192.168.8.3  255.255.255.0 UG    0      0      0 rA-eth3
192.168.5.0    192.168.8.3  255.255.255.0 UG    0      0      0 rA-eth3
192.168.6.0    192.168.10.2 255.255.255.0 UG    0      0      0 rA-eth4
192.168.7.0    192.168.10.2 255.255.255.0 UG    0      0      0 rA-eth4
192.168.8.0    0.0.0.0       255.255.255.0 U     0      0      0 rA-eth3
192.168.10.0   0.0.0.0       255.255.255.0 U     0      0      0 rA-eth4
Kernel IP routing table
Destination    Gateway        Genmask       Flags Metric Ref  Use Iface
192.168.2.0    192.168.8.2  255.255.255.0 UG    0      0      0 rB-eth3
192.168.3.0    192.168.8.2  255.255.255.0 UG    0      0      0 rB-eth3
192.168.4.0    0.0.0.0       255.255.255.0 U     0      0      0 rB-eth1
192.168.5.0    0.0.0.0       255.255.255.0 U     0      0      0 rB-eth2
192.168.6.0    192.168.9.3  255.255.255.0 UG    0      0      0 rB-eth4
192.168.7.0    192.168.9.3  255.255.255.0 UG    0      0      0 rB-eth4
192.168.8.0    0.0.0.0       255.255.255.0 U     0      0      0 rB-eth3
192.168.9.0    0.0.0.0       255.255.255.0 U     0      0      0 rB-eth4
Kernel IP routing table
Destination    Gateway        Genmask       Flags Metric Ref  Use Iface
192.168.2.0    192.168.10.3 255.255.255.0 UG    0      0      0 rC-eth4
192.168.3.0    192.168.10.3 255.255.255.0 UG    0      0      0 rC-eth4
192.168.4.0    192.168.9.2  255.255.255.0 UG    0      0      0 rC-eth3
192.168.5.0    192.168.9.2  255.255.255.0 UG    0      0      0 rC-eth3
192.168.6.0    0.0.0.0       255.255.255.0 U     0      0      0 rC-eth1
192.168.7.0    0.0.0.0       255.255.255.0 U     0      0      0 rC-eth2
192.168.9.0    0.0.0.0       255.255.255.0 U     0      0      0 rC-eth3
192.168.10.0   0.0.0.0       255.255.255.0 U     0      0      0 rC-eth4
```

Routing table for clockwise packet forwarding

Kernel IP routing table							
Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
192.168.2.0	0.0.0.0	255.255.255.0	U	0	0	0	ra-eth1
192.168.3.0	0.0.0.0	255.255.255.0	U	0	0	0	ra-eth2
192.168.4.0	192.168.8.3	255.255.255.0	UG	0	0	0	ra-eth3
192.168.5.0	192.168.8.3	255.255.255.0	UG	0	0	0	ra-eth3
192.168.6.0	192.168.8.3	255.255.255.0	UG	0	0	0	ra-eth3
192.168.7.0	192.168.8.3	255.255.255.0	UG	0	0	0	ra-eth3
192.168.8.0	0.0.0.0	255.255.255.0	U	0	0	0	ra-eth3
192.168.10.0	0.0.0.0	255.255.255.0	U	0	0	0	ra-eth4
Kernel IP routing table							
Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
192.168.2.0	192.168.9.3	255.255.255.0	UG	0	0	0	rb-eth4
192.168.3.0	192.168.9.3	255.255.255.0	UG	0	0	0	rb-eth4
192.168.4.0	0.0.0.0	255.255.255.0	U	0	0	0	rb-eth1
192.168.5.0	0.0.0.0	255.255.255.0	U	0	0	0	rb-eth2
192.168.6.0	192.168.9.3	255.255.255.0	UG	0	0	0	rb-eth4
192.168.7.0	192.168.9.3	255.255.255.0	UG	0	0	0	rb-eth4
192.168.8.0	0.0.0.0	255.255.255.0	U	0	0	0	rb-eth3
192.168.9.0	0.0.0.0	255.255.255.0	U	0	0	0	rb-eth4
Kernel IP routing table							
Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
192.168.2.0	192.168.10.3	255.255.255.0	UG	0	0	0	rc-eth4
192.168.3.0	192.168.10.3	255.255.255.0	UG	0	0	0	rc-eth4
192.168.4.0	192.168.10.3	255.255.255.0	UG	0	0	0	rc-eth4
192.168.5.0	192.168.10.3	255.255.255.0	UG	0	0	0	rc-eth4
192.168.6.0	0.0.0.0	255.255.255.0	U	0	0	0	rc-eth1
192.168.7.0	0.0.0.0	255.255.255.0	U	0	0	0	rc-eth2
192.168.9.0	0.0.0.0	255.255.255.0	U	0	0	0	rc-eth3
192.168.10.0	0.0.0.0	255.255.255.0	U	0	0	0	rc-eth4

Part II

Solution to QN 2(a)

The following table tells us about the parameters and their nature:

Parameter	Nature	Possible Values
config	compulsory	b, c
scheme	optional	reno, vegas, cubic, bbr
loss	optional	1 to 100 (percent)

```
• msiddhartha@msiddhartha-VirtualBox:~/Computer_System/test$ sudo python3 congestionTest.py
usage: congestionTest.py [-h] --config {b,c} [--scheme {reno,vegas,cubic,bbr}] [--loss LOSS]
congestionTest.py: error: the following arguments are required: --config
```

Solution to QN 2(b)

The observed throughput differences among Congestion Control Algorithms (CCAs) stem from their unique mechanisms for addressing network congestion. Reno, the default TCP CCA, employs a loss-based approach, but its conservative rate reduction can limit throughput. Vegas considers both packet loss and delay, potentially leading to higher throughput. Cubic utilizes loss, delay, and packet arrival times for quicker rate adjustments, resulting in higher throughput than Vegas. BBR, Google's CCA, uses a delay-based approach for optimal sending rates, achieving high throughput in certain conditions, but its implementation complexity and potential negative impact from iperf testing methodology may affect results.

Scheme reno

```
• msiddhartha@msiddhartha-VirtualBox:~/Computer_System/test$ sudo python3 a.py --config b --scheme reno --loss 0
h1Client Output:
-----
Client connecting to 10.0.0.4, TCP port 5001
TCP congestion control set to reno
TCP window size: 85.3 KByte (default)
-----
[ 1] local 10.0.0.1 port 37444 connected with 10.0.0.4 port 5001
[ ID] Interval      Transfer     Bandwidth
[ 1] 0.0000-21.7669 sec   8.92 MBytes  3.44 Mbits/sec
```

Scheme bbr

```
• msiddhartha@msiddhartha-VirtualBox:~/Computer_System/test$ sudo python3 a.py --config b --scheme bbr --loss 0
h1Client Output:
-----
Client connecting to 10.0.0.4, TCP port 5001
TCP congestion control set to bbr
TCP window size: 128 KByte (default)
-----
[ 1] local 10.0.0.1 port 33124 connected with 10.0.0.4 port 5001
[ ID] Interval      Transfer     Bandwidth
[ 1] 0.0000-11.5994 sec   2.50 MBytes  1.81 Mbits/sec
```

Scheme cubic

```
• msiddhartha@msiddhartha-VirtualBox:~/Computer_System/test$ sudo python3 a.py --config b --scheme cubic --loss 0
h1Client Output:
-----
Client connecting to 10.0.0.4, TCP port 5001
TCP congestion control set to cubic
TCP window size: 85.3 KByte (default)
-----
[ 1] local 10.0.0.1 port 56316 connected with 10.0.0.4 port 5001
[ ID] Interval      Transfer     Bandwidth
[ 1] 0.0000-17.4826 sec   3.50 MBytes  1.68 Mbits/sec
```

Scheme vegas

```

• msiddhartha@msiddhartha-VirtualBox:~/Computer_System/test$ sudo python3 a.py --config b --scheme vegas --loss 0
h1Client Output:
-----
Client connecting to 10.0.0.4, TCP port 5001
TCP congestion control set to vegas
TCP window size: 85.3 KByte (default)
-----
[ 1] local 10.0.0.1 port 33956 connected with 10.0.0.4 port 5001
[ ID] Interval Transfer Bandwidth
[ 1] 0.0000-11.3301 sec 2.50 MBytes 1.85 Mbits/sec

```

Congestion Choice	Data Transferred	Speed
reno	8.92 MB	3.44 Mb/s
bbr	2.50 MB	1.85 Mb/s
cubic	3.50 MB	1.68 Mb/s
vegas	2.50 MB	1.85 Mb/s

Solution to QN 2(c)

Results were consistent across all strategies. Despite the diversity of CCAs, the akin outcomes may arise from the internal implementation methodology of iperf.

```

Content of 10.0.0.1_output.txt:
-----
Client connecting to 10.0.0.4, TCP port 5001
TCP congestion control set to reno
TCP window size: 85.3 KByte (default)
-----
[ 1] local 10.0.0.1 port 55834 connected with 10.0.0.4 port 5001
[ ID] Interval Transfer Bandwidth
[ 1] 0.0000-10.0757 sec 13.4 GBytes 11.4 Gbits/sec

Content of 10.0.0.2_output.txt:
-----
Client connecting to 10.0.0.4, TCP port 5001
TCP congestion control set to reno
TCP window size: 85.3 KByte (default)
-----
[ 1] local 10.0.0.2 port 53998 connected with 10.0.0.4 port 5001
[ ID] Interval Transfer Bandwidth
[ 1] 0.0000-10.0441 sec 12.0 GBytes 10.2 Gbits/sec

Content of 10.0.0.3_output.txt:
-----
Client connecting to 10.0.0.4, TCP port 5001
TCP congestion control set to reno
TCP window size: 85.3 KByte (default)
-----
[ 1] local 10.0.0.3 port 37958 connected with 10.0.0.4 port 5001
[ ID] Interval Transfer Bandwidth
[ 1] 0.0000-10.0245 sec 13.1 GBytes 11.3 Gbits/sec

```

```
Content of 10.0.0.1_output.txt:  
-----  
Client connecting to 10.0.0.4, TCP port 5001  
TCP congestion control set to bbr  
TCP window size: 128 KByte (default)  
-----  
[ 1] local 10.0.0.1 port 51654 connected with 10.0.0.4 port 5001  
[ ID] Interval Transfer Bandwidth  
[ 1] 0.0000-10.1848 sec 840 MBytes 692 Mbits/sec  
  
Content of 10.0.0.2_output.txt:  
-----  
Client connecting to 10.0.0.4, TCP port 5001  
TCP congestion control set to bbr  
TCP window size: 128 KByte (default)  
-----  
[ 1] local 10.0.0.2 port 59484 connected with 10.0.0.4 port 5001  
[ ID] Interval Transfer Bandwidth  
[ 1] 0.0000-10.1461 sec 935 MBytes 773 Mbits/sec  
  
Content of 10.0.0.3_output.txt:  
-----  
Client connecting to 10.0.0.4, TCP port 5001  
TCP congestion control set to bbr  
TCP window size: 128 KByte (default)  
-----  
[ 1] local 10.0.0.3 port 55296 connected with 10.0.0.4 port 5001  
[ ID] Interval Transfer Bandwidth  
[ 1] 0.0000-10.1861 sec 1.07 GBytes 905 Mbits/sec
```

```
Content of 10.0.0.1_output.txt:  
-----  
Client connecting to 10.0.0.4, TCP port 5001  
TCP congestion control set to vegas  
TCP window size: 85.3 KByte (default)  
-----  
[ 1] local 10.0.0.1 port 59416 connected with 10.0.0.4 port 5001  
[ ID] Interval Transfer Bandwidth  
[ 1] 0.0000-10.0349 sec 11.3 GBytes 9.66 Gbits/sec  
  
Content of 10.0.0.2_output.txt:  
-----  
Client connecting to 10.0.0.4, TCP port 5001  
TCP congestion control set to vegas  
TCP window size: 85.3 KByte (default)  
-----  
[ 1] local 10.0.0.2 port 36684 connected with 10.0.0.4 port 5001  
[ ID] Interval Transfer Bandwidth  
[ 1] 0.0000-10.0075 sec 9.11 GBytes 7.82 Gbits/sec  
  
Content of 10.0.0.3_output.txt:  
-----  
Client connecting to 10.0.0.4, TCP port 5001  
TCP congestion control set to vegas  
TCP window size: 85.3 KByte (default)  
-----  
[ 1] local 10.0.0.3 port 36230 connected with 10.0.0.4 port 5001  
[ ID] Interval Transfer Bandwidth  
[ 1] 0.0000-10.0275 sec 11.5 GBytes 9.87 Gbits/sec
```

```
Content of 10.0.0.1_output.txt:  
-----  
Client connecting to 10.0.0.4, TCP port 5001  
TCP congestion control set to cubic  
TCP window size: 85.3 KByte (default)  
-----  
[ 1] local 10.0.0.1 port 33580 connected with 10.0.0.4 port 5001  
[ ID] Interval Transfer Bandwidth  
[ 1] 0.0000-10.0491 sec 9.92 GBytes 8.48 Gbits/sec  
  
Content of 10.0.0.2_output.txt:  
-----  
Client connecting to 10.0.0.4, TCP port 5001  
TCP congestion control set to cubic  
TCP window size: 85.3 KByte (default)  
-----  
[ 1] local 10.0.0.2 port 38268 connected with 10.0.0.4 port 5001  
[ ID] Interval Transfer Bandwidth  
[ 1] 0.0000-10.0173 sec 9.61 GBytes 8.24 Gbits/sec  
  
Content of 10.0.0.3_output.txt:  
-----  
Client connecting to 10.0.0.4, TCP port 5001  
TCP congestion control set to cubic  
TCP window size: 85.3 KByte (default)  
-----  
[ 1] local 10.0.0.3 port 36100 connected with 10.0.0.4 port 5001  
[ ID] Interval Transfer Bandwidth  
[ 1] 0.0000-10.0111 sec 12.9 GBytes 11.1 Gbits/sec
```

Congestion Scheme	Host Identity	Data Transferred	Bandwidth
reno	h1	13.4 GB	11.4 Gb/s
	h2	12.0 GB	10.2 Gb/s
	h3	13.1 GB	11.3 Gb/s
bbr	h1	840 MB	692 Mb/s
	h2	935 MB	773 Mb/s
	h3	1.07 GB	905 Mb/s
vegas	h1	11.3 GB	9.66 Gb/s
	h2	9.11 GB	7.82 Gb/s
	h3	11.5 GB	9.87 Gb/s
cubic	h1	9.92 GB	8.48 Gb/s
	h2	9.61 GB	8.42 Gb/s
	h3	12.9 GB	11.1 Gb/s

Solution to QN 2(d)

Below we have presented screenshots for running various congestion control algorithms on config b with loss set once to 1 percent and the other time to 3 percent. The table shown at the very bottom is the summary of the speed observed. ‘reno’ has suffered the most when the loss is changed from 1 to 3 percent while ‘bbr’ has the least effect of increase in loss.

```
msiddhartha@msiddhartha-VirtualBox:~/Computer_System/test$ sudo python3 a.py --config b --scheme vegas --loss 1
h1Client Output:
-----
Client connecting to 10.0.0.4, TCP port 5001
TCP congestion control set to vegas
TCP window size: 85.3 KByte (default)
-----
[ 1] local 10.0.0.1 port 43544 connected with 10.0.0.4 port 5001
[ ID] Interval Transfer Bandwidth
[ 1] 0.0000-11.4210 sec 2.13 MBytes 1.56 Mbits/sec
```

```
msiddhartha@msiddhartha-VirtualBox:~/Computer_System/test$ sudo python3 a.py --config b --scheme vegas --loss 3
h1Client Output:
-----
Client connecting to 10.0.0.4, TCP port 5001
TCP congestion control set to vegas
TCP window size: 85.3 KByte (default)
-----
[ 1] local 10.0.0.1 port 55574 connected with 10.0.0.4 port 5001
[ ID] Interval Transfer Bandwidth
[ 1] 0.0000-11.7741 sec 2.38 MBytes 1.69 Mbits/sec
```

```

• msiddhartha@msiddhartha-VirtualBox:~/Computer_System/test$ sudo python3 a.py --config b --scheme bbr --loss 3
h1Client Output:
-----
Client connecting to 10.0.0.4, TCP port 5001
TCP congestion control set to bbr
TCP window size: 128 KByte (default)
-----
[ 1] local 10.0.0.1 port 45892 connected with 10.0.0.4 port 5001
[ ID] Interval      Transfer     Bandwidth
[ 1] 0.0000-11.4396 sec  2.50 MBytes  1.83 Mbits/sec

• msiddhartha@msiddhartha-VirtualBox:~/Computer_System/test$ sudo python3 a.py --config b --scheme bbr --loss 1
h1Client Output:
-----
Client connecting to 10.0.0.4, TCP port 5001
TCP congestion control set to bbr
TCP window size: 128 KByte (default)
-----
[ 1] local 10.0.0.1 port 56872 connected with 10.0.0.4 port 5001
[ ID] Interval      Transfer     Bandwidth
[ 1] 0.0000-11.3156 sec  2.50 MBytes  1.85 Mbits/sec

• msiddhartha@msiddhartha-VirtualBox:~/Computer_System/test$ sudo python3 a.py --config b --scheme cubic --loss 3
h1Client Output:
-----
Client connecting to 10.0.0.4, TCP port 5001
TCP congestion control set to cubic
TCP window size: 85.3 KByte (default)
-----
[ 1] local 10.0.0.1 port 43662 connected with 10.0.0.4 port 5001
[ ID] Interval      Transfer     Bandwidth
[ 1] 0.0000-10.6364 sec  2.25 MBytes  1.77 Mbits/sec

• msiddhartha@msiddhartha-VirtualBox:~/Computer_System/test$ sudo python3 a.py --config b --scheme cubic --loss 1
h1Client Output:
-----
Client connecting to 10.0.0.4, TCP port 5001
TCP congestion control set to cubic
TCP window size: 85.3 KByte (default)
-----
[ 1] local 10.0.0.1 port 58068 connected with 10.0.0.4 port 5001
[ ID] Interval      Transfer     Bandwidth
[ 1] 0.0000-12.0267 sec  2.63 MBytes  1.83 Mbits/sec

• msiddhartha@msiddhartha-VirtualBox:~/Computer_System/test$ sudo python3 a.py --config b --scheme reno --loss 3
h1Client Output:
-----
Client connecting to 10.0.0.4, TCP port 5001
TCP congestion control set to reno
TCP window size: 85.3 KByte (default)
-----
[ 1] local 10.0.0.1 port 48228 connected with 10.0.0.4 port 5001
[ ID] Interval      Transfer     Bandwidth
[ 1] 0.0000-12.5932 sec  2.25 MBytes  1.50 Mbits/sec

• msiddhartha@msiddhartha-VirtualBox:~/Computer_System/test$ sudo python3 a.py --config b --scheme reno --loss 1
h1Client Output:
-----
Client connecting to 10.0.0.4, TCP port 5001
TCP congestion control set to reno
TCP window size: 85.3 KByte (default)
-----
[ 1] local 10.0.0.1 port 51402 connected with 10.0.0.4 port 5001
[ ID] Interval      Transfer     Bandwidth
[ 1] 0.0000-11.4203 sec  2.50 MBytes  1.84 Mbits/sec

```

Congestion Choice	Loss = 1%	Loss = 3%
reno	1.84 Mb/s	1.50 Mb/s
cubic	1.83 Mb/s	1.77 Mb/s
bbr	1.85 Mb/s	1.83 Mb/s
vegas	1.69 Mb/s	1.56 Mb/s