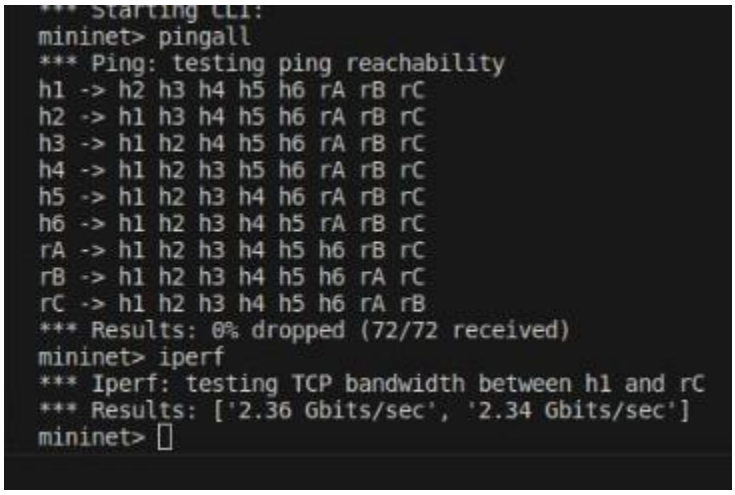
Member 1: M Siddhartha(23210126)
Member 2: Alay Patel (23210010)

Part I:

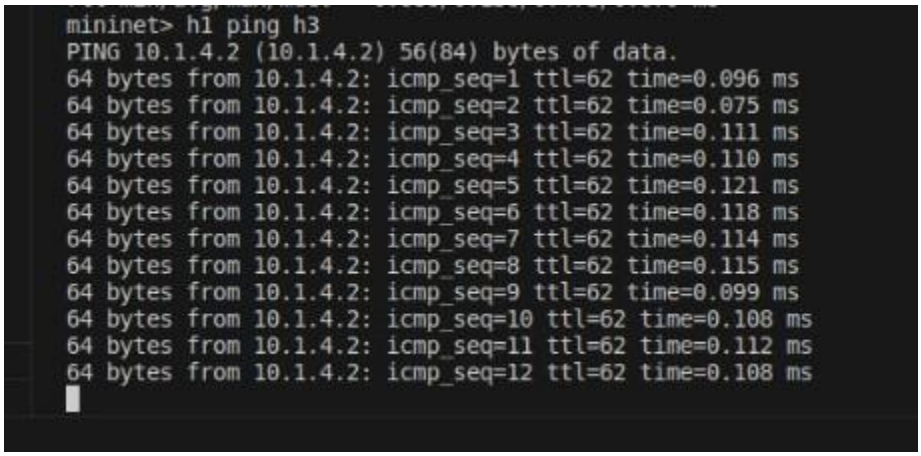
Solution to QN 1(a)

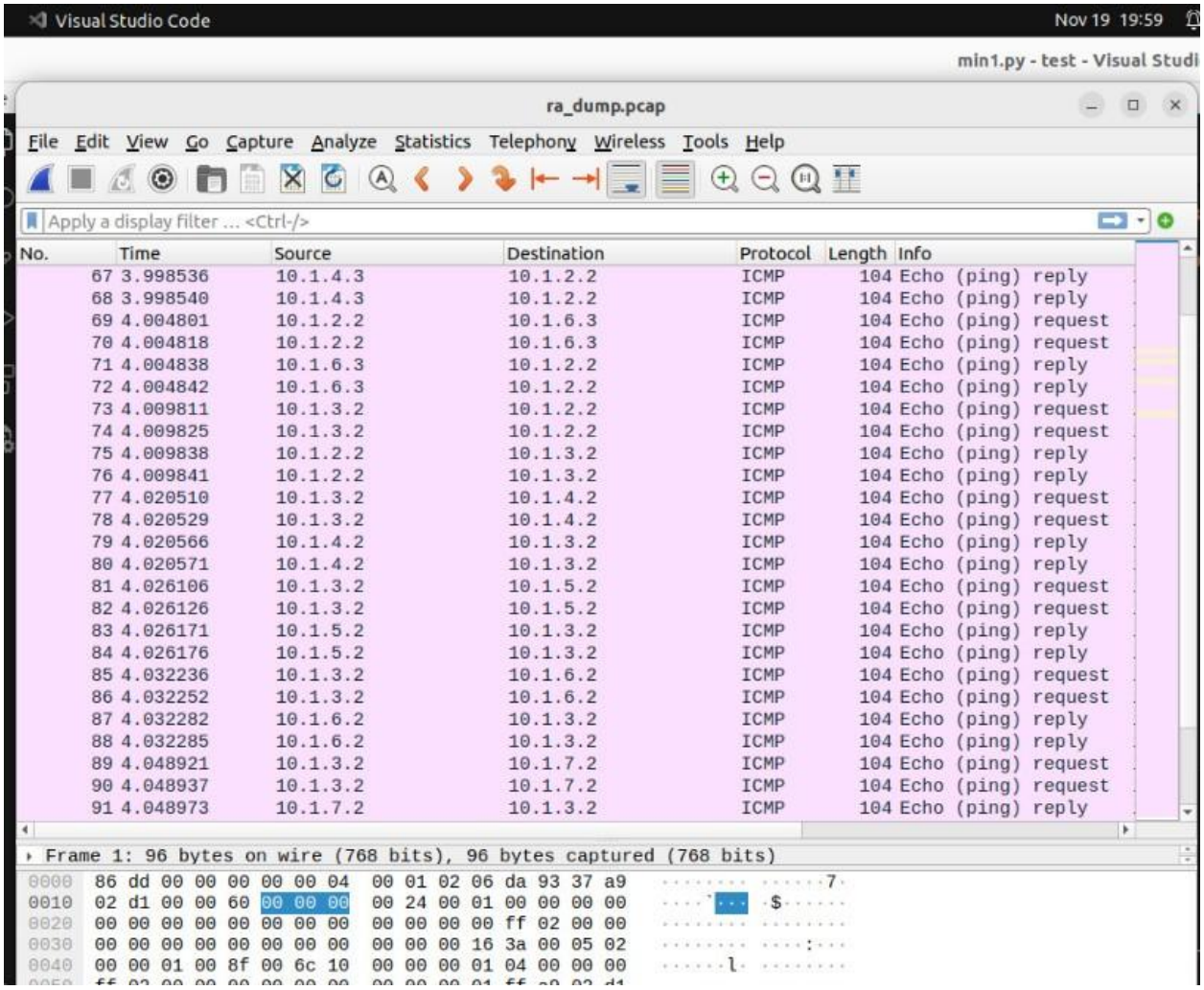
The connection between each and every host is shown below, with the pingall command where we can see that there is 0% drop rate.



Solution to QN 1(b)

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





Solution to QN 1(c)

When we compared 'ping' and 'iperf' for two setups, we observed notable differences in both latency and bandwidth. With three hops, the average time taken for a round trip was 0.208 milliseconds, while for four hops, it increased slightly to 0.229 milliseconds. In terms of data transfer speed, the bandwidth decreased from 14.5 gigabits per second (Gb/s) to 12.4 Gb/s when we added an extra hop in the network path.

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

```
*** Starting CLI:
mininet> h5 iperf -s &
mininet> h1 iperf -c h5
------------------------------------------------------------
Client connecting to 10.1.6.2, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  1] local 10.1.2.2 port 56030 connected with 10.1.6.2 port 5001
[ ID] Interval       Transfer     Bandwidth
[  1] 0.0000-10.0195 sec  14.5 GBytes  12.4 Gbits/sec
mininet> h1 ping h5
PING 10.1.6.2 (10.1.6.2) 56(84) bytes of data.
64 bytes from 10.1.6.2: icmp_seq=1 ttl=62 time=0.090 ms
64 bytes from 10.1.6.2: icmp_seq=2 ttl=62 time=0.221 ms
64 bytes from 10.1.6.2: icmp_seq=3 ttl=62 time=0.083 ms
64 bytes from 10.1.6.2: icmp_seq=4 ttl=62 time=0.085 ms
64 bytes from 10.1.6.2: icmp_seq=5 ttl=62 time=0.071 ms
64 bytes from 10.1.6.2: icmp_seq=6 ttl=62 time=0.085 ms
64 bytes from 10.1.6.2: icmp_seq=7 ttl=62 time=0.091 ms
64 bytes from 10.1.6.2: icmp_seq=8 ttl=62 time=0.080 ms
^C
--- 10.1.6.2 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7249ms
```

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
10.1.2.0        0.0.0.0         255.255.255.0   U     0      0        0 rA-eth1
10.1.3.0        0.0.0.0         255.255.255.0   U     0      0        0 rA-eth2
10.1.4.0        10.1.8.3        255.255.255.0   UG    0      0        0 rA-eth3
10.1.5.0        10.1.8.3        255.255.255.0   UG    0      0        0 rA-eth3
10.1.6.0        10.1.10.2       255.255.255.0   UG    0      0        0 rA-eth4
10.1.7.0        10.1.10.2       255.255.255.0   UG    0      0        0 rA-eth4
10.1.8.0        0.0.0.0         255.255.255.0   U     0      0        0 rA-eth3
10.1.10.0       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
10.1.2.0        10.1.8.2        255.255.255.0   UG    0      0        0 rB-eth3
10.1.3.0        10.1.8.2        255.255.255.0   UG    0      0        0 rB-eth3
10.1.4.0        0.0.0.0         255.255.255.0   U     0      0        0 rB-eth1
10.1.5.0        0.0.0.0         255.255.255.0   U     0      0        0 rB-eth2
10.1.6.0        10.1.9.3        255.255.255.0   UG    0      0        0 rB-eth4
10.1.7.0        10.1.9.3        255.255.255.0   UG    0      0        0 rB-eth4
10.1.8.0        0.0.0.0         255.255.255.0   U     0      0        0 rB-eth3
10.1.9.0        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
10.1.2.0        10.1.10.3       255.255.255.0   UG    0      0        0 rC-eth4
10.1.3.0        10.1.10.3       255.255.255.0   UG    0      0        0 rC-eth4
10.1.4.0        10.1.9.2        255.255.255.0   UG    0      0        0 rC-eth3
10.1.5.0        10.1.9.2        255.255.255.0   UG    0      0        0 rC-eth3
10.1.6.0        0.0.0.0         255.255.255.0   U     0      0        0 rC-eth1
10.1.7.0        0.0.0.0         255.255.255.0   U     0      0        0 rC-eth2
10.1.9.0        0.0.0.0         255.255.255.0   U     0      0        0 rC-eth3
10.1.10.0       0.0.0.0         255.255.255.0   U     0      0        0 rC-eth4
Run the commands as per CLI_scripts.md :
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6 rA rB rC
```

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)

Different congestion control algorithms (CCAs) manage network congestion in various ways, affecting how fast data can move through a network. Reno, the basic TCP CCA, lowers its speed when it detects data loss, which can sometimes slow things down too much. Vegas looks at both data loss and delays, often leading to faster speeds. Cubic adjusts its speed by considering loss, delays, and when packets arrive, making it faster than Vegas. BBR, Google's CCA, focuses on delays to find the best speeds, which can be really fast in certain situations. But it's a bit complicated to use and might not perform well under certain test conditions like iperf testing.

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

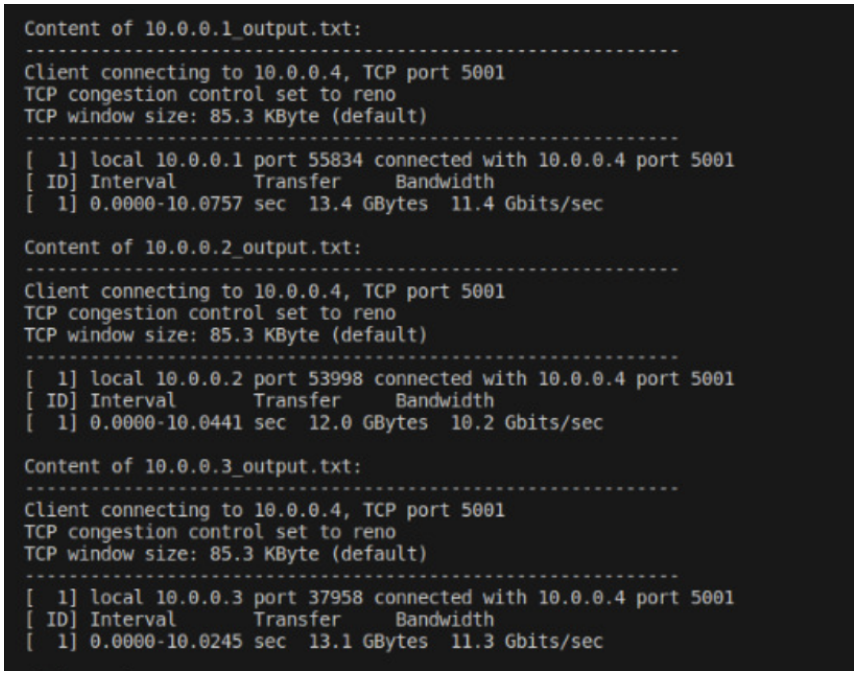
```
msiddhartha@msiddhartha-VirtualBox:~/Computer_System/test$ sudo python3 a.py --config b --scheme vegas --loss 0
h1Client Output:
------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5001
TCP congestion control set to vegas
TCP window size: 85.3 KByte (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 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 Transfered	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