Assignment 6

Team members:

Tanmay M Saurav, Roll:22210062

Tanmay Somkuwar, Roll:23210112

1.Question

a) Implementation:

Topology setup: Three routers (ra, rb, and rc) connect hosts h1 and h2 to ra through switch s1, hosts h3 and h4 to rb via switch s2, and hosts h5 and h6 to rc through switch s3, while the routers are interconnected through switch s4.

IP Configurations:

- 192.168.1.0/24 (ra-h1-h2, IP: 192.168.1.{1, 100, 101})
- 192.168.2.0/24 (rb-h3-h4, IP: 192.168.2.{1, 100, 101})
- 192.168.3.0/24 (rc-h5-h6, IP: 192.168.3.{1, 100, 101})
- 192.168.4.0/24 (ra-rb-rc, IP: 192.168.4.{1, 100, 101})

Working with the network topology:

We have successfully set up routers (ra, rb, and rc), switches (s1, s2, s3, and s4), and hosts (h1, h2, h3, h4, h5, and h6). As part of our validation process, we conducted a ping test between all hosts and routers to confirm the proper functionality and communication within the network:

```
*** Done
mininet@mininet-vm:~/workingdir$ vim topo1.py
mininet@mininet-vm:~/workingdir$ sudo python topo1.py
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6 ra rb rc
*** Adding switches:
s1 s2 s3 s4
*** Adding links:
(h1, s1) (h2, s1) (h3, s2) (h4, s2) (h5, s3) (h6, s3) (s1, ra) (s2, rb) (s3, rc) (s4, ra) (s4, rb) (s4, rc)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 ra rb rc
*** Starting controller
c0
*** Starting 4 switches
s1 s2 s3 s4 ...
*** Waiting for switches to connect
s1 s2 s3 s4
*** Starting CLI:
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)
mininet> iperf h1 h3
*** Iperf: testing TCP bandwidth between h1 and h3
```

Dumping:-

```
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)
mininet> dump
<Host h1: h1-eth0:192.168.1.100 pid=4091>
<Host h2: h2-eth0:192.168.1.101 pid=4093>
<Host h3: h3-eth0:192.168.2.100 pid=4095>
<Host h4: h4-eth0:192.168.2.101 pid=4097>
<Host h5: h5-eth0:192.168.3.100 pid=4099>
<Host h6: h6-eth0:192.168.3.101 pid=4101>
<LinuxRouter ra: ra-eth1:192.168.1.1,ra-eth2:192.168.4.1 pid=4105>
<LinuxRouter rb: rb-eth1:192.168.2.1,rb-eth2:192.168.4.2 pid=4107>
<LinuxRouter rc: rc-eth1:192.168.3.1,rc-eth2:192.168.4.3 pid=4109>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None,s1-eth3:None pid=4114>
<OVSSwitch s2: lo:127.0.0.1,s2-eth1:None,s2-eth2:None,s2-eth3:None pid=4117>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None,s3-eth3:None pid=4120>
<OVSSwitch s4: lo:127.0.0.1,s4-eth1:None,s4-eth2:None,s4-eth3:None pid=4123>
<Controller c0: 127.0.0.1:6653 pid=4084>
```

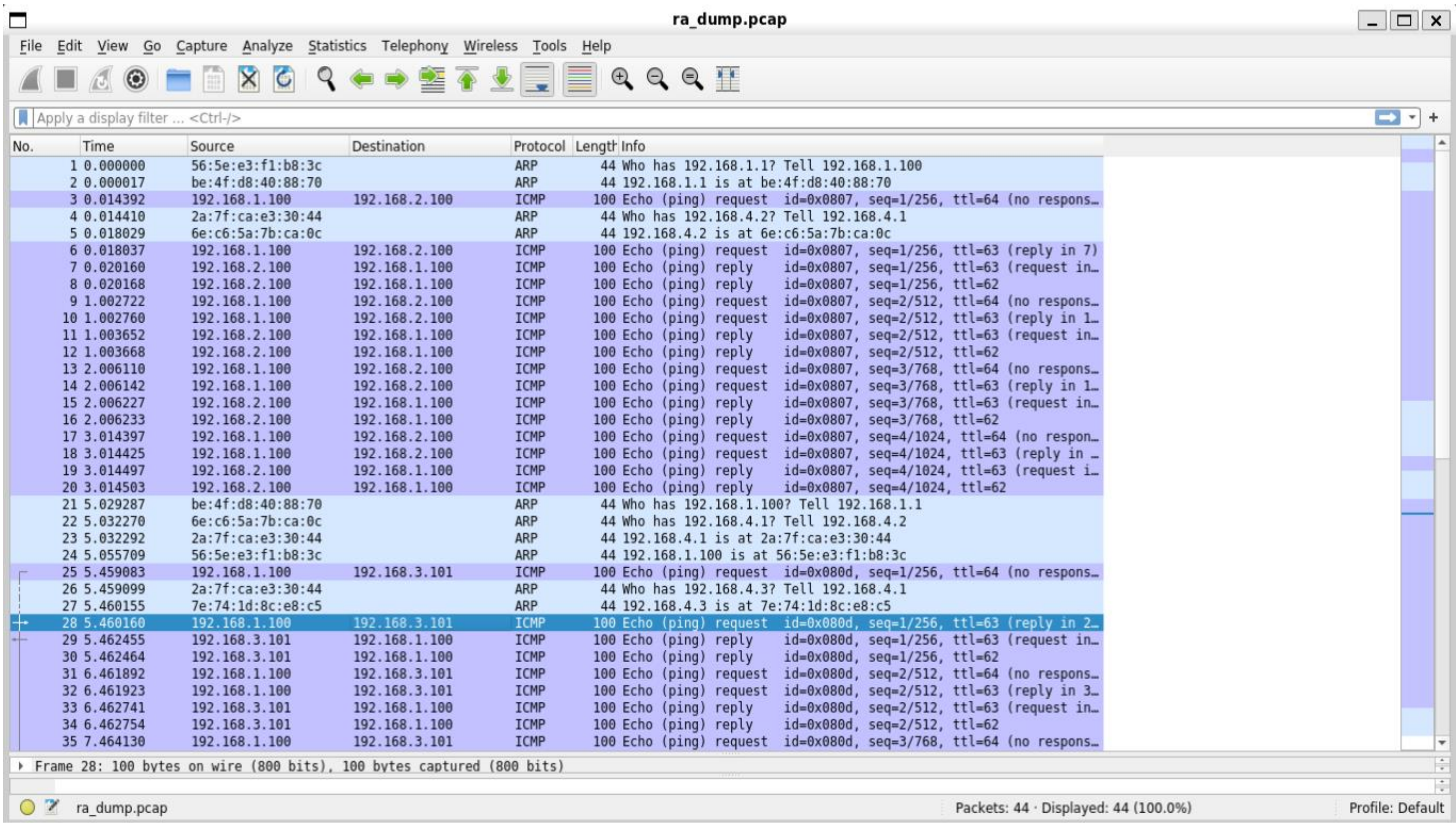
Question **b) Observations:** Pinging 4 packets from h1 to h3 and h1 to h6.

```
mininet> h1 ping -c 4 h3
PING 192.168.2.100 (192.168.2.100) 56(84) bytes of data.
64 bytes from 192.168.2.100: icmp_seq=1 ttl=62 time=13.9 ms
64 bytes from 192.168.2.100: icmp_seq=2 ttl=62 time=6.00 ms
64 bytes from 192.168.2.100: icmp_seq=3 ttl=62 time=1.12 ms
64 bytes from 192.168.2.100: icmp_seq=4 ttl=62 time=0.061 ms

--- 192.168.2.100 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 0.061/5.266/13.876/5.452 ms
mininet> h1 ping -c 4 h6
PING 192.168.3.101 (192.168.3.101) 56(84) bytes of data.
64 bytes from 192.168.3.101: icmp_seq=1 ttl=62 time=14.9 ms
64 bytes from 192.168.3.101: icmp_seq=2 ttl=62 time=3.09 ms
64 bytes from 192.168.3.101: icmp_seq=3 ttl=62 time=0.448 ms
64 bytes from 192.168.3.101: icmp_seq=4 ttl=62 time=0.104 ms

--- 192.168.3.101 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3018ms
rtt min/avg/max/mdev = 0.104/4.638/14.912/6.042 ms
mininet>
```

Upon initiating tcpdump through Wireshark at routers ra, rb, and rc, specifically examining the ra_dump.pcap file, the analysis reveals that, after pinging from host h1 to h3, the process commences with an ARP request from 192.168.1.100 (h1) to determine 'who has 192.168.1.1?'. Subsequently, an ARP reply containing the MAC address is received. The packet transmission ensues from 192.168.1.100 (h1) to 192.168.2.100 (h3), with replies reciprocated from 192.168.2.100 to 192.168.1.100 via router ra.



ra_dump.pcap

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	56:5e:e3:f1:b8:3c		ARP	44	Who has 192.168.1.1? Tell 192.168.1.100
2	0.000017	be:4f:d8:40:88:70		ARP	44	192.168.1.1 is at be:4f:d8:40:88:70
3	0.014392	192.168.1.100	192.168.2.100	ICMP	100	Echo (ping) request id=0x0807, seq=1/256, ttl=64 (no respons…
4	0.014410	2a:7f:ca:e3:30:44		ARP	44	Who has 192.168.4.2? Tell 192.168.4.1
5	0.018029	6e:c6:5a:7b:ca:0c		ARP	44	192.168.4.2 is at 6e:c6:5a:7b:ca:0c
6	0.018037	192.168.1.100	192.168.2.100	ICMP	100	Echo (ping) request id=0x0807, seq=1/256, ttl=63 (reply in 7)
7	0.020160	192.168.2.100	192.168.1.100	ICMP	100	Echo (ping) reply id=0x0807, seq=1/256, ttl=63 (request in…
8	0.020168	192.168.2.100	192.168.1.100	ICMP	100	Echo (ping) reply id=0x0807, seq=1/256, ttl=62
9	1.002722	192.168.1.100	192.168.2.100	ICMP	100	Echo (ping) request id=0x0807, seq=2/512, ttl=64 (no respons…
10	1.002760	192.168.1.100	192.168.2.100	ICMP	100	Echo (ping) request id=0x0807, seq=2/512, ttl=63 (reply in 1…
11	1.003652	192.168.2.100	192.168.1.100	ICMP	100	Echo (ping) reply id=0x0807, seq=2/512, ttl=63 (request in…
12	1.003668	192.168.2.100	192.168.1.100	ICMP	100	Echo (ping) reply id=0x0807, seq=2/512, ttl=62
13	2.006110	192.168.1.100	192.168.2.100	ICMP	100	Echo (ping) request id=0x0807, seq=3/768, ttl=64 (no respons…
14	2.006142	192.168.1.100	192.168.2.100	ICMP	100	Echo (ping) request id=0x0807, seq=3/768, ttl=63 (reply in 1…
15	2.006227	192.168.2.100	192.168.1.100	ICMP	100	Echo (ping) reply id=0x0807, seq=3/768, ttl=63 (request in…
16	2.006233	192.168.2.100	192.168.1.100	ICMP	100	Echo (ping) reply id=0x0807, seq=3/768, ttl=62
17	3.014397	192.168.1.100	192.168.2.100	ICMP	100	Echo (ping) request id=0x0807, seq=4/1024, ttl=64 (no respon…
18	3.014425	192.168.1.100	192.168.2.100	ICMP	100	Echo (ping) request id=0x0807, seq=4/1024, ttl=63 (reply in …
19	3.014497	192.168.2.100	192.168.1.100	ICMP	100	Echo (ping) reply id=0x0807, seq=4/1024, ttl=63 (request i…
20	3.014503	192.168.2.100	192.168.1.100	ICMP	100	Echo (ping) reply id=0x0807, seq=4/1024, ttl=62
21	5.029287	be:4f:d8:40:88:70		ARP	44	Who has 192.168.1.100? Tell 192.168.1.1
22	5.032270	6e:c6:5a:7b:ca:0c		ARP	44	Who has 192.168.4.1? Tell 192.168.4.2
23	5.032292	2a:7f:ca:e3:30:44		ARP	44	192.168.4.1 is at 2a:7f:ca:e3:30:44
24	5.055709	56:5e:e3:f1:b8:3c		ARP	44	192.168.1.100 is at 56:5e:e3:f1:b8:3c
25	5.459083	192.168.1.100	192.168.3.101	ICMP	100	Echo (ping) request id=0x080d, seq=1/256, ttl=64 (no respons…
26	5.459099	2a:7f:ca:e3:30:44		ARP	44	Who has 192.168.4.3? Tell 192.168.4.1
27	5.460155	7e:74:1d:8c:e8:c5		ARP	44	192.168.4.3 is at 7e:74:1d:8c:e8:c5
28	5.460160	192.168.1.100	192.168.3.101	ICMP	100	Echo (ping) request id=0x080d, seq=1/256, ttl=63 (reply in 2…
29	5.462455	192.168.3.101	192.168.1.100	ICMP	100	Echo (ping) reply id=0x080d, seq=1/256, ttl=63 (request in…
30	5.462464	192.168.3.101	192.168.1.100	ICMP	100	Echo (ping) reply id=0x080d, seq=1/256, ttl=62
31	6.461892	192.168.1.100	192.168.3.101	ICMP	100	Echo (ping) request id=0x080d, seq=2/512, ttl=64 (no respons…
32	6.461923	192.168.1.100	192.168.3.101	ICMP	100	Echo (ping) request id=0x080d, seq=2/512, ttl=63 (reply in 3…
33	6.462741	192.168.3.101	192.168.1.100	ICMP	100	Echo (ping) reply id=0x080d, seq=2/512, ttl=63 (request in…
34	6.462754	192.168.3.101	192.168.1.100	ICMP	100	Echo (ping) reply id=0x080d, seq=2/512, ttl=62
35	7.464130	192.168.1.100	192.168.3.101	ICMP	100	Echo (ping) request id=0x080d, seq=3/768, ttl=64 (no respons…

Frame 28: 100 bytes on wire (800 bits), 100 bytes captured (800 bits)

ra_dump.pcap   Packets: 44 · Displayed: 44 (100.0%)   Profile: Default

c) Comparing latency differences between default and changed route.

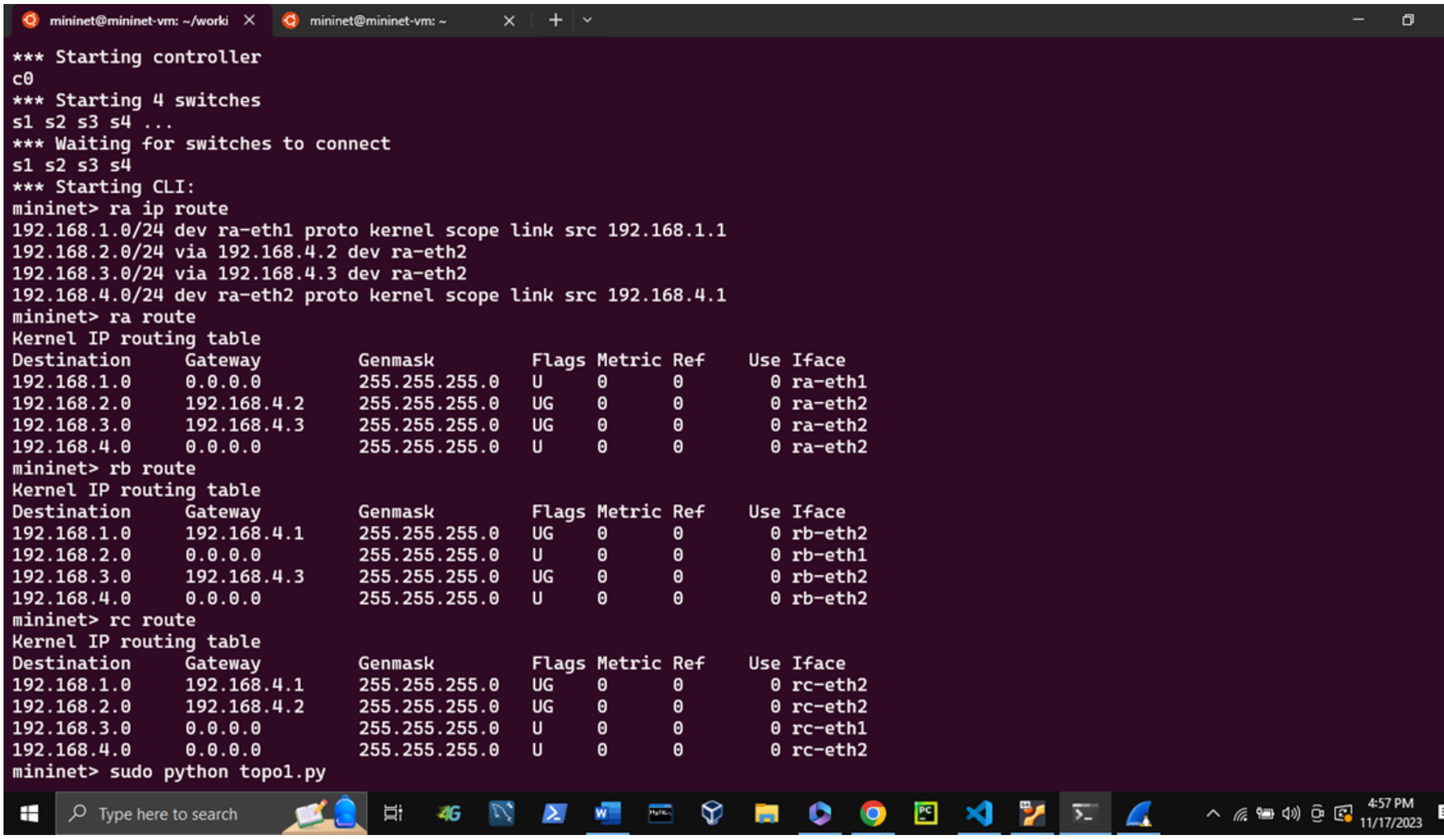
```
*** Starting CLI:
mininet> ra ip route
192.168.1.0/24 dev ra-eth1 proto kernel scope link src 192.168.1.1
192.168.2.0/24 via 192.168.4.2 dev ra-eth2
192.168.3.0/24 via 192.168.4.3 dev ra-eth2
192.168.4.0/24 dev ra-eth2 proto kernel scope link src 192.168.4.1
mininet> ra ip route del 192.168.3.0/24
mininet> ra ip route add 192.168.3.0/24 via 192.168.4.2 dev ra-eth2
mininet> ra ip route
192.168.1.0/24 dev ra-eth1 proto kernel scope link src 192.168.1.1
192.168.2.0/24 via 192.168.4.2 dev ra-eth2
192.168.3.0/24 via 192.168.4.2 dev ra-eth2
192.168.4.0/24 dev ra-eth2 proto kernel scope link src 192.168.4.1
mininet> h1 traceroute h6
traceroute to 192.168.3.101 (192.168.3.101), 30 hops max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  7.462 ms  7.987 ms  8.085 ms
 2  192.168.4.2 (192.168.4.2)  16.138 ms  17.506 ms  19.843 ms
 3  192.168.4.3 (192.168.4.3)  29.138 ms  31.784 ms  33.183 ms
 4  192.168.3.101 (192.168.3.101)  38.076 ms  40.750 ms  41.021 ms
mininet> h1 ping -c 4 h6
PING 192.168.3.101 (192.168.3.101) 56(84) bytes of data.
64 bytes from 192.168.3.101: icmp_seq=1 ttl=62 time=4.68 ms
64 bytes from 192.168.3.101: icmp_seq=2 ttl=62 time=23.3 ms
64 bytes from 192.168.3.101: icmp_seq=3 ttl=62 time=0.642 ms
64 bytes from 192.168.3.101: icmp_seq=4 ttl=62 time=0.117 ms

--- 192.168.3.101 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3021ms
rtt min/avg/max/mdev = 0.117/7.176/23.265/9.455 ms
mininet> iperf h1 h6
*** Iperf: testing TCP bandwidth between h1 and h6
*** Results: ['2.56 Gbits/sec', '2.58 Gbits/sec']
```

```
--- 192.168.3.101 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3021ms
rtt min/avg/max/mdev = 0.117/7.176/23.265/9.455 ms
mininet> iperf h1 h6
*** Iperf: testing TCP bandwidth between h1 and h6
*** Results: ['2.56 Gbits/sec', '2.58 Gbits/sec']
mininet> ra ip route del 192.168.3.0/24
mininet> ra ip route add 192.168.3.0/24 via 192.168.4.2 dev ra-eth2
mininet> h1 ping -c 4 h6
PING 192.168.3.101 (192.168.3.101) 56(84) bytes of data.
64 bytes from 192.168.3.101: icmp_seq=1 ttl=62 time=5.46 ms
64 bytes from 192.168.3.101: icmp_seq=2 ttl=62 time=18.1 ms
64 bytes from 192.168.3.101: icmp_seq=3 ttl=62 time=0.416 ms
64 bytes from 192.168.3.101: icmp_seq=4 ttl=62 time=0.122 ms

--- 192.168.3.101 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3016ms
rtt min/avg/max/mdev = 0.122/6.032/18.130/7.299 ms
mininet>
```

d) Dumping routing tables:



```
*** Starting controller
c0
*** Starting 4 switches
s1 s2 s3 s4 ...
*** Waiting for switches to connect
s1 s2 s3 s4
*** Starting CLI:
mininet> ra ip route
192.168.1.0/24 dev ra-eth1 proto kernel scope link src 192.168.1.1
192.168.2.0/24 via 192.168.4.2 dev ra-eth2
192.168.3.0/24 via 192.168.4.3 dev ra-eth2
192.168.4.0/24 dev ra-eth2 proto kernel scope link src 192.168.4.1
mininet> ra route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 ra-eth1
192.168.2.0     192.168.4.2     255.255.255.0   UG    0      0        0 ra-eth2
192.168.3.0     192.168.4.3     255.255.255.0   UG    0      0        0 ra-eth2
192.168.4.0     0.0.0.0         255.255.255.0   U     0      0        0 ra-eth2
mininet> rb route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.1.0     192.168.4.1     255.255.255.0   UG    0      0        0 rb-eth2
192.168.2.0     0.0.0.0         255.255.255.0   U     0      0        0 rb-eth1
192.168.3.0     192.168.4.3     255.255.255.0   UG    0      0        0 rb-eth2
192.168.4.0     0.0.0.0         255.255.255.0   U     0      0        0 rb-eth2
mininet> rc route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.1.0     192.168.4.1     255.255.255.0   UG    0      0        0 rc-eth2
192.168.2.0     192.168.4.2     255.255.255.0   UG    0      0        0 rc-eth2
192.168.3.0     0.0.0.0         255.255.255.0   U     0      0        0 rc-eth1
192.168.4.0     0.0.0.0         255.255.255.0   U     0      0        0 rc-eth2
mininet> sudo python topo1.py
```

PART II

We established a network configuration with four hosts, namely h1, h2, h3, and h4. Additionally, there are two switches, labeled s1 and s2. The connections are set up as follows: h1 and h2 are linked to switch s1, while h3 and h4 are connected to switch s2. Furthermore, switches s1 and s2 are interconnected.

IP of the hosts:
h1: 10.0.0.1/24
h2: 10.0.0.2/24
h3: 10.0.0.3/24
h4: 10.0.0.4/24

(a)

```
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2
*** Adding links:
(2.00Mbit 2 delay) (2.00Mbit 2 delay) (h1, s1) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (h2, s1) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (
h3, s2) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (h4, s2) (8.00Mbit 2 delay 0.00000% loss) (8.00Mbit 2 delay 0.00000% loss) (s1, s2)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...(2.00Mbit 2 delay) (2.00Mbit 2 delay) (8.00Mbit 2 delay 0.00000% loss) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (8.00Mbit 2 del
ay 0.00000% loss)
*** Waiting for switches to connect
s1 s2
------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.1 port 58910 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.4 sec  10.2 MBytes  8.27 Mbits/sec

*** Starting CLI:
mininet> 
```

(b) reno

```
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2
*** Adding links:
(2.00Mbit 2 delay) (2.00Mbit 2 delay) (h1, s1) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (h2, s1) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (
h3, s2) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (h4, s2) (8.00Mbit 2 delay 0.00000% loss) (8.00Mbit 2 delay 0.00000% loss) (s1, s2)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...(2.00Mbit 2 delay) (2.00Mbit 2 delay) (8.00Mbit 2 delay 0.00000% loss) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (8.00Mbit 2 del
ay 0.00000% loss)
*** Waiting for switches to connect
s1 s2
------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.1 port 58910 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.4 sec  10.2 MBytes  8.27 Mbits/sec

*** Starting CLI:
mininet> 
```

Vegas

```
mininet@mininet-vm:~$ sudo python topo2.py --config b --cc vegas
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2
*** Adding links:
(2.00Mbit 2 delay) (2.00Mbit 2 delay) (h1, s1) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (h2, s1) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (
h3, s2) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (h4, s2) (8.00Mbit 2 delay 0.00000% loss) (8.00Mbit 2 delay 0.00000% loss) (s1, s2)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...(2.00Mbit 2 delay) (2.00Mbit 2 delay) (8.00Mbit 2 delay 0.00000% loss) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (8.00Mbit 2 del
ay 0.00000% loss)
*** Waiting for switches to connect
s1 s2
------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.1 port 58918 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.2 sec  2.62 MBytes  2.16 Mbits/sec

*** Starting CLI:
mininet>
```

Cubic

```
mininet@mininet-vm:~$ sudo python topo2.py --config b --cc cubic
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2
*** Adding links:
(2.00Mbit 2 delay) (2.00Mbit 2 delay) (h1, s1) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (h2, s1) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (
h3, s2) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (h4, s2) (8.00Mbit 2 delay 0.00000% loss) (8.00Mbit 2 delay 0.00000% loss) (s1, s2)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...(2.00Mbit 2 delay) (2.00Mbit 2 delay) (8.00Mbit 2 delay 0.00000% loss) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (8.00Mbit 2 del
ay 0.00000% loss)
*** Waiting for switches to connect
s1 s2
------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.1 port 58926 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-11.2 sec  4.50 MBytes  3.37 Mbits/sec

*** Starting CLI:
mininet>
```

Bbr

```
mininet@mininet-vm:~$ sudo python topo2.py --config b --cc bbr
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2
*** Adding links:
(2.00Mbit 2 delay) (2.00Mbit 2 delay) (h1, s1) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (h2, s1) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (
h3, s2) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (h4, s2) (8.00Mbit 2 delay 0.00000% loss) (8.00Mbit 2 delay 0.00000% loss) (s1, s2)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...(2.00Mbit 2 delay) (2.00Mbit 2 delay) (8.00Mbit 2 delay 0.00000% loss) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (8.00Mbit 2 del
ay 0.00000% loss)
*** Waiting for switches to connect
s1 s2
------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5001
TCP window size:  178 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.1 port 58934 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.3 sec  2.50 MBytes  2.04 Mbits/sec

*** Starting CLI:
```

Reasoning of the throughput:

The throughput variations observed among different Congestion Control Algorithms (CCAs) arise from their distinct approaches to managing network congestion. Each CCA adjusts transmission rates based on network conditions, influencing overall throughput.

Reno: Serving as the default TCP CCA, Reno adopts a loss-based strategy, decreasing transmission rates upon detecting packet loss. Although straightforward, its conservative rate reduction might constrain throughput in congested networks. The iperf test methodology, however, may have unintentionally favored Reno, resulting in higher observed throughput.

Vegas: Vegas takes into account both packet loss and delay information to estimate congestion, allowing for more precise rate adjustments compared to Reno. Nevertheless, the iperf test methodology may not have favored Reno, leading to a comparatively lower throughput for Vegas.

Cubic: Cubic combines loss, delay, and packet arrival times for congestion estimation, enabling faster rate adjustments than Vegas. This could elucidate its higher throughput compared to Vegas.

BBR: Google's BBR CCA relies on a delay-based approach, using a network model to estimate the optimal sending rate. While BBR can achieve high throughput in networks with high latency or packet loss, its implementation complexity may impact its use. Additionally, the iperf testing methodology may have negatively influenced the observed throughput for BBR.

(c):

To run clients h1, h2, and h3 simultaneously on server h4, we initiated the iperf server on h4. Next, we established three threads corresponding to h1, h2, and h3, serving as clients to h4. These threads were concurrently executed during the active network period, with all functionalities defined within the run() function.

reno

```
*** Waiting for switches to connect
s1 s2
Running hosts simultaneously
Iperf throughput from h3 to H4 with {'reno'}:
 ------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.3 port 34800 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.9 sec  2.12 MBytes  1.63 Mbits/sec

Iperf throughput from h2 to H4 with {'reno'}:
 ------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.2 port 56242 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-11.8 sec  1.88 MBytes  1.33 Mbits/sec

Iperf throughput from h1 to H4 with {'reno'}:
 ------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.1 port 58942 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-12.3 sec  2.12 MBytes  1.45 Mbits/sec

Ping latency from h3 to H4 with {'reno'}:
 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=3756 ms
```

Vegas

```
*** Starting 2 switches
s1 s2 ...(2.00Mbit 2 delay) (2.00Mbit 2 delay) (8.00Mbit 2 delay 0.00000% loss) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (8.00Mbit 2 del
ay 0.00000% loss)
*** Waiting for switches to connect
s1 s2
Running hosts simultaneously
Iperf throughput from h2 to H4 with {'vegas'}:
 ------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.2 port 56254 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.1 sec  1.62 MBytes  1.35 Mbits/sec

Iperf throughput from h3 to H4 with {'vegas'}:
 ------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.3 port 34810 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-11.0 sec  1.88 MBytes  1.43 Mbits/sec

Iperf throughput from h1 to H4 with {'vegas'}:
 ------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.1 port 58956 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-11.9 sec  2.12 MBytes  1.50 Mbits/sec
```

Cubic

```
s1 s2
Running hosts simultaneously
Iperf throughput from h2 to H4 with {'cubic'}:
 ------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.2 port 56262 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-11.6 sec  2.12 MBytes  1.54 Mbits/sec

Iperf throughput from h3 to H4 with {'cubic'}:
 ------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.3 port 34826 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-11.8 sec  2.12 MBytes  1.52 Mbits/sec

Iperf throughput from h1 to H4 with {'cubic'}:
 ------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.1 port 58968 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-12.6 sec  2.12 MBytes  1.42 Mbits/sec
```

Bbr

```
*** Waiting for switches to connect
s1 s2
Running hosts simultaneously
Iperf throughput from h3 to H4 with {'bbr'}:
 ------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.3 port 34834 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-11.5 sec  2.12 MBytes  1.55 Mbits/sec

Iperf throughput from h2 to H4 with {'bbr'}:
 ------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.2 port 56276 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-12.3 sec  2.12 MBytes  1.45 Mbits/sec

Iperf throughput from h1 to H4 with {'bbr'}:
 ------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.1 port 58982 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-12.5 sec  2.12 MBytes  1.43 Mbits/sec
```

Summary of the schemes:

Scheme	Source	Throughput	File Transferred
reno	h1->h4	1.63 Mb/s	2.12 MB
reno	h2->h4	1.33 Mb/s	1.88 MB
reno	h3->h4	1.45 Mb/s	2.12 MB
vegas	h1->h4	1.35 Mb/s	1.62 MB
vegas	h2->h4	1.43 Mb/s	1.88 MB
vegas	h3->h4	1.5 Mb/s	2.12 MB
cubic	h1->h4	1.54 Mb/s	2.12 MB
cubic	h2->h4	1.52 Mb/s	2.12 MB
cubic	h3->h4	1.42 Mb/s	2.12 MB
bbr	h1->h4	1.55 Mb/s	2.12 MB
bbr	h2->h4	1.45 Mb/s	2.12 MB
bbr	h3->h4	1.43 Mb/s	2.12 MB

(d)

Reno with 1% loss

```
mininet@mininet-vm:~$ sudo python topo2.py --config b --loss 1 --cc reno
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2
*** Adding links:
(2.00Mbit 2 delay) (2.00Mbit 2 delay) (h1, s1) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (h2, s1) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (
h3, s2) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (h4, s2) (8.00Mbit 2 delay 1.00000% loss) (8.00Mbit 2 delay 1.00000% loss) (s1, s2)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...(2.00Mbit 2 delay) (2.00Mbit 2 delay) (8.00Mbit 2 delay 1.00000% loss) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (8.00Mbit 2 del
ay 1.00000% loss)
*** Waiting for switches to connect
s1 s2
------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.1 port 58998 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.5 sec  2.62 MBytes  2.09 Mbits/sec

*** Starting CLI:
mininet> 
```

Reno with 3% loss

```
mininet@mininet-vm:~$ sudo python topo2.py --config b --loss 3 --cc reno
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2
*** Adding links:
(2.00Mbit 2 delay) (2.00Mbit 2 delay) (h1, s1) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (h2, s1) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (
h3, s2) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (h4, s2) (8.00Mbit 2 delay 3.00000% loss) (8.00Mbit 2 delay 3.00000% loss) (s1, s2)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...(2.00Mbit 2 delay) (2.00Mbit 2 delay) (8.00Mbit 2 delay 3.00000% loss) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (8.00Mbit 2 del
ay 3.00000% loss)
*** Waiting for switches to connect
s1 s2
------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.1 port 59006 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.4 sec  2.88 MBytes  2.32 Mbits/sec

*** Starting CLI:
mininet>
```

Vegas with 1% loss

```
mininet@mininet-vm:~$ sudo python topo2.py --config b --loss 1 --cc vegas
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2
*** Adding links:
(2.00Mbit 2 delay) (2.00Mbit 2 delay) (h1, s1) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (h2, s1) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (
h3, s2) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (h4, s2) (8.00Mbit 2 delay 1.00000% loss) (8.00Mbit 2 delay 1.00000% loss) (s1, s2)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...(2.00Mbit 2 delay) (2.00Mbit 2 delay) (8.00Mbit 2 delay 1.00000% loss) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (8.00Mbit 2 del
ay 1.00000% loss)
*** Waiting for switches to connect
s1 s2
------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.1 port 59758 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec  2.50 MBytes  2.09 Mbits/sec

*** Starting CLI:
mininet>
```

Vegas with 3% loss

```
*** Done
mininet@mininet-vm:~$ sudo python topo2.py --config b --loss 3 --cc vegas
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2
*** Adding links:
(2.00Mbit 2 delay) (2.00Mbit 2 delay) (h1, s1) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (h2, s1) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (
h3, s2) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (h4, s2) (8.00Mbit 2 delay 3.00000% loss) (8.00Mbit 2 delay 3.00000% loss) (s1, s2)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...(2.00Mbit 2 delay) (2.00Mbit 2 delay) (8.00Mbit 2 delay 3.00000% loss) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (8.00Mbit 2 del
ay 3.00000% loss)
*** Waiting for switches to connect
s1 s2
------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.1 port 59766 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.2 sec  2.25 MBytes  1.86 Mbits/sec

*** Starting CLI:
mininet>
```

Cubic with 1% loss

```
mininet@mininet-vm:~$ sudo python topo2.py --config b --loss 1 --cc cubic
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2
*** Adding links:
(2.00Mbit 2 delay) (2.00Mbit 2 delay) (h1, s1) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (h2, s1) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (
h3, s2) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (h4, s2) (8.00Mbit 2 delay 1.00000% loss) (8.00Mbit 2 delay 1.00000% loss) (s1, s2)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...(2.00Mbit 2 delay) (2.00Mbit 2 delay) (8.00Mbit 2 delay 1.00000% loss) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (8.00Mbit 2 del
ay 1.00000% loss)
*** Waiting for switches to connect
s1 s2
------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.1 port 59774 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.3 sec  2.62 MBytes  2.14 Mbits/sec
```

Cubic with 3% loss

```
*** Done
mininet@mininet-vm:~$ sudo python topo2.py --config b --loss 3 --cc cubic
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2
*** Adding links:
(2.00Mbit 2 delay) (2.00Mbit 2 delay) (h1, s1) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (h2, s1) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (
h3, s2) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (h4, s2) (8.00Mbit 2 delay 3.00000% loss) (8.00Mbit 2 delay 3.00000% loss) (s1, s2)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...(2.00Mbit 2 delay) (2.00Mbit 2 delay) (8.00Mbit 2 delay 3.00000% loss) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (8.00Mbit 2 del
ay 3.00000% loss)
*** Waiting for switches to connect
s1 s2
------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.1 port 59782 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.3 sec  2.50 MBytes  2.04 Mbits/sec

*** Starting CLI:
mininet>
```

Bbr with 1% loss

```
mininet@mininet-vm:~$ sudo python topo2.py --config b --loss 1 --cc bbr
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2
*** Adding links:
(2.00Mbit 2 delay) (2.00Mbit 2 delay) (h1, s1) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (h2, s1) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (
h3, s2) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (h4, s2) (8.00Mbit 2 delay 1.00000% loss) (8.00Mbit 2 delay 1.00000% loss) (s1, s2)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...(2.00Mbit 2 delay) (2.00Mbit 2 delay) (8.00Mbit 2 delay 1.00000% loss) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (8.00Mbit 2 del
ay 1.00000% loss)
*** Waiting for switches to connect
s1 s2
------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5001
TCP window size:  153 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.1 port 59790 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.3 sec  2.50 MBytes  2.03 Mbits/sec

*** Starting CLI:
mininet>
```

Bbr with 3% loss

```
mininet@mininet-vm:~$ sudo python topo2.py --config b --loss 3 --cc bbr
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2
*** Adding links:
(2.00Mbit 2 delay) (2.00Mbit 2 delay) (h1, s1) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (h2, s1) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (
h3, s2) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (h4, s2) (8.00Mbit 2 delay 3.00000% loss) (8.00Mbit 2 delay 3.00000% loss) (s1, s2)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...(2.00Mbit 2 delay) (2.00Mbit 2 delay) (8.00Mbit 2 delay 3.00000% loss) (2.00Mbit 2 delay) (2.00Mbit 2 delay) (8.00Mbit 2 del
ay 3.00000% loss)
*** Waiting for switches to connect
s1 s2
------------------------------------------------------------
Client connecting to 10.0.0.4, TCP port 5001
TCP window size:  153 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.1 port 59798 connected with 10.0.0.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec  2.38 MBytes  1.99 Mbits/sec

*** Starting CLI:
mininet>
```

Summary of the results: (h1->h4)

Scheme	Throughput	Transferred Data
RENO (with no loss)	7.51 Mbits/sec	9.38 MB
RENO (with 1% loss)	2.09 Mbits/sec	2.62 MB
RENO (with 3% loss)	2.32 Mbits/sec	2.88 MB
VEGAS (with no loss)	2.13 Mbits/sec	2.62 MB
VEGAS (with 1% loss)	2.16 Mbits/sec	2.62 MB
VEGAS (with 3% loss)	1.86 Mbits/sec	2.25 MB
CUBIC (with no loss)	3.42 Mbits/sec	4.62 MB
CUBIC (with 1% loss)	2.14 Mbits/sec	2.62 MB
CUBIC (with 3% loss)	2.04 Mbits/sec	2.50 MB
BBR (with no loss)	2.03 Mbits/sec	2.50 MB
BBR (with 1% loss)	2.03 Mbits/sec	2.50 MB
BBR (with 3% loss)	1.99 Mbits/sec	2.38 MB

In the case of Reno, notable reductions in throughput are evident with 1% and 3% losses on the link. Vegas, on the other hand, experiences a slight decrease in throughput only with a 3% loss on the link, while at 1% loss, throughput remains comparable. Cubic demonstrates reduced throughput with both 1% and 3% losses on the link. However, for BBR, the impact of 1% and 3% losses on the link is not substantial.