CN Assignment-4:

Question 1:

a) Maximum Expected Throughput: 7 Mbps, because the throughput is constrained by the bottleneck link (N1 -> N2) with the lowest capacity of 7 Mbps.

b) BDP(Bandwidth Delay Product) (in bits) = 7 × 106(bits/second) × 0.22(seconds) = 1.54×106(bits)

In terms of Packets:

- **For Payload + Header :**

  1. Packet Size = 1460(bytes) + 40(bytes) = 1500(bytes)
  2. Packet Size(in bits) = 1500(bytes) × 8(bits/byte) = 12,000(bits)
  3. BDP(in packets) = Packet Size(in bits) x BDP(in bits) = (1.54 × 10^6) / 12000 ≈ 128.33packets

- **Payload Only:**
  1. Payload Size(in bits) = 1460bytes×8bits/byte=11,680bits.
  2. BDP(in packets) = Payload Size(in bits) x BDP(in bits) = (1.54 × 10^6) / 11680 ≈ 131.88packets.

c) Use these command to convert the tshark output to easyly readable txt file.

- "tshark -r tcp-example-0-0.pcap -T fields -e frame.time_epoch -e ip.src -e ip.dst -e tcp.port -e tcp.seq -e tcp.len -e tcp.ack > tcp-example-0-0_output.txt"
- "tshark -r tcp-example-1-0.pcap -T fields -e frame.time_epoch -e ip.src -e ip.dst -e tcp.port -e tcp.seq -e tcp.len -e tcp.ack > tcp-example-0-0_output.txt"
- "tshark -r tcp-example-2-0.pcap -T fields -e frame.time_epoch -e ip.src -e ip.dst -e tcp.port -e tcp.seq -e tcp.len -e tcp.ack > tcp-example-0-0_output.txt"

Alternatively we can view the output directly on terminal via:

- “tshark -r tcp-example-0-0.pcap -T fields -e frame.time_epoch -e ip.src -e ip.dst -e tcp.port -e tcp.seq -e tcp.len -e tcp.ack”

Alternatively:

I have created a shell script u can run it directly

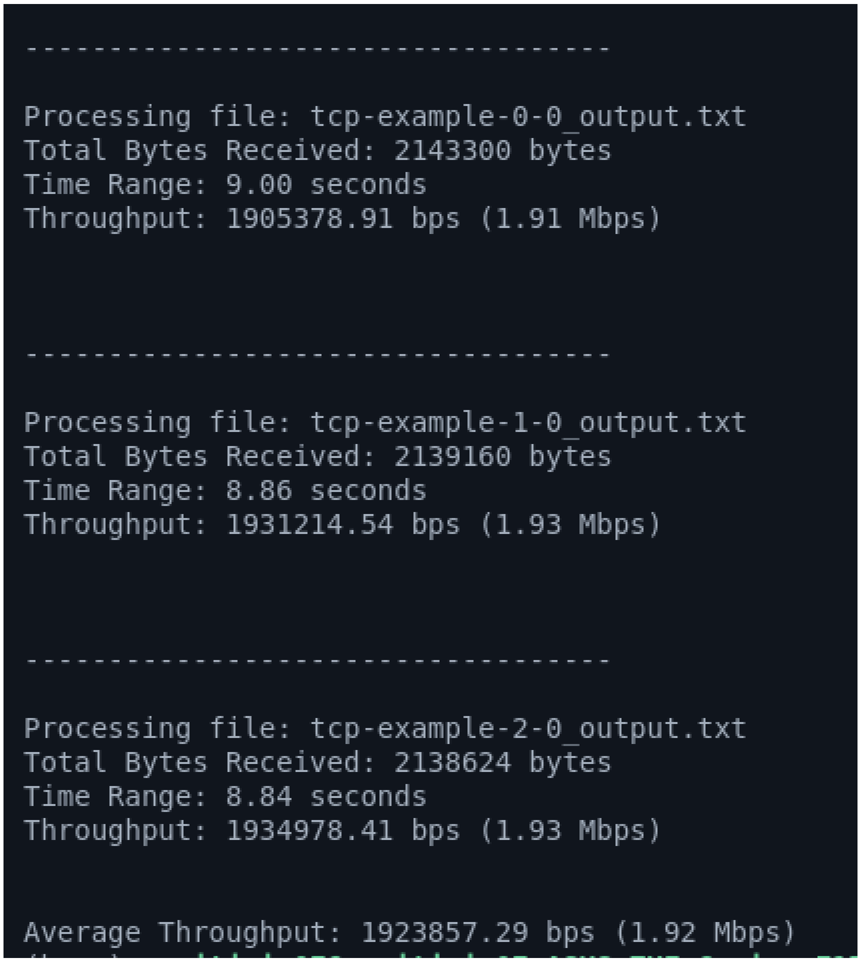
- “chmod +x generate-output-files.sh”
- “./generate-output-files.sh”

Run ThroughputCalculator.java to see the actual Throughput:

Cmd:

- “javac ThroughputCalculator.java”
- “java ThroughputCalculator”

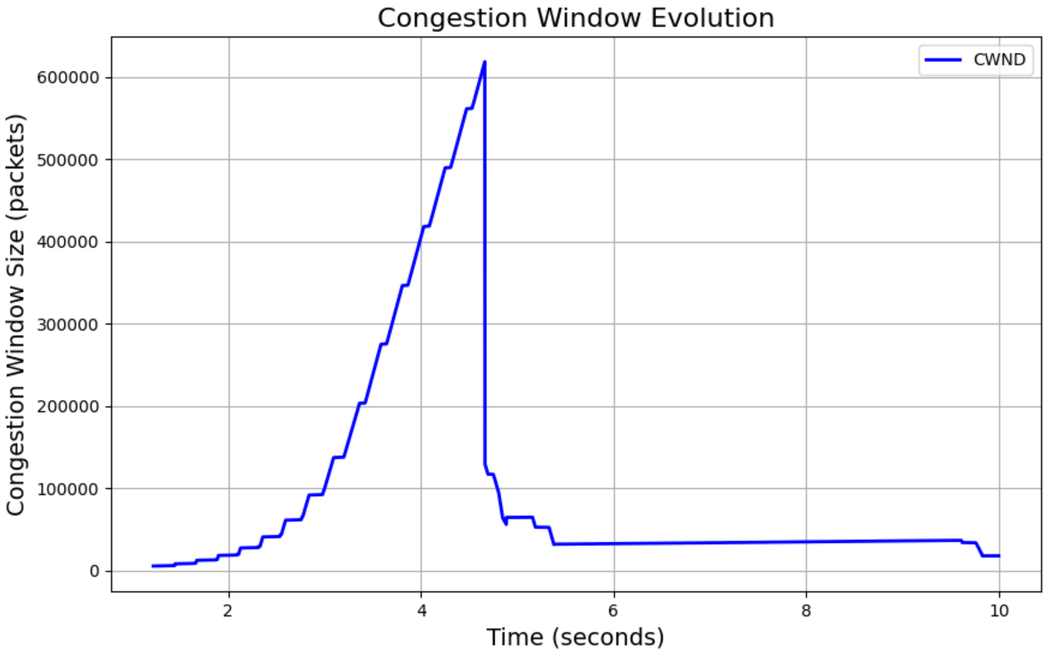
Output:



```
-----------------------------------

Processing file: tcp-example-0-0_output.txt
Total Bytes Received: 2143300 bytes
Time Range: 9.00 seconds
Throughput: 1905378.91 bps (1.91 Mbps)



-----------------------------------

Processing file: tcp-example-1-0_output.txt
Total Bytes Received: 2139160 bytes
Time Range: 8.86 seconds
Throughput: 1931214.54 bps (1.93 Mbps)



-----------------------------------

Processing file: tcp-example-2-0_output.txt
Total Bytes Received: 2138624 bytes
Time Range: 8.84 seconds
Throughput: 1934978.41 bps (1.93 Mbps)


Average Throughput: 1923857.29 bps (1.92 Mbps)
```

d) The achieved throughput (1.92Mbps) is much lower than the maximum expected throughput (7Mbps).
Because:

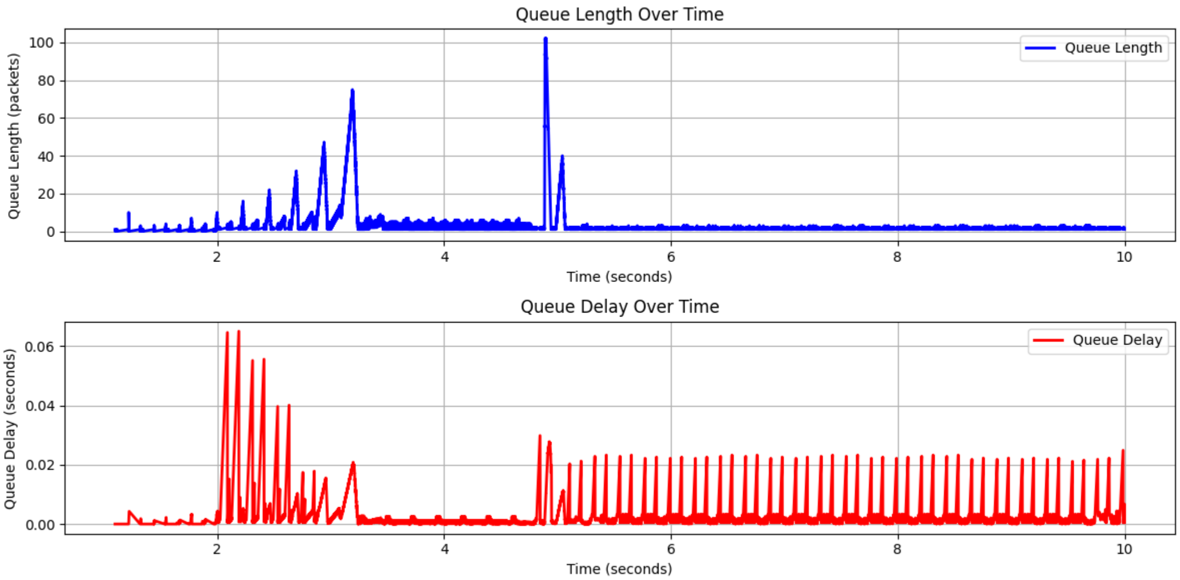
- High RTT leads to limited TCP performance.
- window size limitations due to insufficient BDP matching.
- TCP's congestion control dynamics not fully utilizing the bandwidth.
- Packet loss or retransmissions caused by congestion or buffer overflows.
- Application-level constraints.



e)

We can Obtain this graph by running this command:

- “python3 plot-congestion-control.py”



f)

We can obtain this graph by running this command:

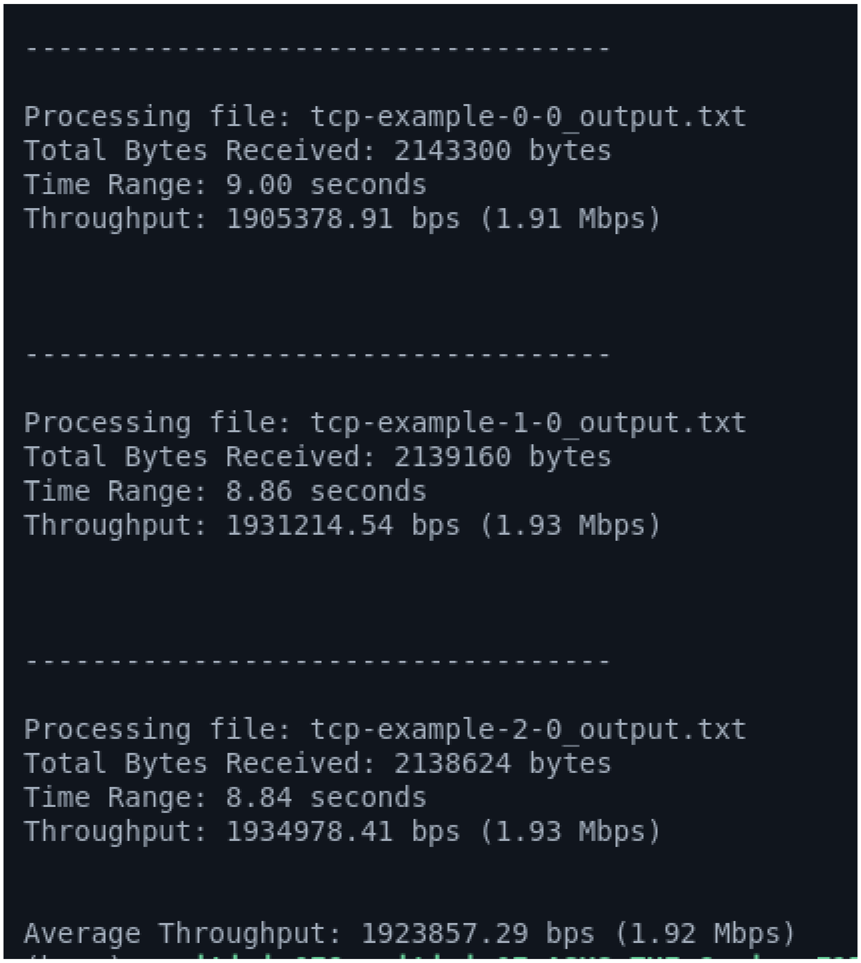
- python3 plotting-queue-delay.py

g) YES,

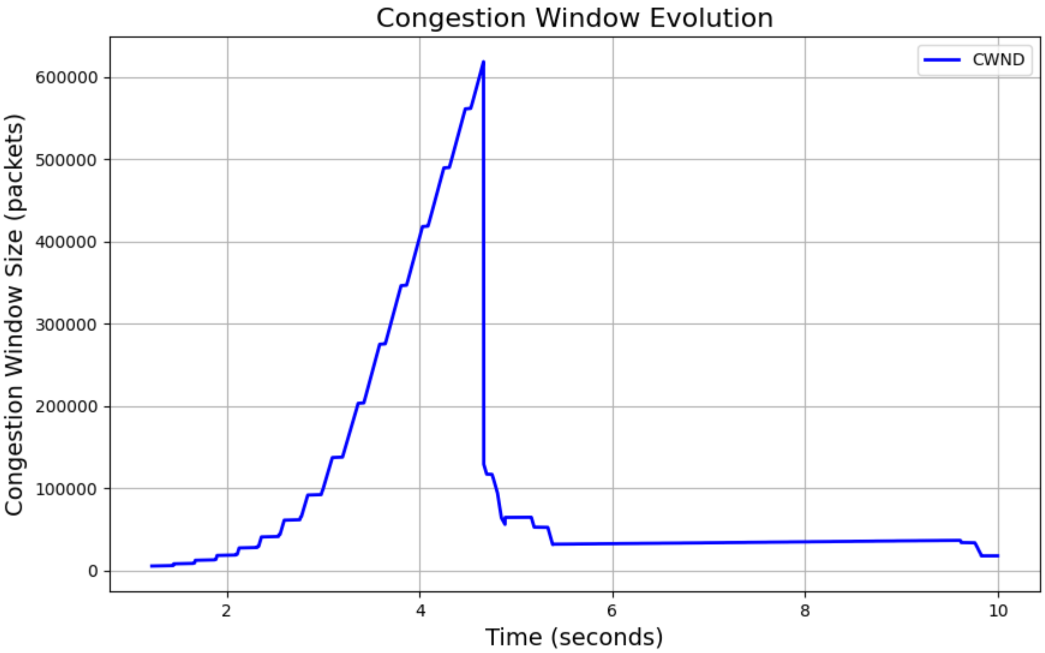
- In plot 1(e), the CWND increases gradually until it reaches a peak (around 4.5 seconds), followed by a sharp decline, which likely suggests packet loss or congestion.
- In plot 1(f), the queueing delay also rises notably as CWND grows, peaking around the same time CWND reaches its maximum. This indicates network congestion, leading to delays caused by queuing.

Question 2:

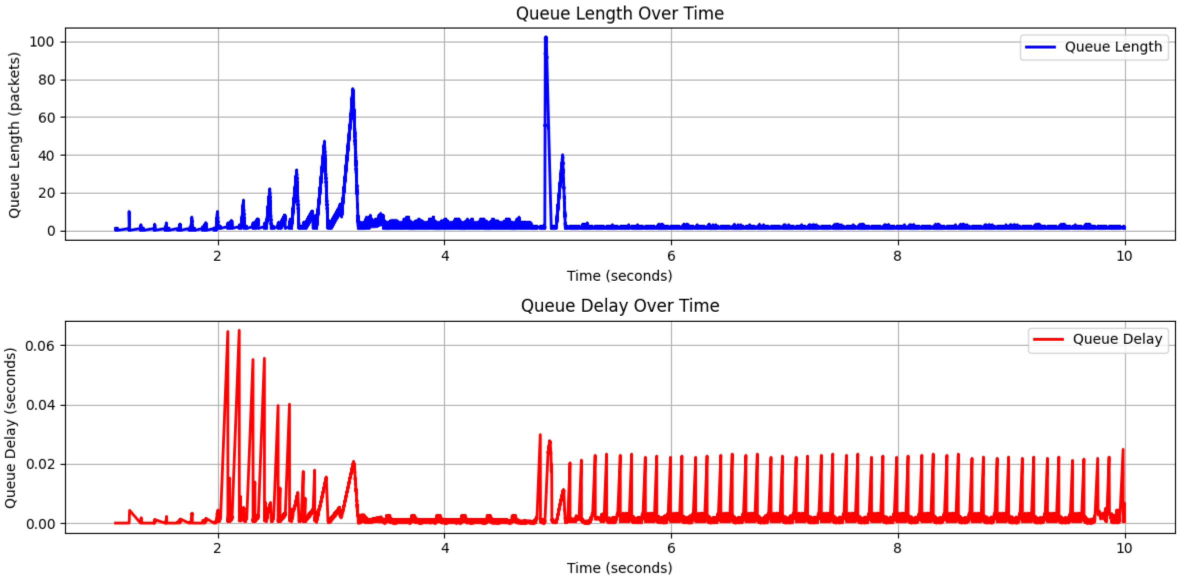
a)



```
-----------------------------------

Processing file: tcp-example-0-0_output.txt
Total Bytes Received: 2143300 bytes
Time Range: 9.00 seconds
Throughput: 1905378.91 bps (1.91 Mbps)



-----------------------------------

Processing file: tcp-example-1-0_output.txt
Total Bytes Received: 2139160 bytes
Time Range: 8.86 seconds
Throughput: 1931214.54 bps (1.93 Mbps)



-----------------------------------

Processing file: tcp-example-2-0_output.txt
Total Bytes Received: 2138624 bytes
Time Range: 8.84 seconds
Throughput: 1934978.41 bps (1.93 Mbps)


Average Throughput: 1923857.29 bps (1.92 Mbps)
```



b)



c)

d) The CWND (Congestion Window) plots for Q.1 and Q.2 look the same because increasing the queue size to 1000 doesn't directly impact how TCP adjusts its congestion window. TCP’s behavior mainly depends on factors like packet loss and acknowledgment timing, which are influenced by the network's capacity and delay rather than the queue size.

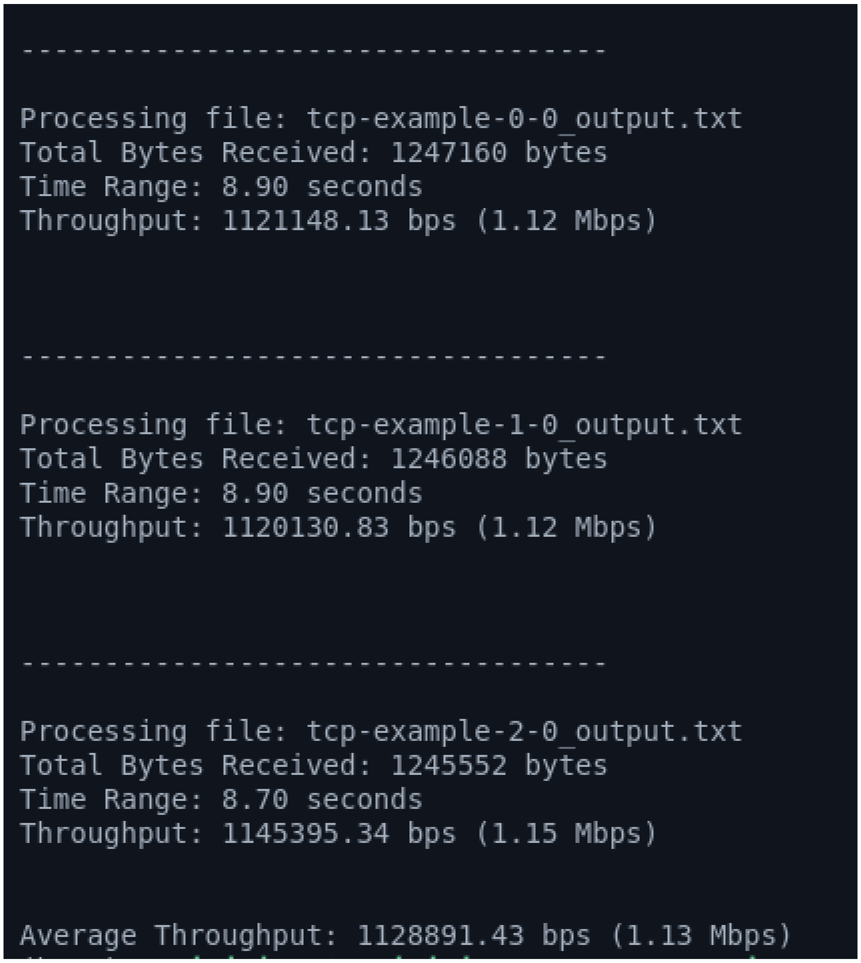
Insights Gained:

- Queue Size Doesn’t Change CWND: The congestion window is controlled by TCP algorithms (like AIMD), which react to packet loss or round-trip time. Since those factors didn’t change, the CWND plots stayed identical.

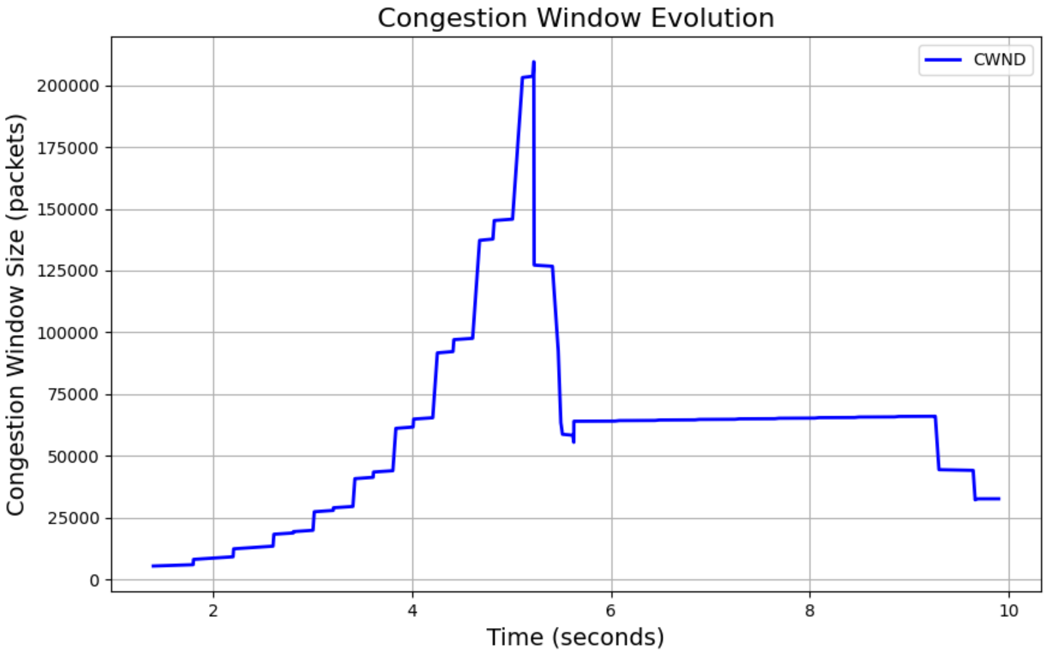
- Queue Size Affects Delay, Not CWND: A bigger queue means more packets can sit in the buffer, which increases the queueing delay. This shows that while the throughput stays consistent, the delay can change depending on the queue size.

Question 3:

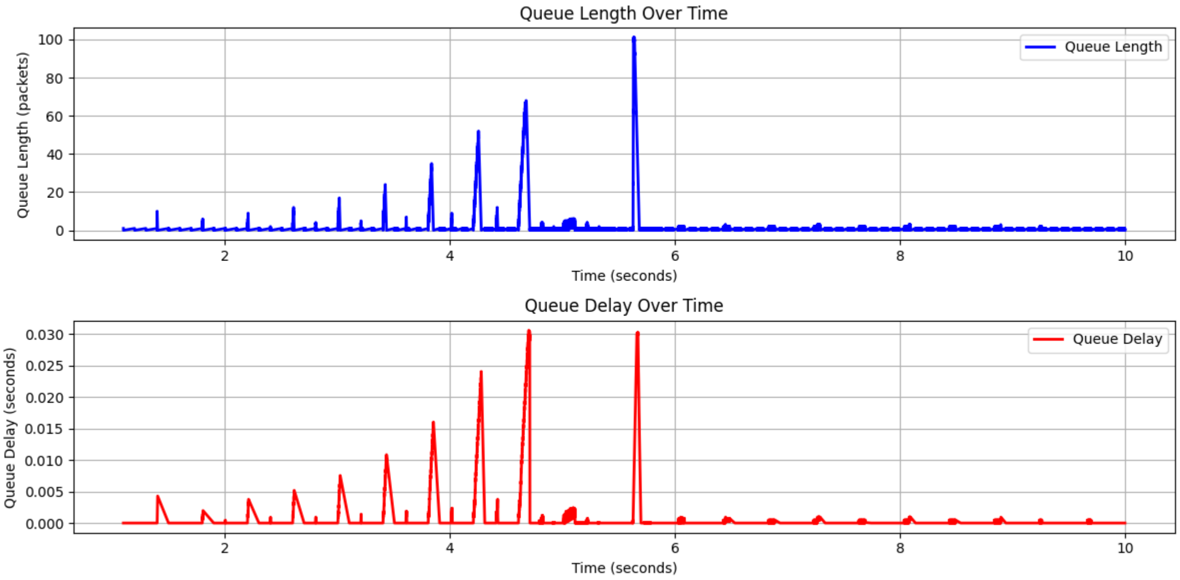
a)



```
-----------------------------------

Processing file: tcp-example-0-0_output.txt
Total Bytes Received: 1247160 bytes
Time Range: 8.90 seconds
Throughput: 1121148.13 bps (1.12 Mbps)



-----------------------------------

Processing file: tcp-example-1-0_output.txt
Total Bytes Received: 1246088 bytes
Time Range: 8.90 seconds
Throughput: 1120130.83 bps (1.12 Mbps)



-----------------------------------

Processing file: tcp-example-2-0_output.txt
Total Bytes Received: 1245552 bytes
Time Range: 8.70 seconds
Throughput: 1145395.34 bps (1.15 Mbps)


Average Throughput: 1128891.43 bps (1.13 Mbps)
```



b)



c)

d)

- **Queueing Delay in Q1 vs. Q3:**
  - Q1 has higher and more unpredictable delay peaks.

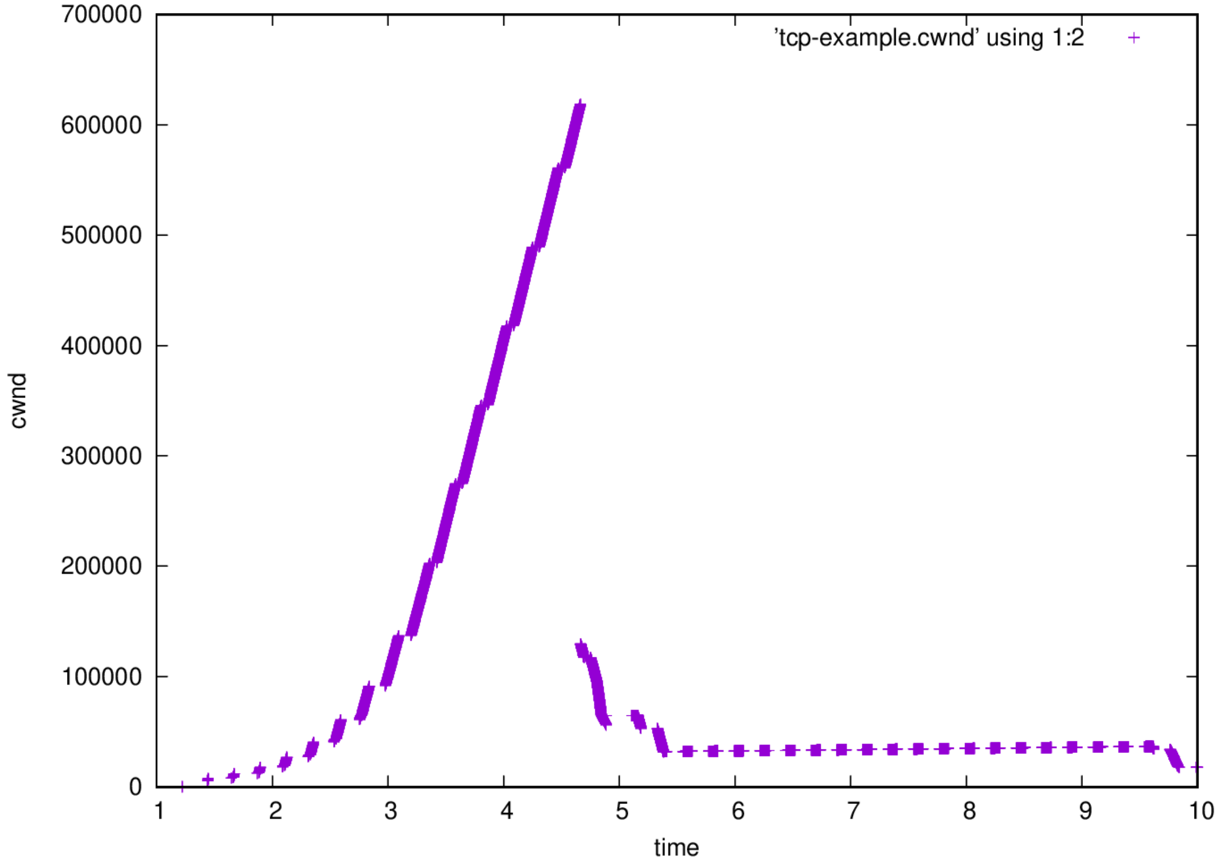
    - Q3 shows lower, steadier delays with quicker stabilization.
- **Impact of Bandwidth Increase in Q3:**
    - N1-N2 bandwidth increased to 10 Mbps, reducing queue buildup.
    - Resulted in smoother packet transmission.
- **Effect of 100ms Propagation Delay:**
    - Ensured consistent packet arrivals.
    - Contributed to stabilizing delays.

Using GNU Plot:

Cmd to run GNU plot:

- “gnuplot example.gpp”
- “evince cwnd.eps”

GNU Plot Screenshot:


'tcp-example.cwnd' using 1:2 +
cwnd
time
0
100000
200000
300000
400000
500000
600000
700000
1
2
3
4
5
6
7
8
9
10