Aryan Behal

2019026

CN Assignment - 3

Q1

a)

→ Size of packet = 1460 bytes = 11680 bits

→ Total number of packets received at n2 = 10^5

→ total data at n2 = 1168Mb

Total time taken = total data/5 + 0.025 considering delays. 5Mbps is the bottleneck speed as link between n1 and n2 is 5Mbps but link between n0 and n1 is 10Mbps. So after receiving first packet, n1 will have to wait for no time as packets will accumulate at it.

Maximum expected theoretical value of throughput is the limit of throughput as the amount of data becomes very large. So, maximum expected theoretical throughput is

$$\lim_{n\to\infty} \frac{n}{\frac{n}{5}+0.025} = 4.99 \text{ Mbps}$$

b) RTT = 2 * (0.01+0.015) sec, Data Link Capacity = 5 mbps = $5x10^6$ bps, size of packet = 1460 bytes = 1460*8 = 11680 bits

→ So BDP = (Data link capacity x RTT)/ (size of 1 packet) = 21.404 = 21 packets (rounded down)

c)

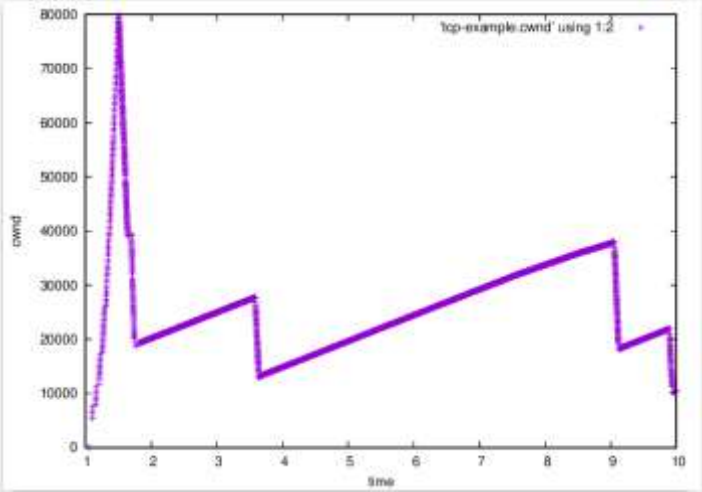
Average Throughput of TCP transfer = 3.895 Mbps

d) No, the computed throughput (= 3.895Mbps) is lesser than maximum expected throughput (= 5Mbps).

The throughput depends on factors like queuing delay (due to queue size), delays in network link, error rate of simulation.

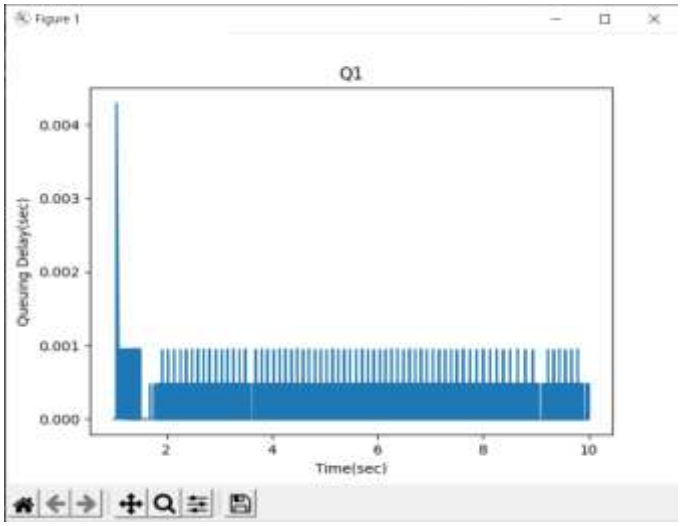
Limited queue size and high congestion rates led to packets dropping, thus, decreasing the throughput by wasting the bandwidth.

e)

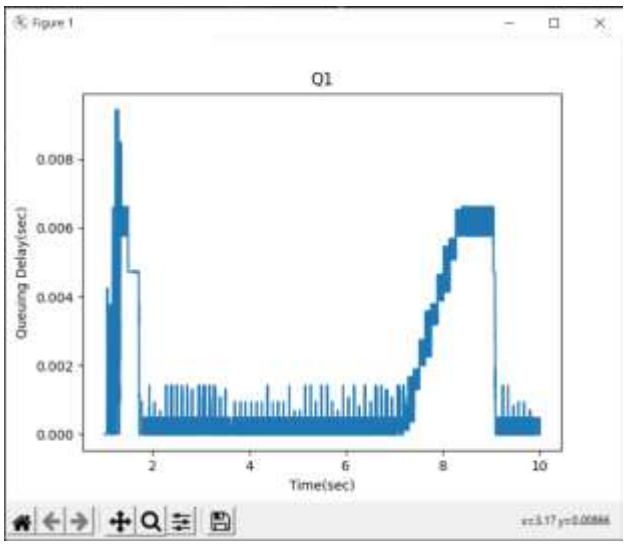


f)

Plot for N0



Plot for N1



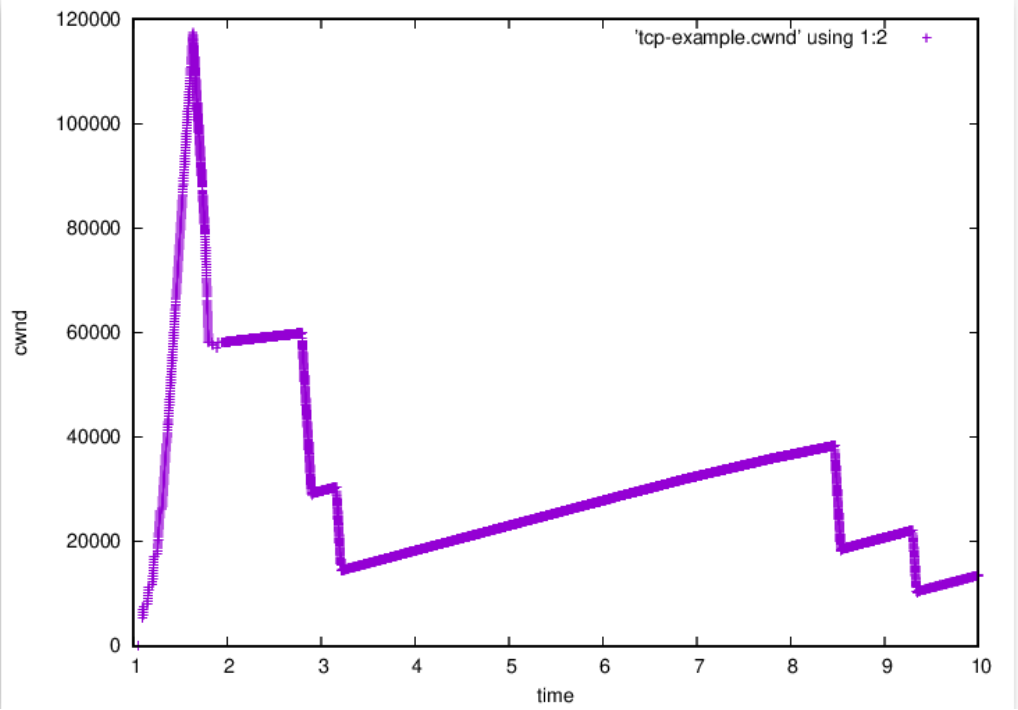
g) Yes, the plots are related. The congestion window and queuing delay increase linearly in size as transmission rates increase. But as transmission rate reach a threshold, the congestion window is reduced to half to reduce the congestion and queuing delay. They again start to increase linearly and the same process is repeated. This process can be easily noticed from the graphs for queuing delay(n1) and congestion window with time.

Q2

a)

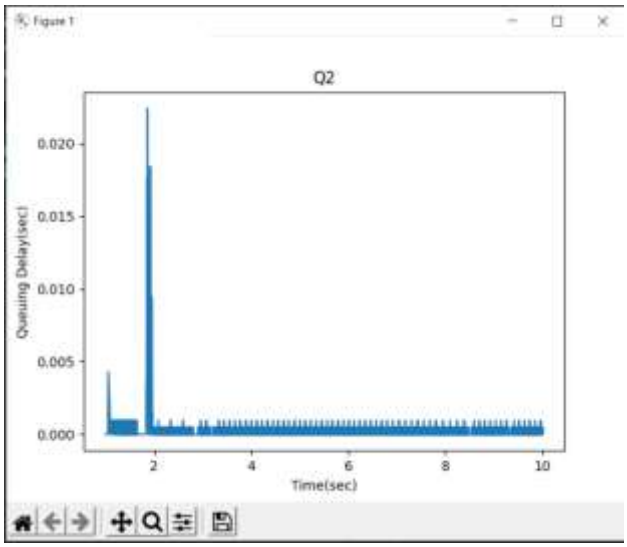
Average Throughput of TCP transfer = 4.024 Mbps

b)

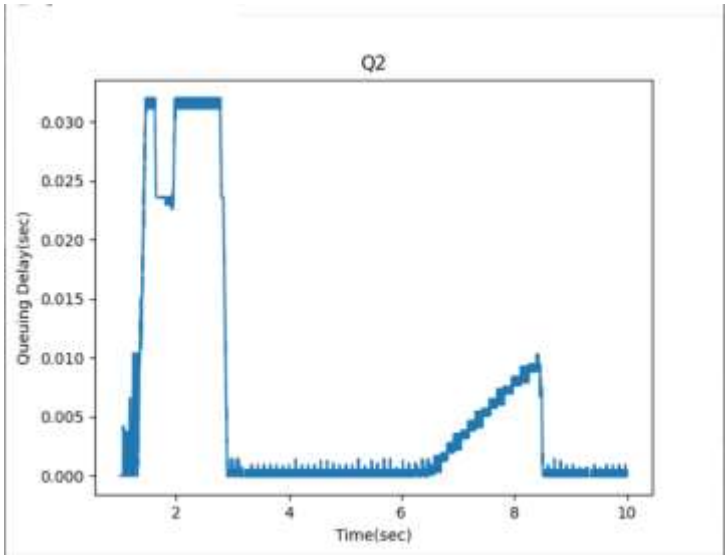


c)

Plot of N0



Plot of N1



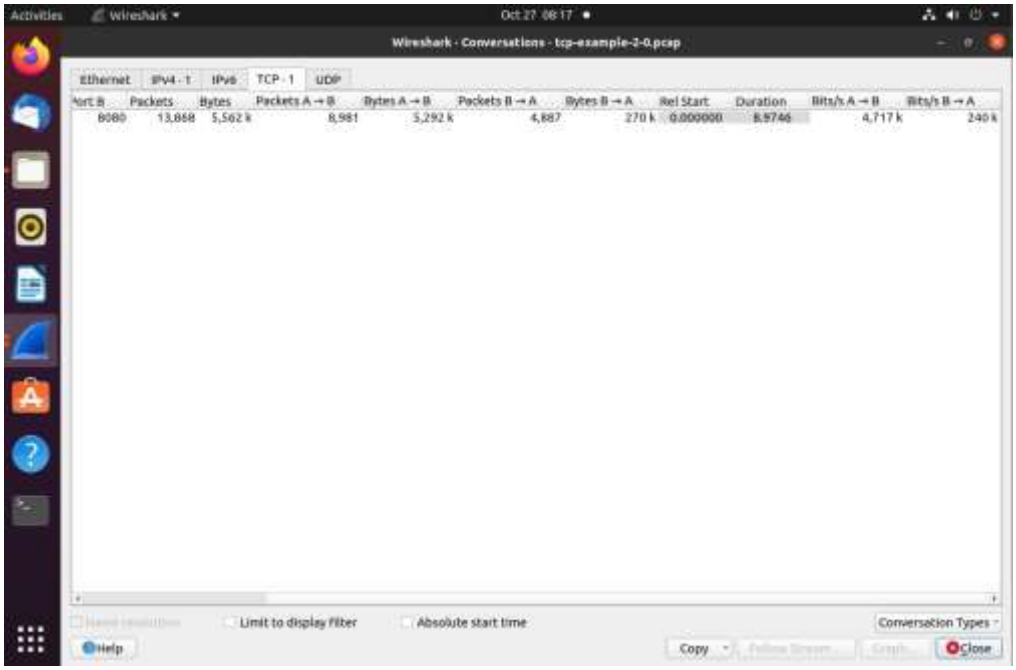
d)

→ Increase in queue size to 50 leads to more buffer space for packets. So CWND for 2) is 120k but only 80k in 1)

→ Bigger queue size leads to leads to higher tolerance to congestion. So, reduce in congestion window is more gradual for bigger queue size than smaller size queue as we can see in the graph.

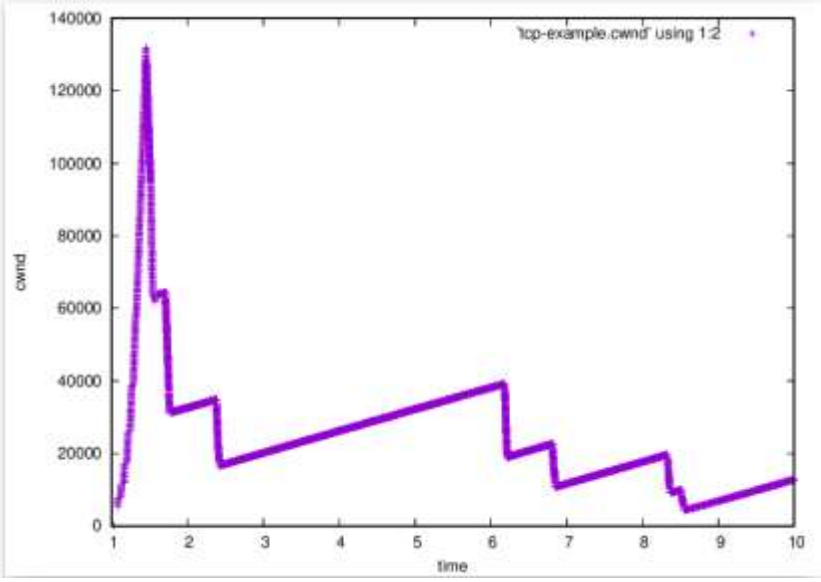
Q3

a)

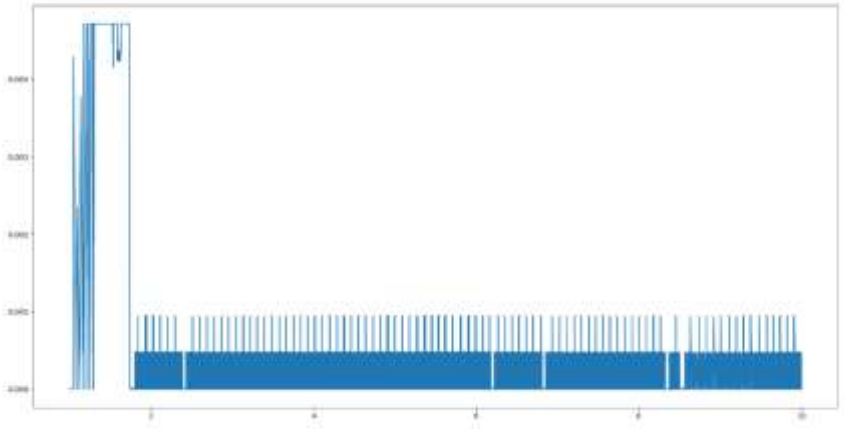


Average Throughput of TCP transfer = 4.717 Mbps

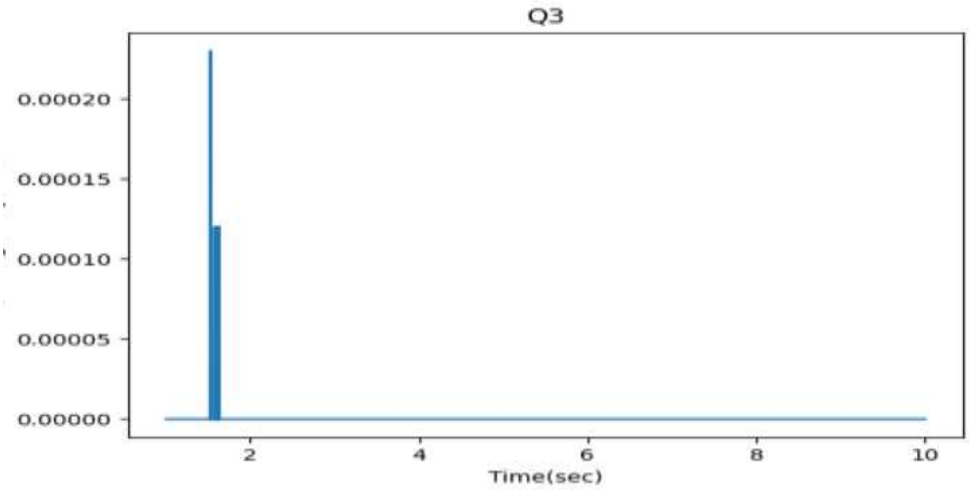
b)



c) Plot of N0



Plot of N1

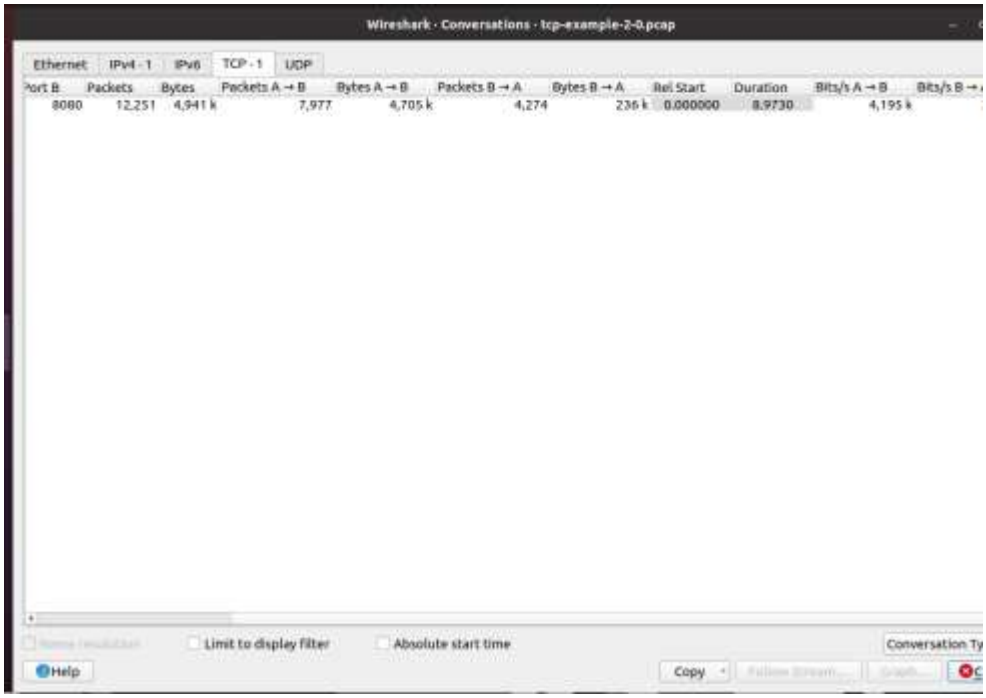


d)

For 3), the bandwidth and delays are the same for both links. Hence the queueing delay for 3) hardly happens as there is no congestion and is 0 for most of the time. But for 1), bandwidth and delay are different between the 2 links. Hence there will always be queuing delays and the graphs will not be flat. This can be observed from the 2 graphs.

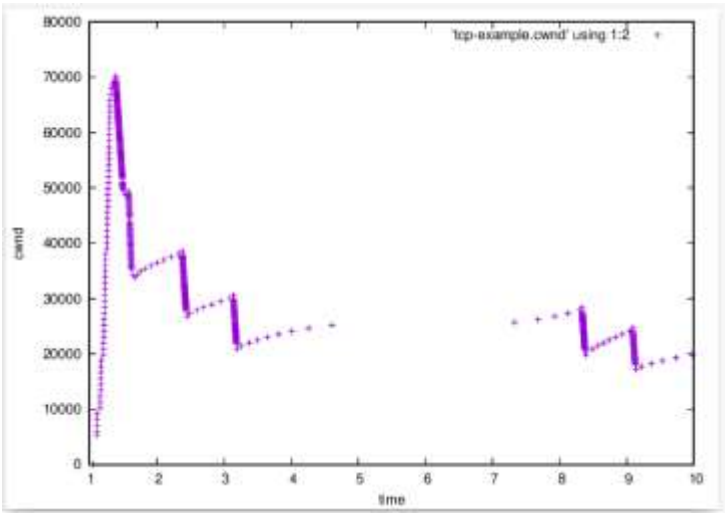
Q4

a)



Average Throughput of TCP transfer = 4.195 Mbps

b)



c)

→ The slow start stages are similar in both.

→ In the congestion avoidance phase, TCP Reno's congestion window growth function is AIMD (Additive Increase Multiplicative Decrease), whereas TCP Cubic's is the cubic function. Thus, in Q1, there is a linear increase in the CWND size. TCP Reno checks linearly with the rise in the bottleneck bandwidth whereas TCP Cubic checks cubically. Hence, TCP Cubic

initially increases fast but slows down the increase in the size of CWND as it reaches the maximum value. So, it never overshoots the maximum by a large value.

→ For fast recovery phase, in TCP Reno, congestion window is reduced to half but for TCP Cubic, congestion factor changes differently.
