CSE 232: Assignment 3

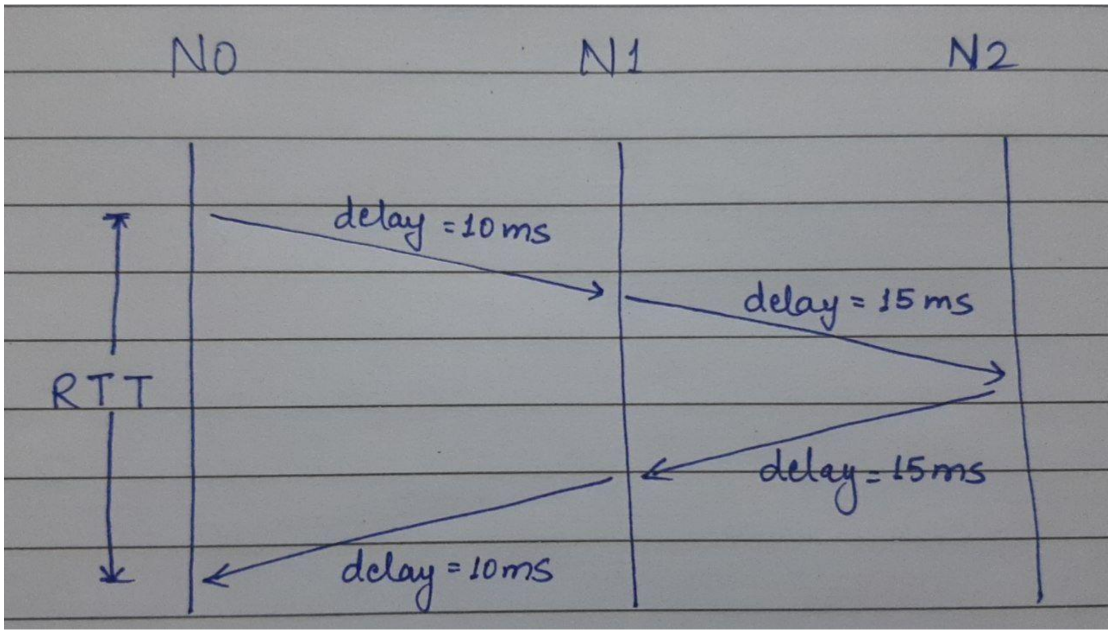
Kesar Shrivastava

2019051

1.a.

The maximum expected value of throughput is **5Mbps**. This is because it is the minimum bandwidth in the given connection and hence the bottleneck throughput.

1.b.



RTT = 2 * (10 + 15) ms = 50 ms

Throughput = 5 Mbps

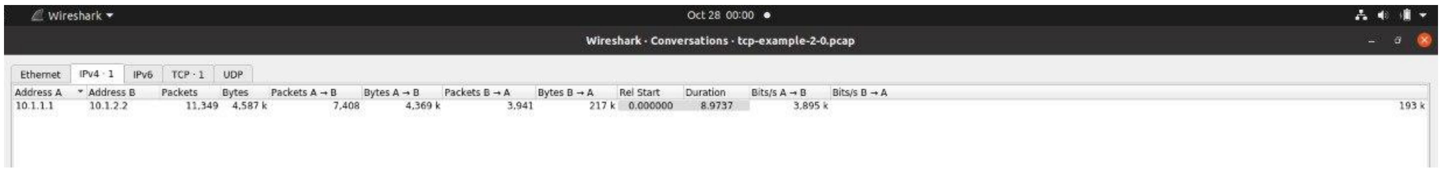
Given size of packet = 1460 bytes

Bandwidth Delay Product = (Data link capacity x RTT)/ (size of a packet)

= $(5*50*10^6/10^3)/(1460*8)$ = **21.40 packets**

= **21 packets (approx)**

1.c.



Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
10.1.1.1	10.1.2.2	11,349	4,587 k	7,408	4,369 k	3,941	217 k	0.000000	8.9737	3,895 k	193 k

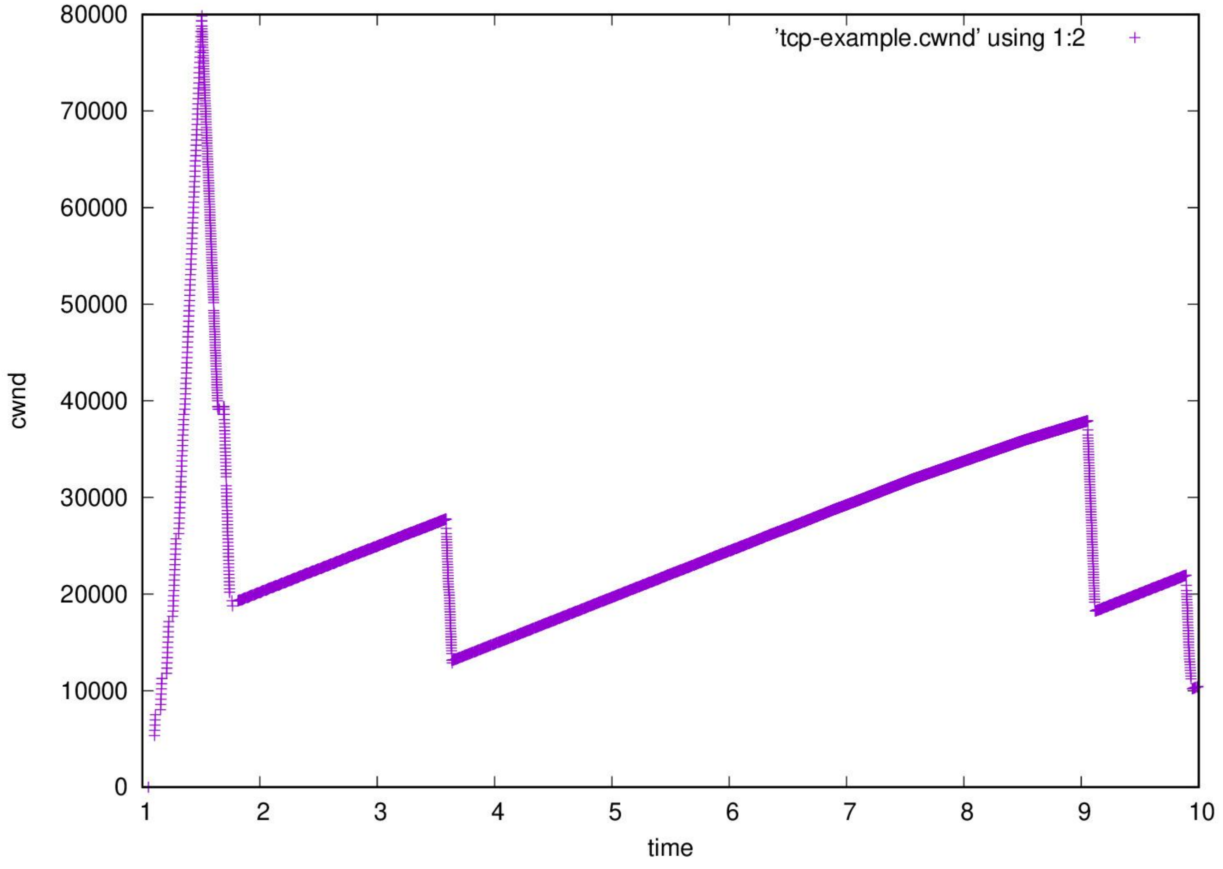
Average throughput = 3895kb/s (A →B) +193kb/s (B →A) = 4088 kb/s

4.088 Mbps

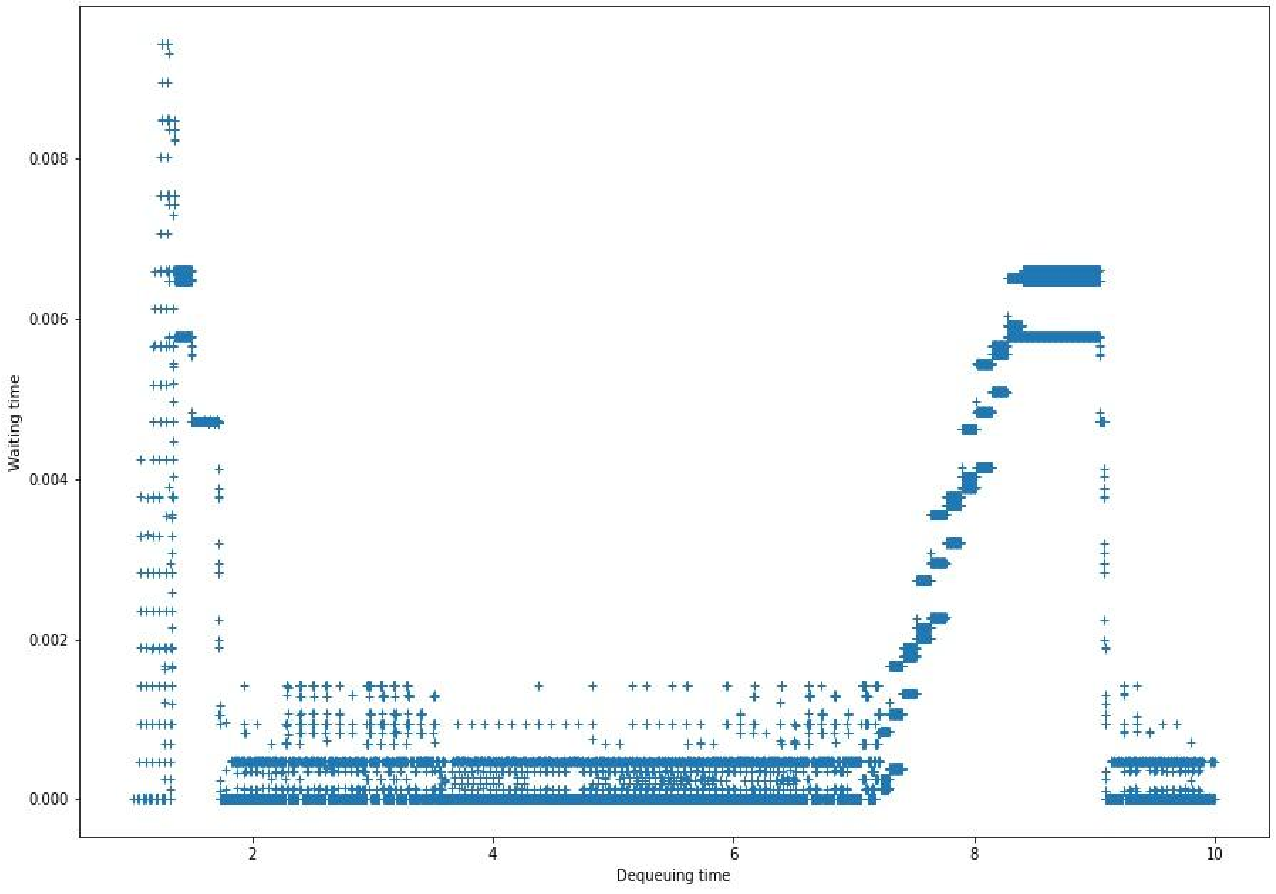
1.d.

The average computed throughput is less than the expected throughput. This is because congestion, queuing delay, delay in the network link, and the error rate in simulation also affect the throughput. The expected throughput is the ideal throughput that does not take into account other factors. On reducing the delays and increasing the queue size, we can improve throughput and make it go closer to 5Mbps.

1.e. CWND with time



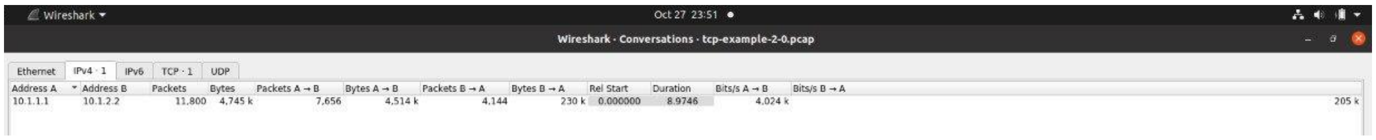
1.f. Queueing delay



1.g.

Yes, both the plots are related. The CWND size increases linearly. As the CWND increases, more packets can be sent in one RTT hence increasing the queue length at the router. After each RTT, the window size doubles spiking the CWND graph. This spike can be seen in the queuing delay plot too because as the CWND increases the sender sends more packets leading them to get queued up. Now, to reduce the congestion, the sender halves the CWND and hence we see a drop in the queuing delay. A similar trend can be seen in the rest of the graph. Higher CWND leads to higher queueing delays because though there is a large queue size, the packets still have to wait in the queue.

2.a.

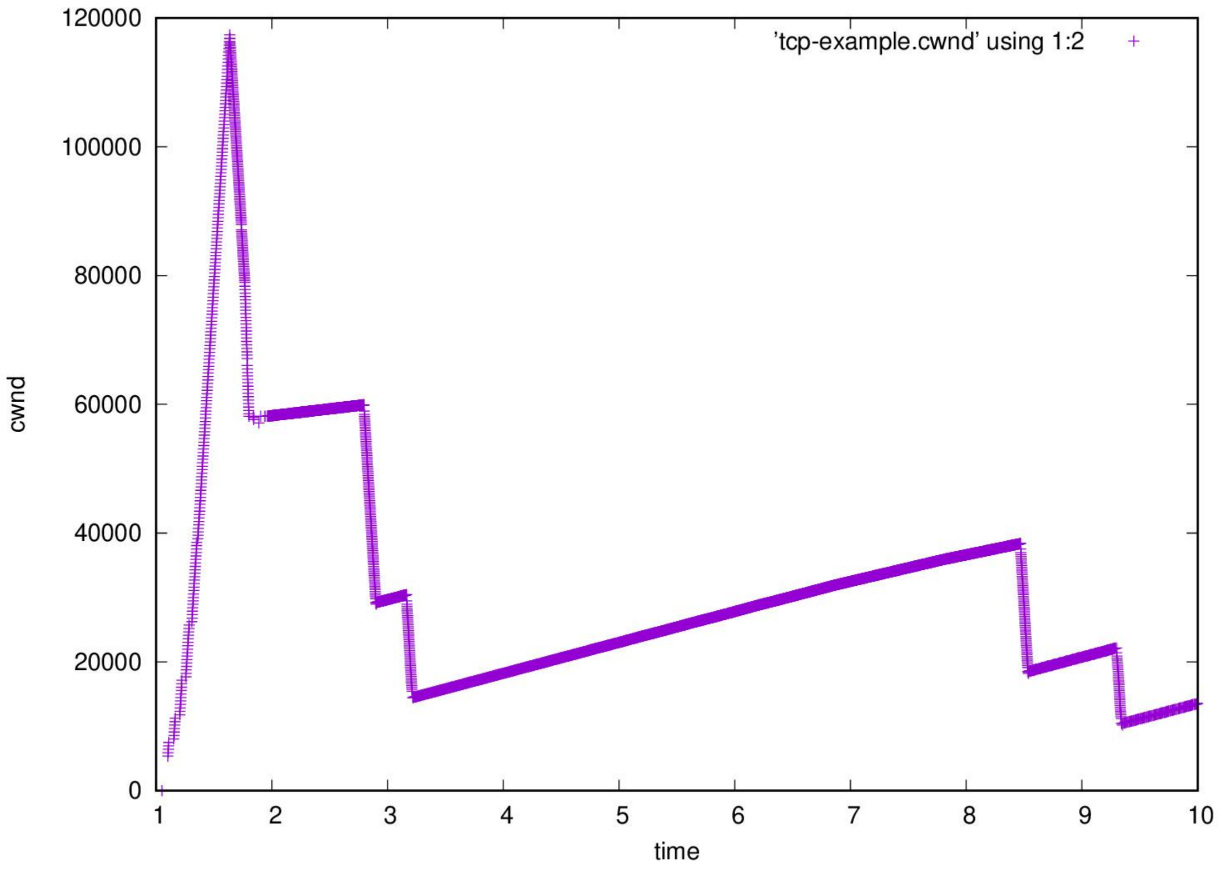


Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
10.1.1.1	10.1.2.2	11,800	4,745 k	7,656	4,514 k	4,144	230 k	0.000000	8.9746	4,024 k	205 k

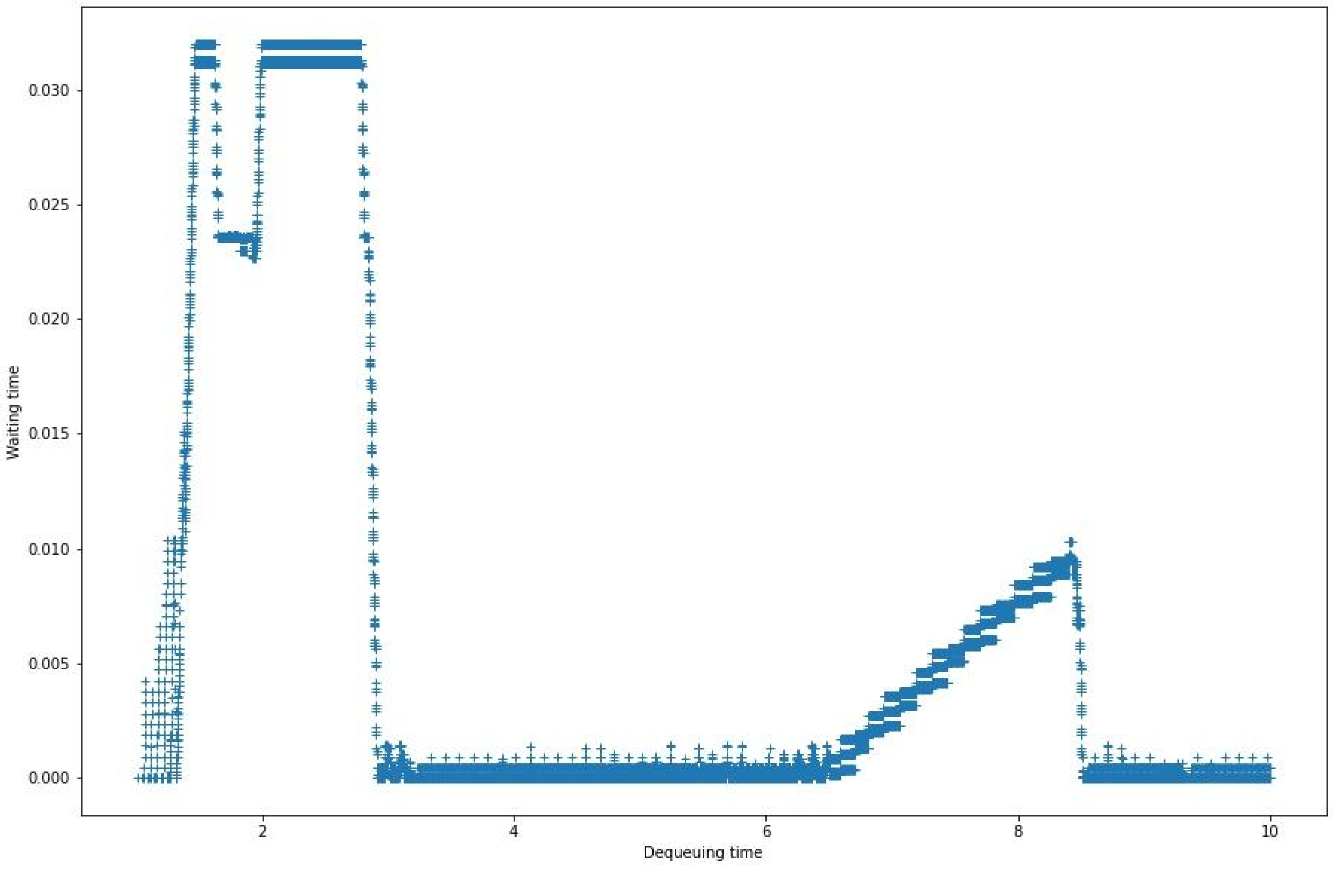
Average throughput = 4024kb/s (A →B) +205kb/s (B →A) = 4229 kb/s

4.229Mbps

2.b.



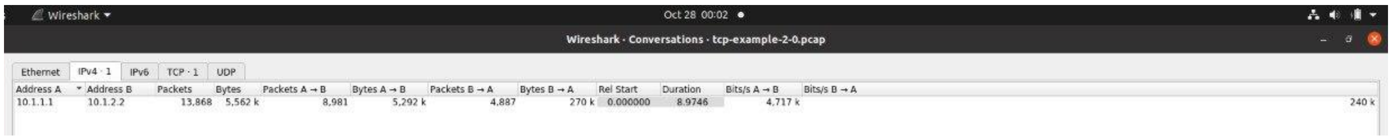
2.c. Queuing, delay



2.d.

Higher queue sizes lead to high congestion tolerance. Thus, the CWND can take greater values until the congestion avoidance phase is reached. The drop in CWND size is much less sudden for the queue size of 50 than for the queue size of 10. A higher queue size in Q2 means CWND can reach higher values before congestion avoidance. This is why CWND goes 120,000 in Q2 while only 80,000 in Q1. However, due to the high queue size in Q2, the queuing delay is more significant than in Q1. Since the queuing delay directly indicates the congestion of a network, we see higher congestion values in Q2 than in Q1. However, after the congestion avoidance, the queue size does not matter as the bandwidth-delay product is the same, so CWND too has similar values.

3.a.

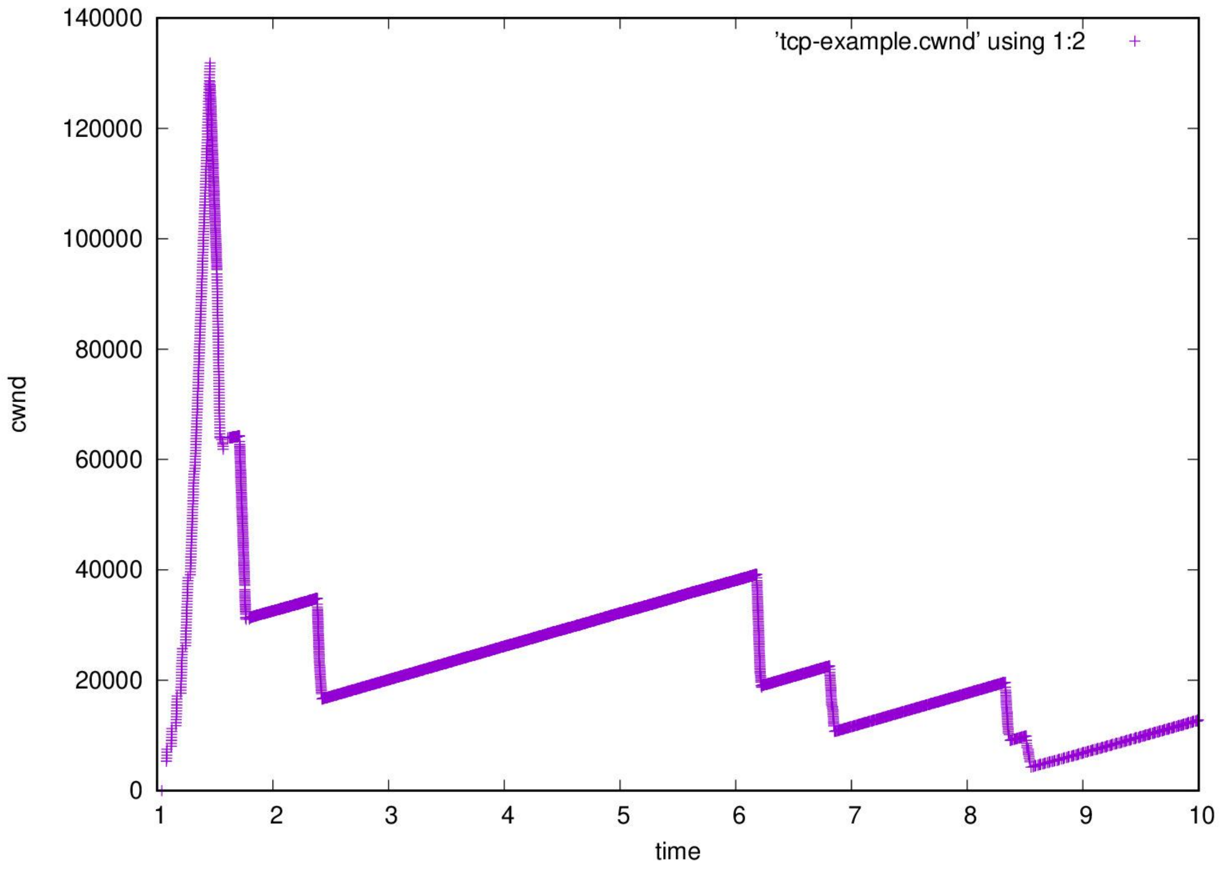


Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
10.1.1.1	10.1.2.2	13,868	5,562 k	8,981	5,292 k	4,887	270 k	0.000000	8.9746	4,717 k	240 k

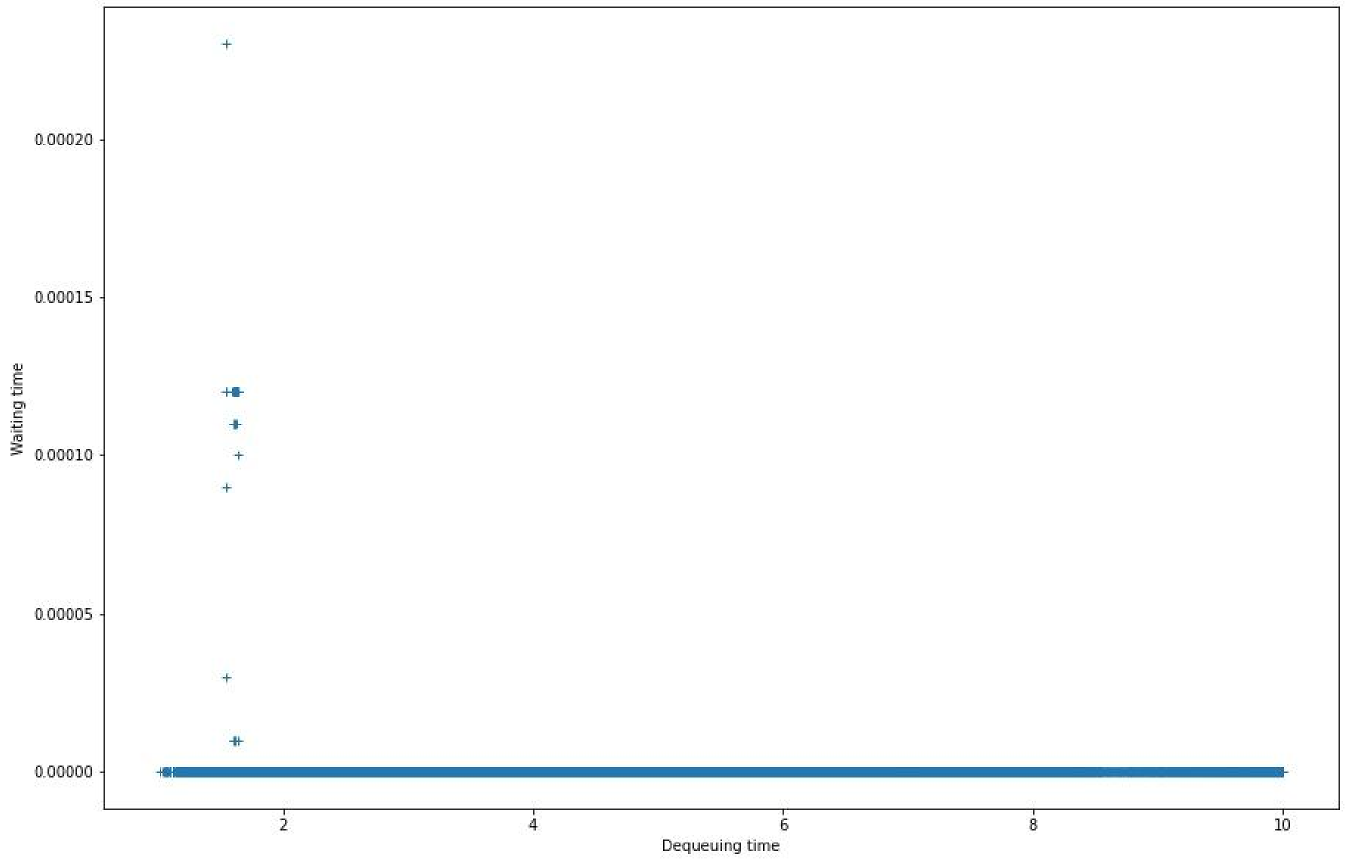
Average throughput = 4717kb/s (A →B) +240kb/s (B →A) = 4957 kb/s

4.957 Mbps

3.b.



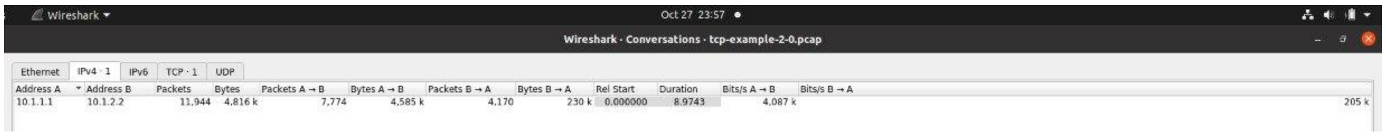
3.c. Queuing delay



3.d.

The N0 - N1 bandwidth is greater than the N1 - N2 bandwidth in Q1. Here, the rate at which the sender sends the packets to N1 is much less than the rate N1 sends to the receiver. This leads to the queuing up of packets at N1. Hence the packets have to wait in the queue extending the queuing time. In Q3, the bandwidth of N0 - N1 is the same as the bandwidth of N1 - N2. Here, the rate at which the sender sends to N1 is the same as what N1 sends to the receiver. This is why we see an almost constant (or zero) queuing delay throughout the simulation. Due to congestion at N1 in Q1, we see queuing delay peaking; however, this is not seen in Q3.

4.a.

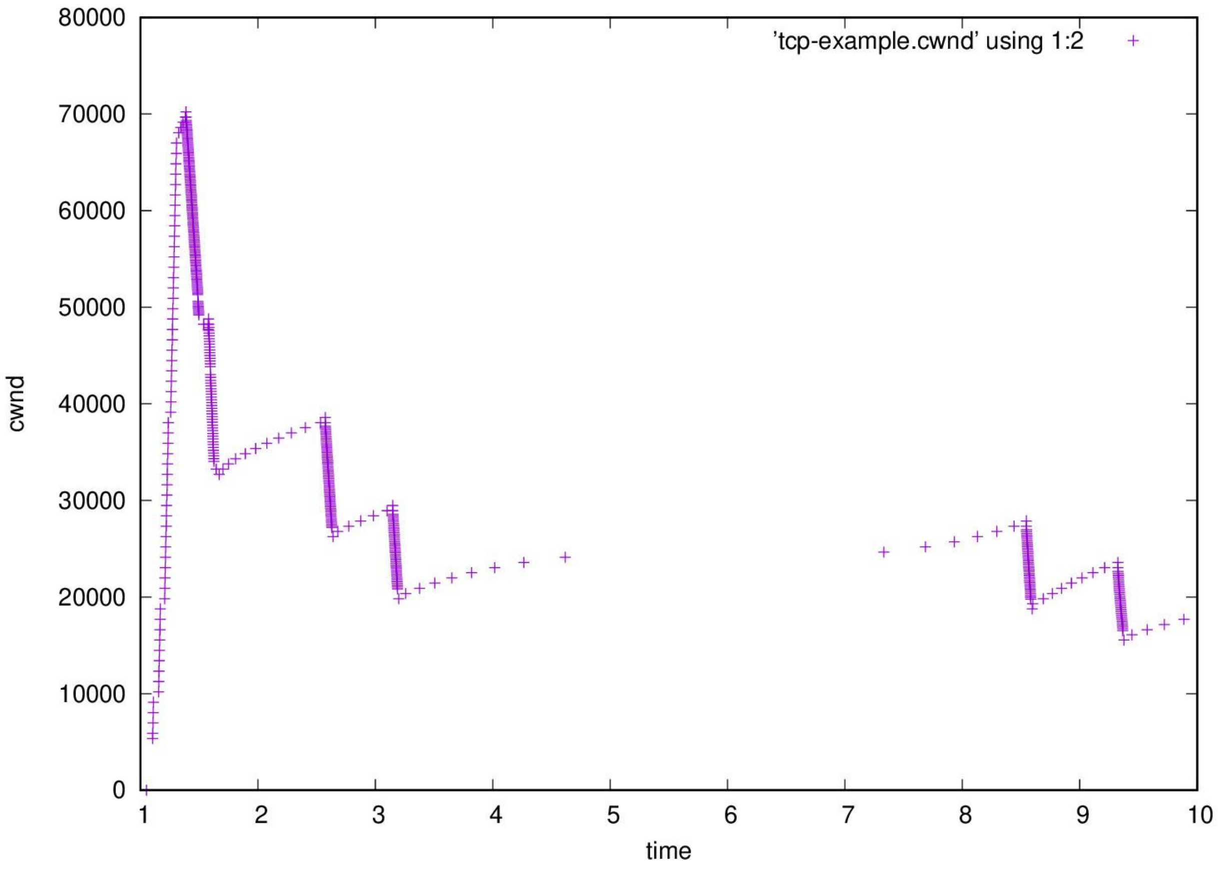


Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
10.1.1.1	10.1.2.2	11,944	4,816 k	7,774	4,585 k	4,170	230 k	0.000000	8.9743	4,087 k	205 k

Average throughput = 4087kb/s (A →B) +205kb/s (B →A) = 4292 kb/s

4.292 Mbps

4.b.



4.c.

The main difference between the both is the way they adjust the CWND. The slow start phase is similar.

There is a **difference in the congestion avoidance phase.** TcpNewReno uses AIMD (additive increase multiplicative decrease) hence the CWND increases linearly, but as the name suggests, the growth is cubic in TcpCubic. TcpCubic bandwidth probing method has higher throughput because when bottleneck bandwidth increases, it probes cubically, whereas Reno probes linearly.

The fast recovery phase is the same in both.

Script used to plot queuing delay

```
import matplotlib.pyplot as plt
f = open("tcp-example1.tr",'r')
lines = f.readlines()
queue = []
x_axis = []
y_axis = []
for line in lines:
  line = line.split(" ")
  if line[0]=='+' and line[2][10]=='1':
    queue.append(float(line[1]))
  if line[0]=='-' and line[2][10]=='1':
    x_axis.append(float(line[1]))
    waiting_time = float(line[1])-queue[0]
    y_axis.append(waiting_time)
    queue.pop(0)
plt.xlabel("Dequeuing time")
plt.ylabel("Waiting time")
plt.plot(x_axis,y_axis,"+")
plt.show()
```