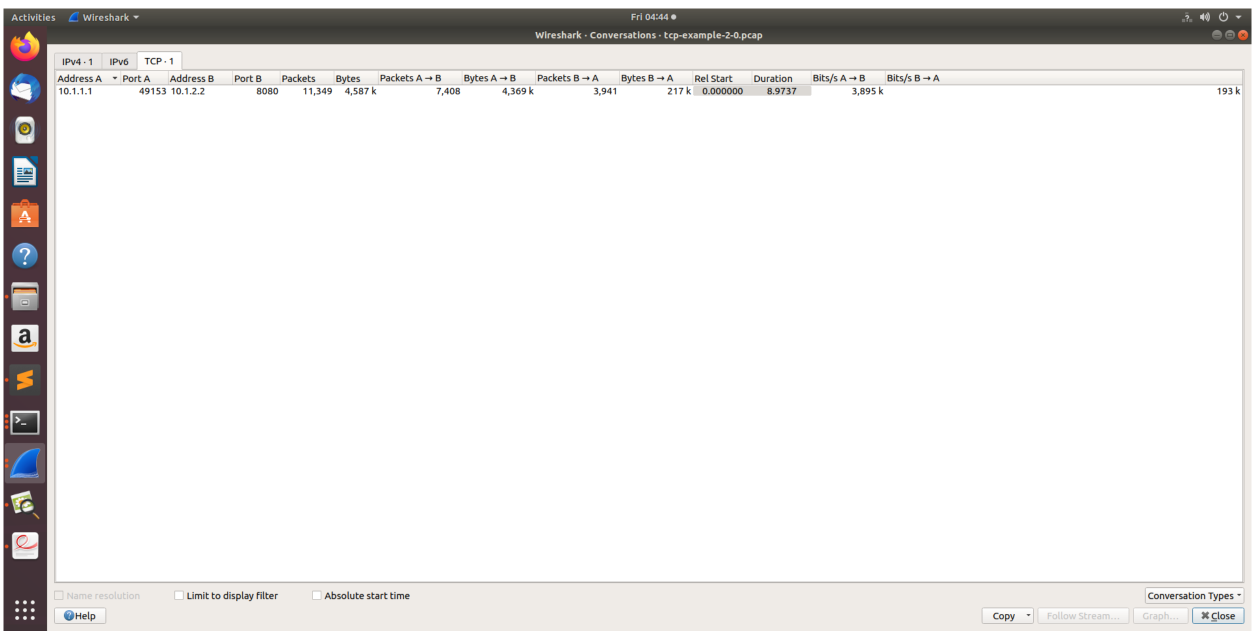
ASSIGNMENT 3
Computer Networks
Sneh Suman 2019337

Sol 1-

a)The maximum possible throughput theoretically is 5Mbps. This is because it is the bottleneck value present to us at node2.

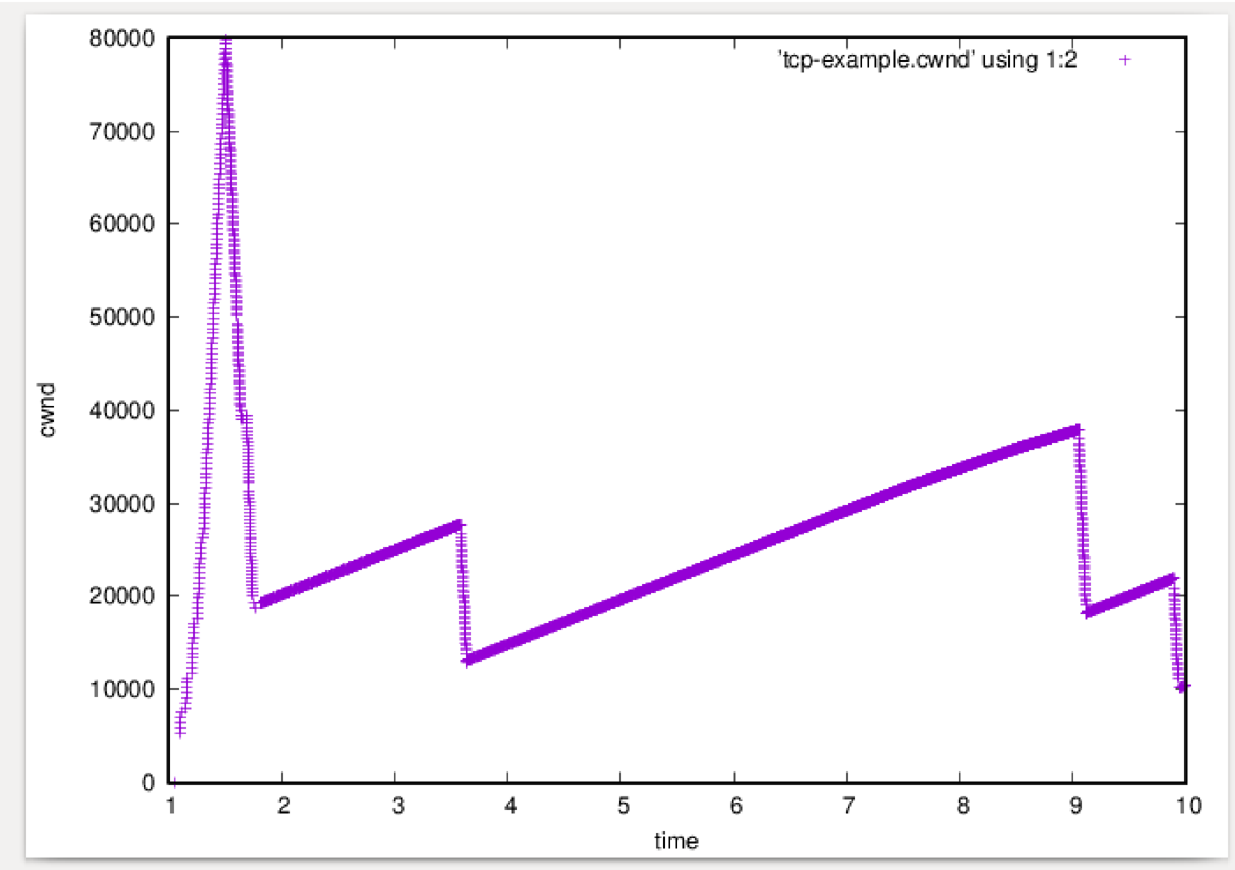
b)The bandwidth =5Mbps
Total delay=10ms+15ms(from n0n1 and n1n2)
RTT=2*25=50ms
Bandwidth delay product=5*50=250kb
Application payload size=1460Bytes=1460*8bits
No. of packets transferred in the process=250kb/1460*8=21.4=21(approximately)

c) The average throughput can be calculated by the value we get from the data from Wireshark 4587 bytes and time is 8.97 seconds.
Avg.Throughput =4587/8.97*8/1000000=4.09Mbps .



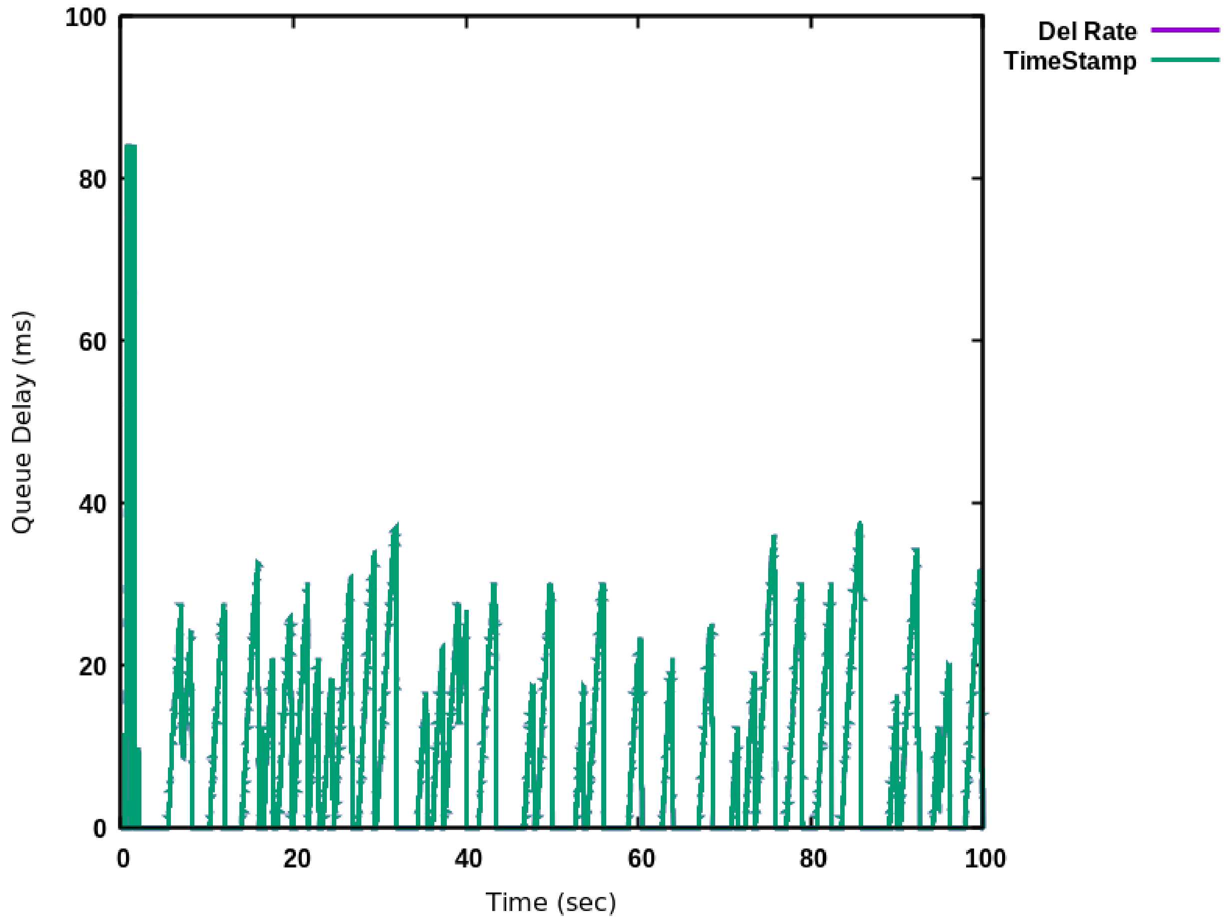
d)No, the throughout from a) and c) are not the same. It is because of the fact that we are getting delays from the network queue and another transmission, propagation type of delays. This delay is resulting in the difference between the possible and the actual throughputs.

e)



'tcp-example.cwnd' using 1:2
cwnd
time
80000
70000
60000
50000
40000
30000
20000
10000
0
1
2
3
4
5
6
7
8
9
10

f)



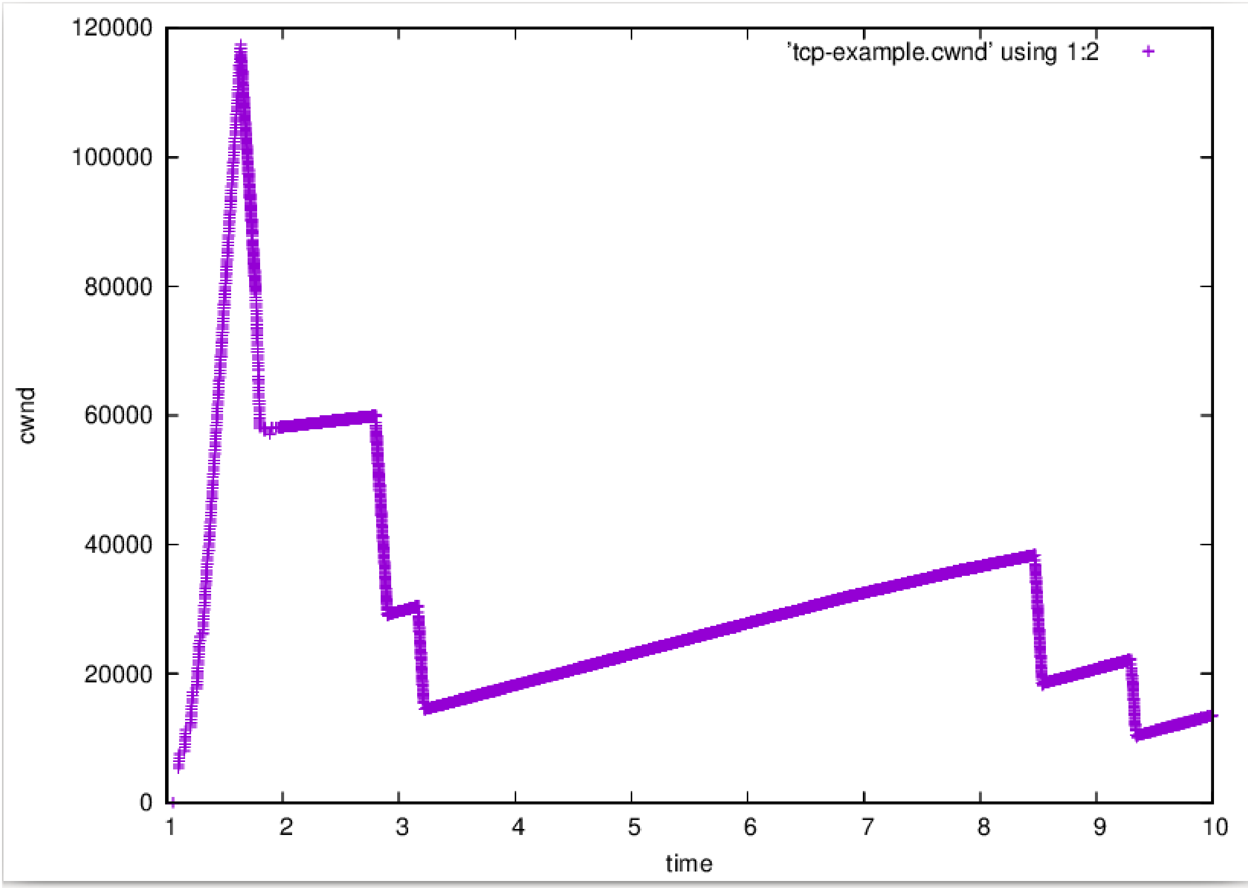
g)

Yes, the plots are related. We can see that the queueing delay graph has higher values when the congestion plot is also high. Also, we can see that the line is solid in the congestion window where a lot of lines are seen in the queue delay plot. From these observations, we can say that if the queue delays are high we can expect high congestions as evident from the graph.

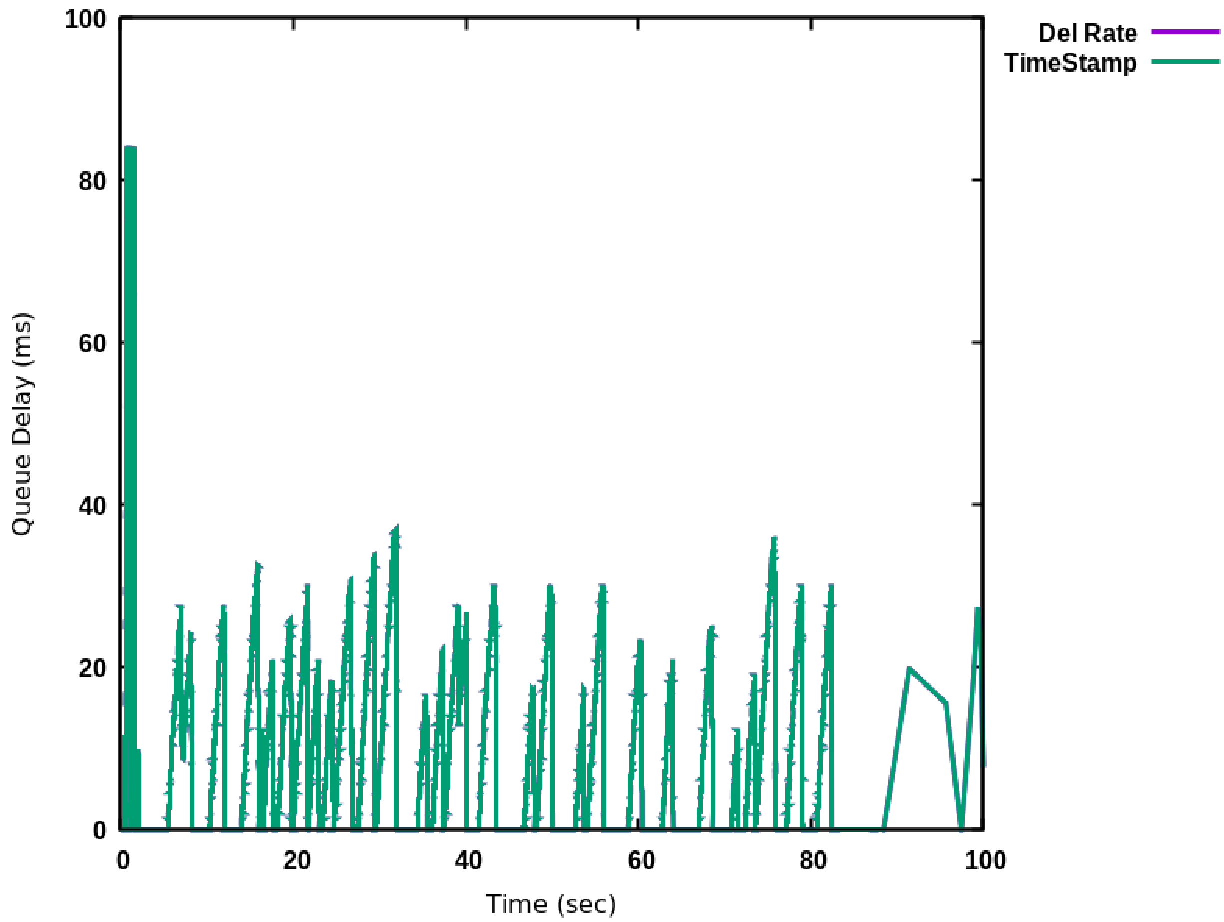
Sol 2-

a)Average throughput calculated 4745000/8.9746*8/1000000=4.22Mbps

b)


'tcp-example.cwnd' using 1:2
cwnd
time
0
20000
40000
60000
80000
100000
120000
1
2
3
4
5
6
7
8
9
10

c)



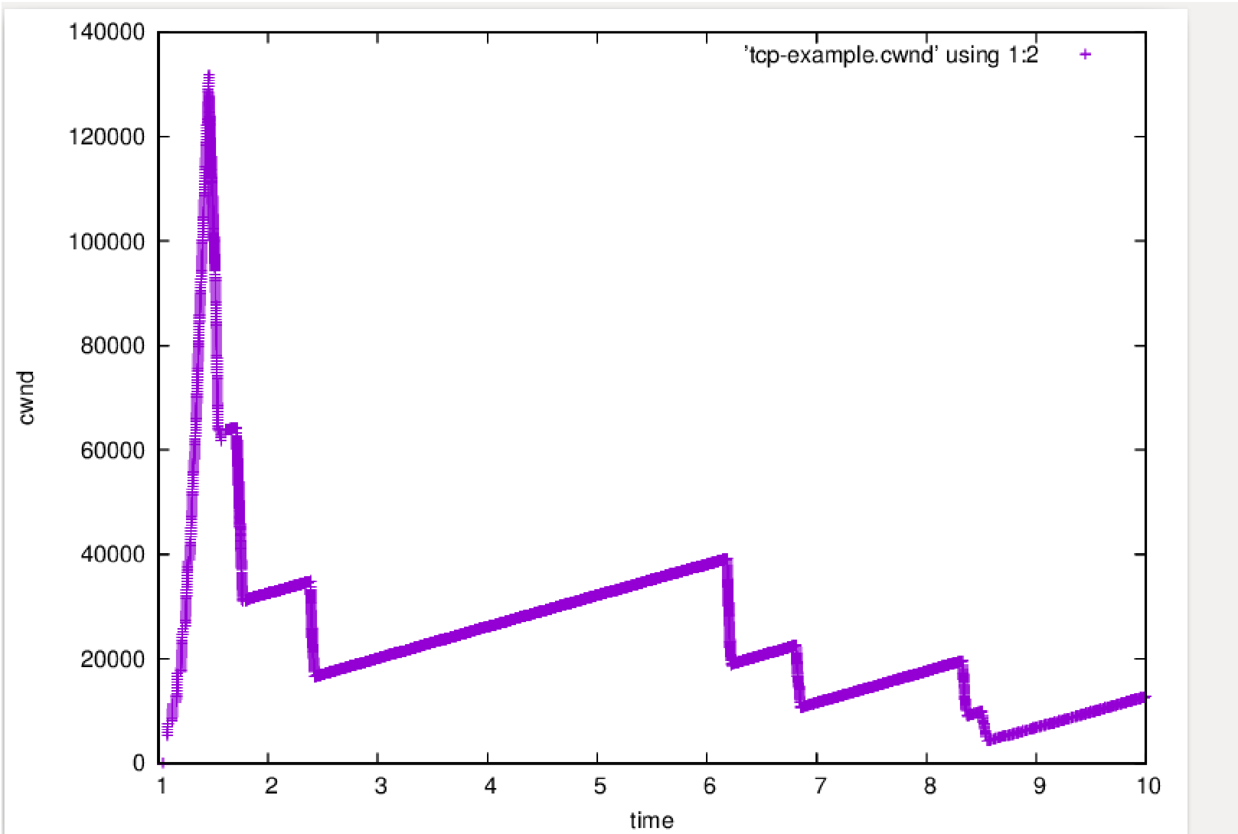
d)

After comparing the cwnd plots we can say that the fast recovery phase was established much better in the second case than in the first one. It's because we increased the queue size. The queue size increment provides the network with more data in the queue and it is able to try to make new congestion windows during the same time when it had made none previously. After that, the graphs are almost the same and we can say that increasing queue size helped in increasing the window formation rates.

Sol 3-

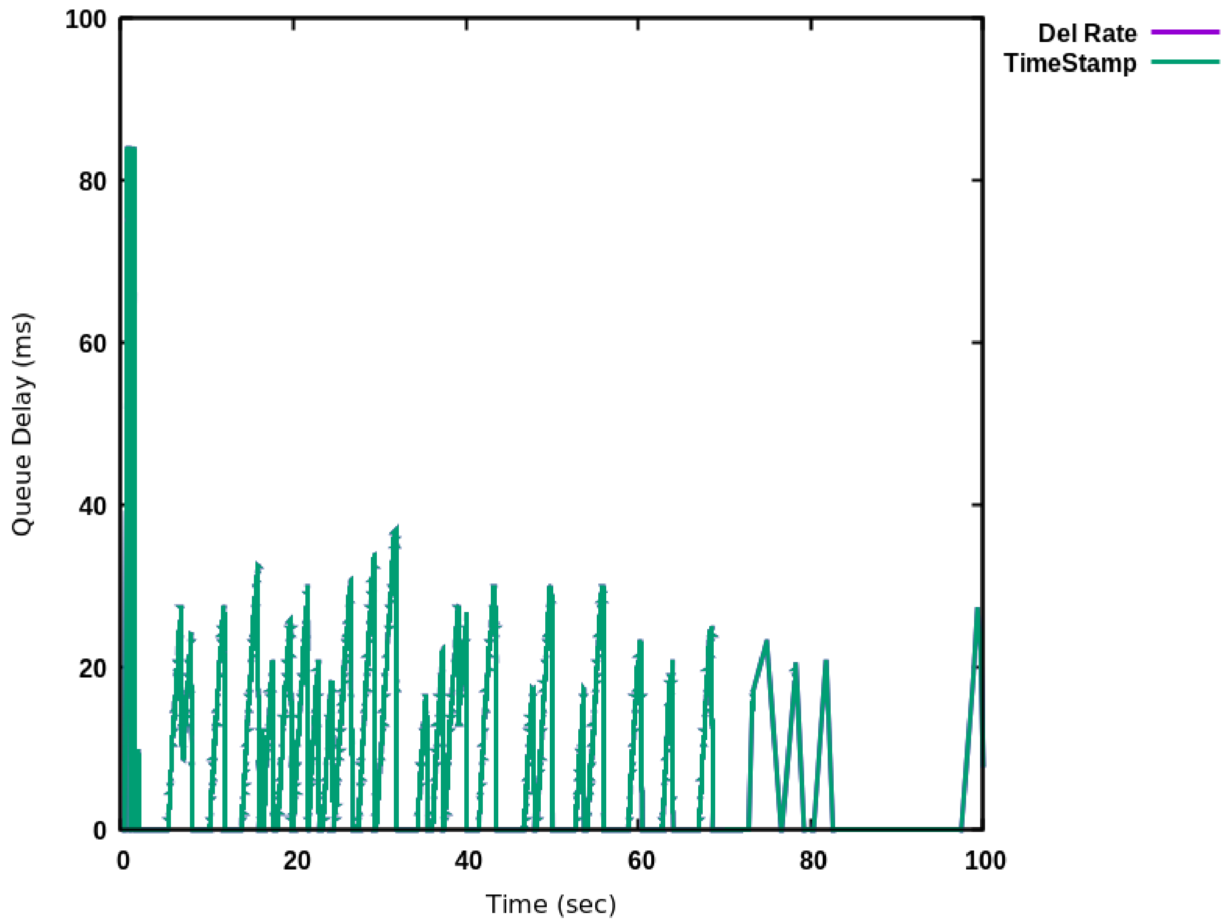
a)Average throughput=5562000/8.9746*8/1000000=4.957 Mbps

b)



'tcp-example.cwnd' using 1:2
cwnd
time
0
20000
40000
60000
80000
100000
120000
140000
1
2
3
4
5
6
7
8
9
10

c)



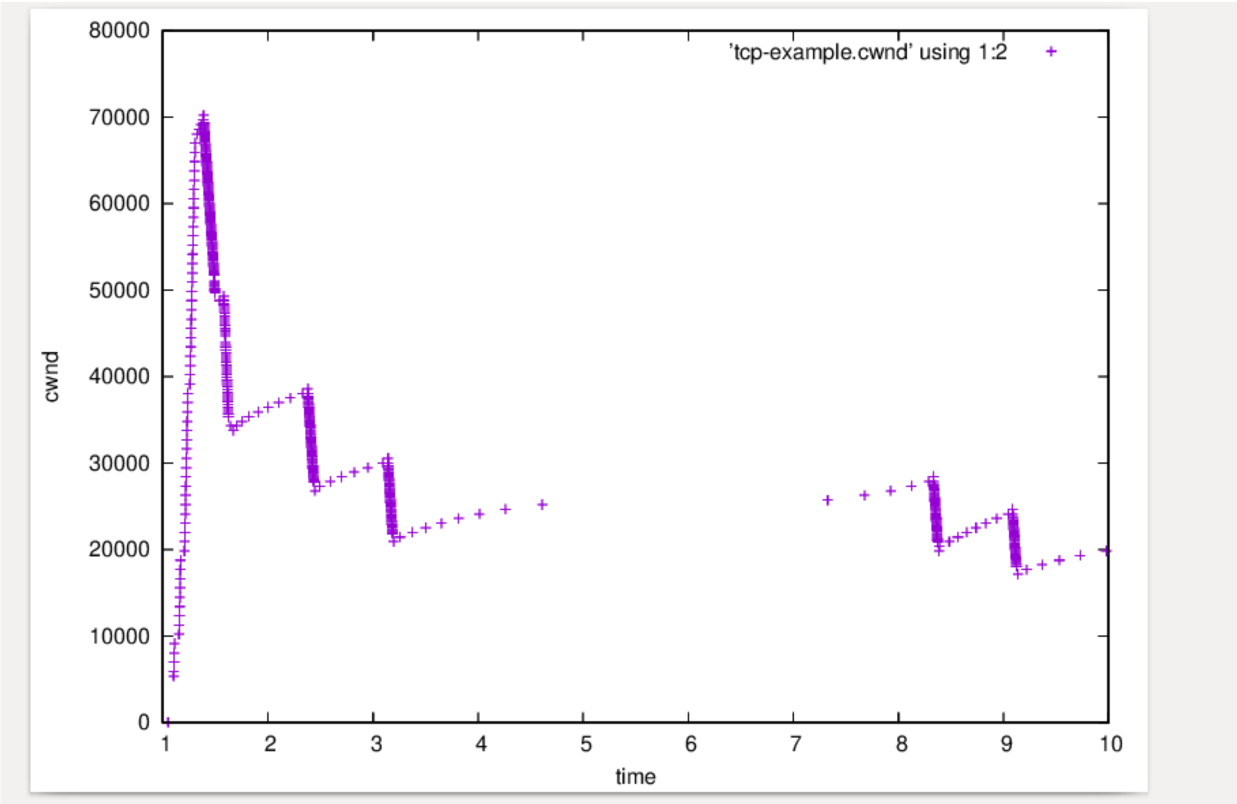
d)

Since the bandwidth is increased and the delay is also decreased. We can see that from the first and third plots that the graph has gone a little smaller in height as and the lines have eased out a little. It's because from first to here the delay time is more and buffer size is less. Also, we see that the lines have eased out because of the increase in delay time and decrement of the queue size.

Sol 4-

a)Average throughput obtained= 4941000/8.9730*8/1000000=4.405Mbps

b)



c)

From the two graphs, we can see that

At the very beginning, both are **slower**. But the cubic variation reaches the top or plateau. Then the graph becomes denser for the cubic variation. This means the bandwidth is getting higher here for cubic. Here we see that the new reno takes more time to reach the top. Then also it is still getting less data and is not yet a well-stabilized network.

Now, here when we look at the graphs we see that the new reno has **developed better** to **avoid congestion** of the network. The size of the congestion window for new reno is higher than cubic.

Now for fast recovery, we see that at the first time the network is trying to find to congestion window. In cubic, the window is found at around midway of the first value. The same is not the case with new reno. It takes a lot more time to get the congestion window. But we see that during the course of time when the network is being worked. Tcpnewreno picks up its speed and gets every new congestion window at around halfway marks than the previous one. And also cubic loses the congestion window for a few seconds before again catching it up. This shows tcpnewreno is **more stable** and keeps the network connected throughout.