Assignment 3 - COL334

Aayush Goyal

October 2021

Contents

1 Changing different Congestion Protocols

1.1 For each protocol, generate a plot having Congestion window size on the y-axis and time on the x-axis(till t=30s).

The following 4 plots have been obtained for the the different congestion protocols that have been mentioned in the assignment statement. The protocols are:

1. Tcp NewReno

2. Tcp HighSpeed
3. Tcp Veno
4. Tcp Vegas

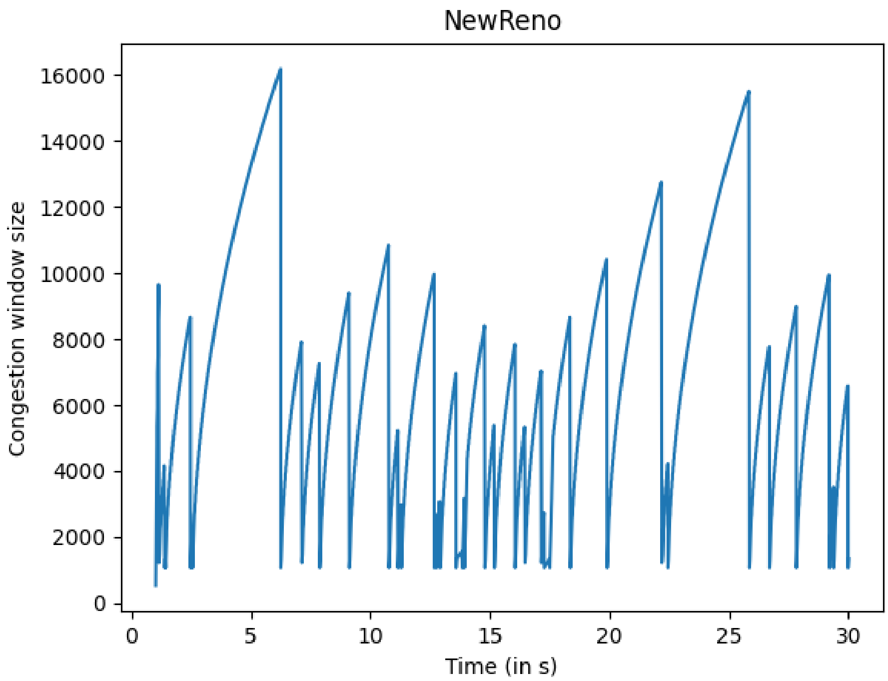

Figure 1: For the TCP New Reno protocol

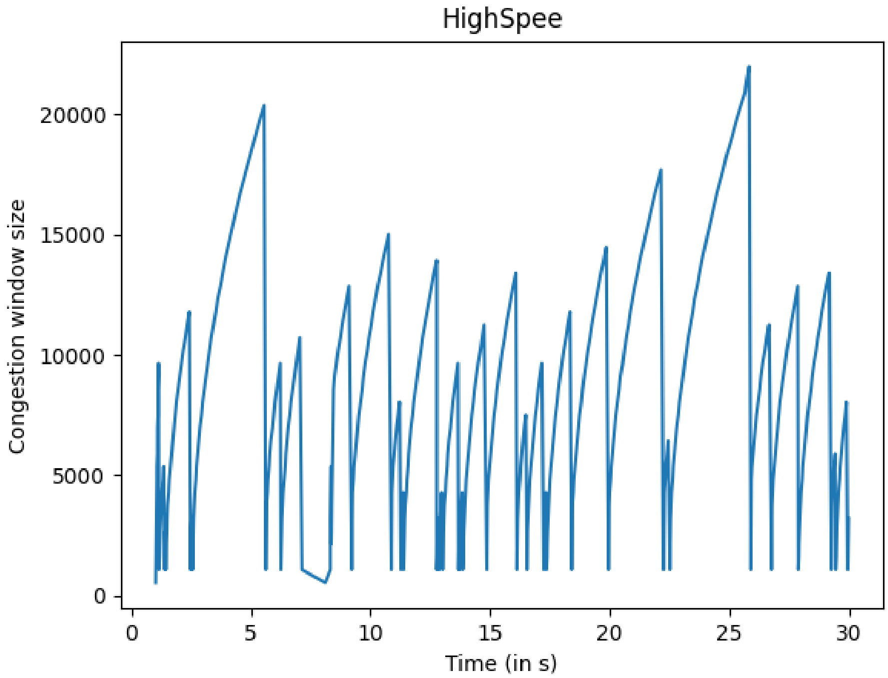

Figure 2: For the TCP High Speed protocol

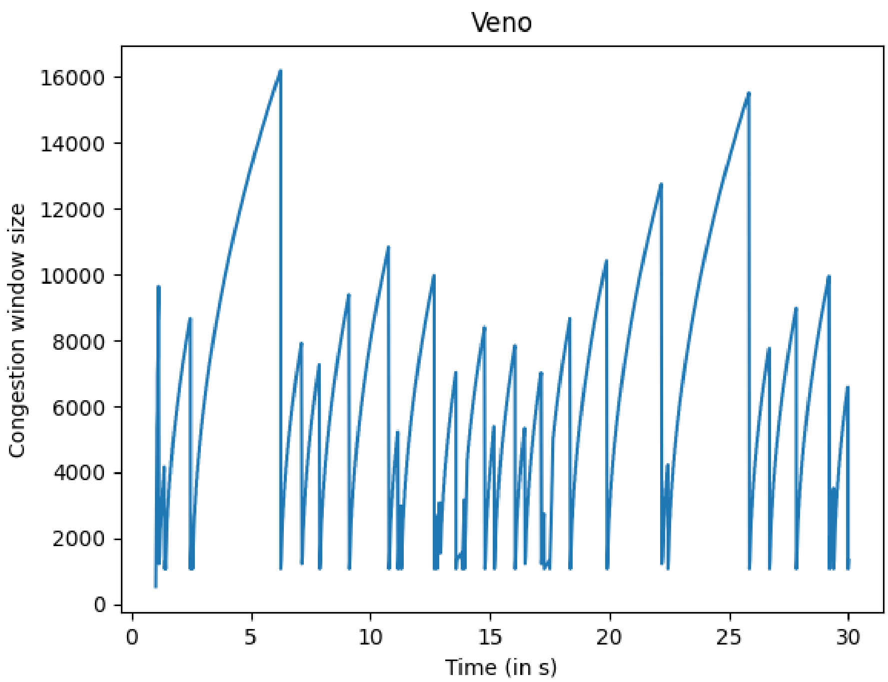

Figure 3: For the TCP Veno protocol

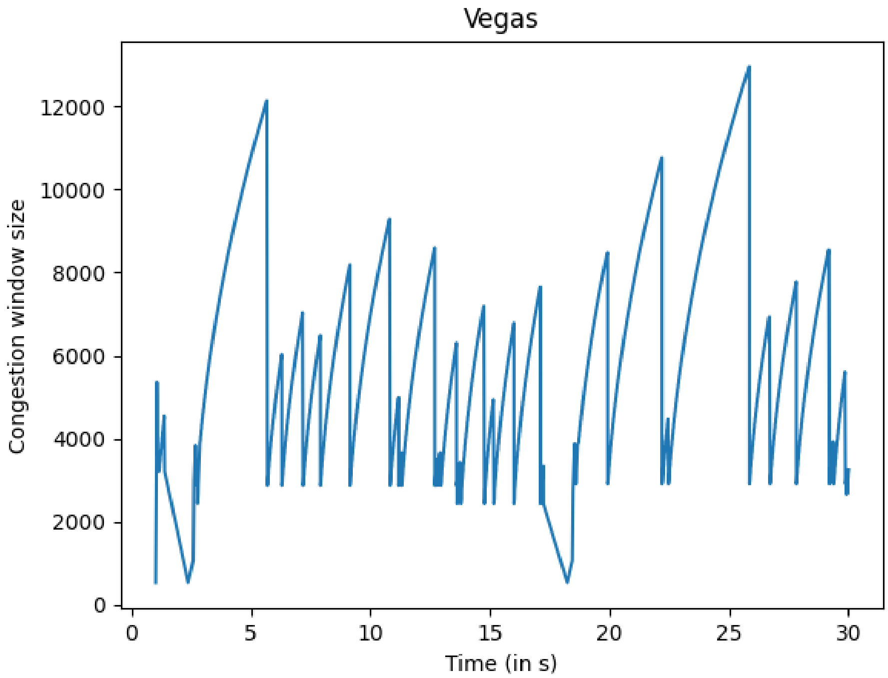

Figure 4: For the TCP Vegas protocol

1.2 For each protocol, find the number of dropped packets in total. What is your inference?

1. Tcp NewReno: 38
2. Tcp HighSpeed: 38
3. Tcp Veno: 38
4. Tcp Vegas: 39

The packet drop happens because of the crowding of the window. We have to drop a packet along with reducing the size of the window to about half. From the data we can see that all of them have almost equal number of packet drops (Tcp Vegas has one more packet drop). The packet drops can be observed from the graph as well. Whenever the size of the congestion window reduces it means there was a packet drop that happened. Generally it happens when we are in the congestion avoidance phase. For all of them it stays true that we can determine the number of packets drops using the number of times there is a decrease in the window size. The only thing different in these protocols is the speed at which the size increases.

1.3 For each protocol, describe what you observed (4-5 sentences per protocol is enough). You can talk about the trend you observed in the above plot, the algorithms they used for different phases etc.

1.3.1 TcpNewReno

1. During the slow start phase the size of the congestion window increases by the segment size. Thus we can see that the increase is quite linear in the start, that is the slow start phase.

2. During the congestion avoidance phase, it increases by $\frac{(seg-size)^2}{cwnd}$. Between that the size of window in reduced to half due to packet drop happening.

3. The peak of the congestion window size goes to about 16000 and the least value of it remains at around 2000. The average value stays at around the value of 8000.

1.3.2 TcpHighSpeed

1. This is a different protocol from the otther ones since there is nothing like SlowStart phase and Congestion avoidance phase.

$$\mathbf{cwnd \mathrel{+}= a(cwnd)/cwnd}$$

   here a(cwnd) is obtanied from the lookup table depending on the value of congestion window size. This can be seen from the huge if else condition given in the code of TcpHighSpeed.

2. The peak of the congestion window size goes to about 20000 and the least value of it remains at around 0. The average value stays at around the value of 12000.

3. When compared to the other graphs it can be seen that highspeed is a good method. The maximum is much more and it thus seems that the it will perform better than the other other ones. This is basically because of the absence of the slow start phase.

1.3.3 TcpVeno

1. During the slow start phase the size of the congestion window increases by the segment size. Thus we can see that the increase is quite linear in the start, that is the slow start phase.

2. During the congestion avoidance phase, it increases by $\frac{(seg-size)^2}{cwnd}$. Between that the size of window in reduced to half due to packet drop happening.

3. The Slow start and the congestion phase are both as that of TcpNewReno. This is the reason that both NewReno and Veno has almost similar graph. It includes some good factors of the Vegas also but they are not observable in this small time range.

4. The peak of the congestion window size goes to about 16000 and the least value of it remains at around 1000. The average value stays at around the value of 8000.

1.3.4 TcpVegas

1. It keeps a record of the RTT values and compares it to othetrs with the expected throughput. The difference between them provides an estimate of the backlog of the packets. The packet loss increases and decreases linearly with the number of backlog of packets.

2. The peak of the congestion window size goes to about 12000 and the least value of it remains at around 500. The average least value looks like to stay at around 3000. The average value of the congestion window stays at around 7000.

3. This seems to be the least best method among the others since it is actually based on the model that it works to avoid the packet drops. Thus it tries to prevent the drop instead of dropping them.

2 Changing different Data Rate and Application Data Rate

2.1 Plot the congestion window size vs time graph for the TCP connection at different Channel Data Rates (2Mbps, 4Mbps, 10 Mbps, 20Mbps, 50 Mbps) between N1 and N2. Use Application data rate as 2Mbps. You need to create a plot for each Channel Data Rate. Explain the trends that you observe.

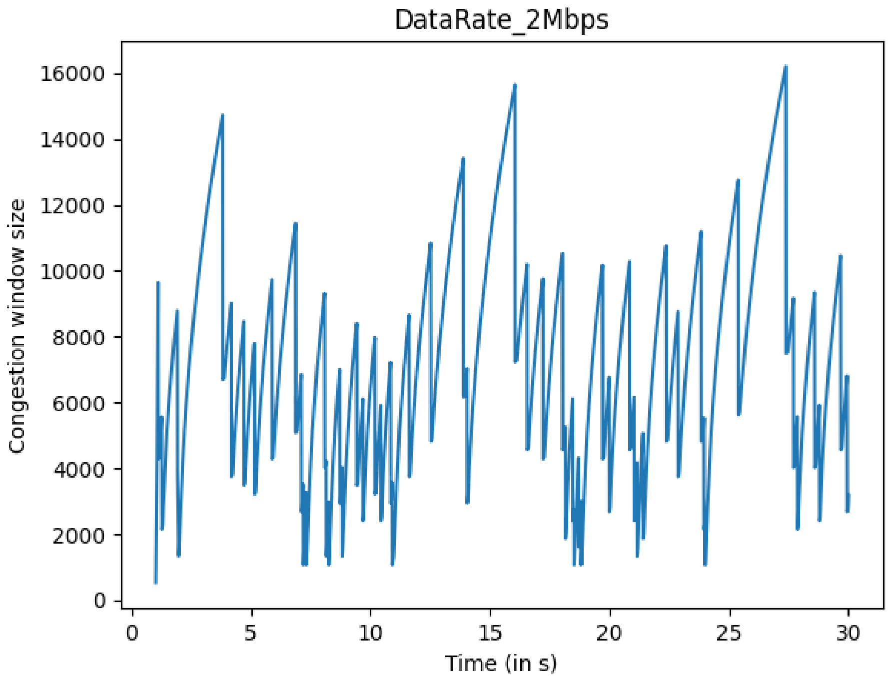

Figure 5: Data Rate has been fixed to 2Mbps

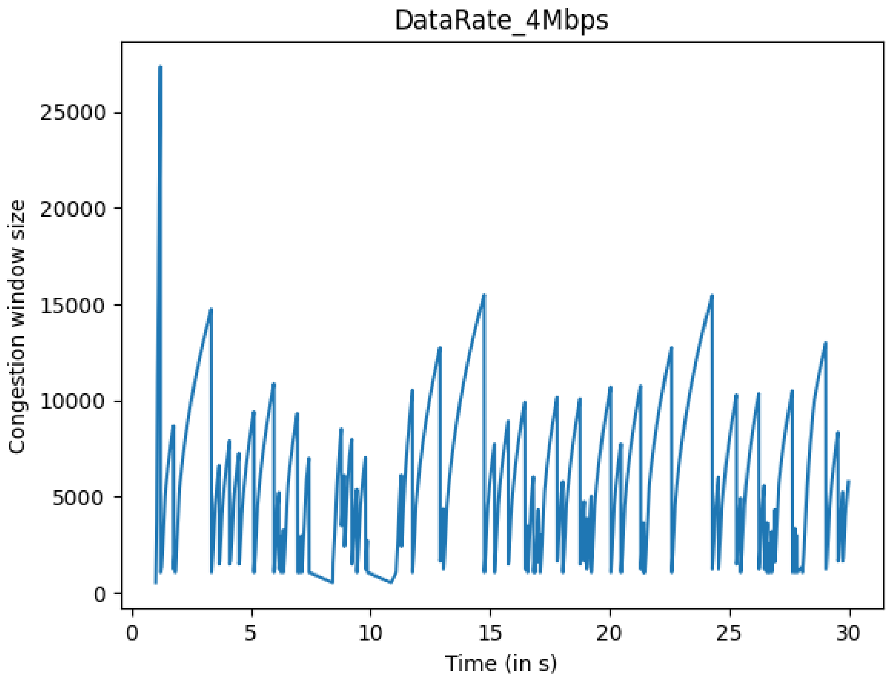

Figure 6: Data Rate has been fixed to 4Mbps

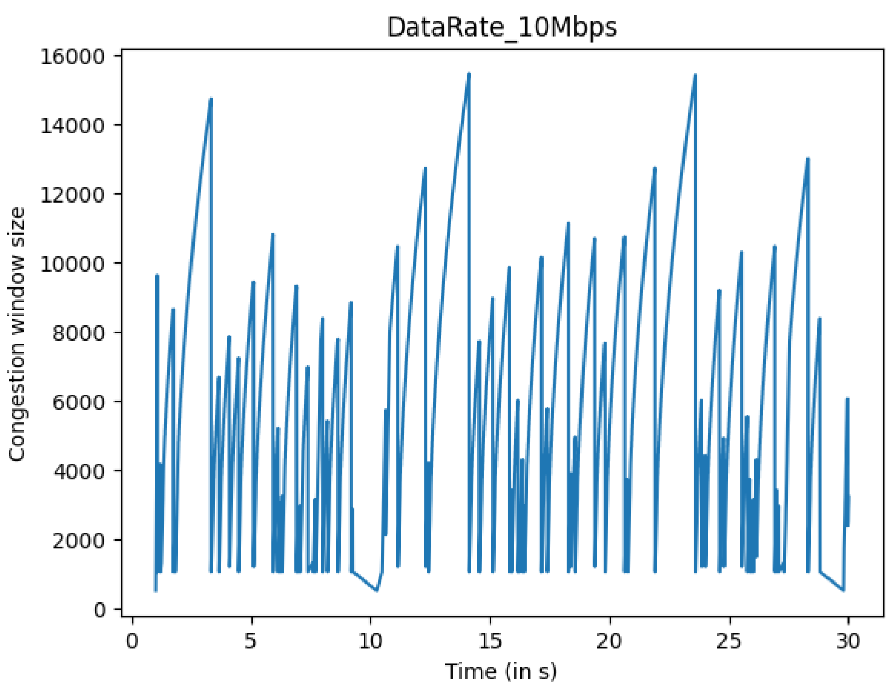

Figure 7: Data Rate has been fixed to 10Mbps

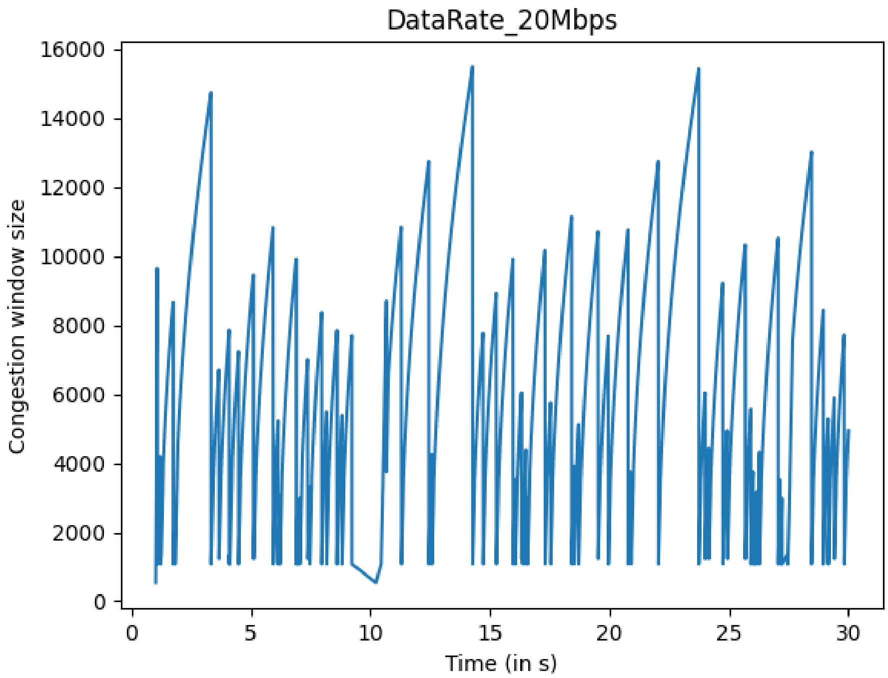

Figure 8: Data Rate has been fixed to 20Mbps

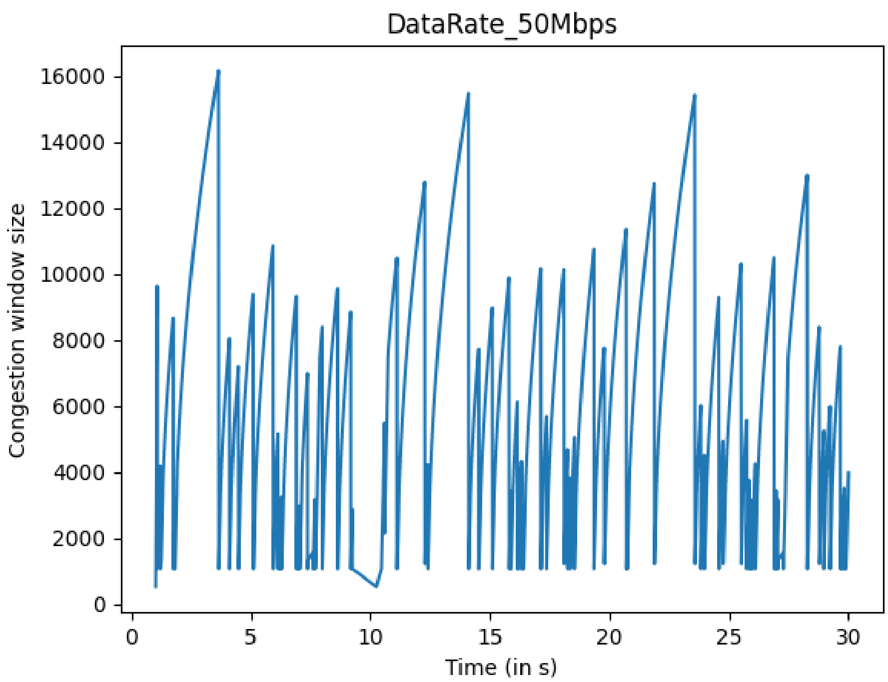

Figure 9: Data Rate has been fixed to 50Mbps

Larger variation in the congestion window for larger channel rate is due to the high frequence of packet drops. We can see that amount of variation in the congestion window per second increases. This will happen as it becomes more easy to transmit more number of packets over the link. Thus we can also see that the last three graphs are almost same due to above reason.

2.2 Plot the congestion window size vs time graph for the TCP connection at different Application Data Rates (0.5 Mbps, 1Mbps, 2Mbps, 4Mbps, 10 Mbps). Use Channel data rate between N1 and N2 as 6 Mbps. You need to create a plot for each Application data rate. Explain the trends that you observe.

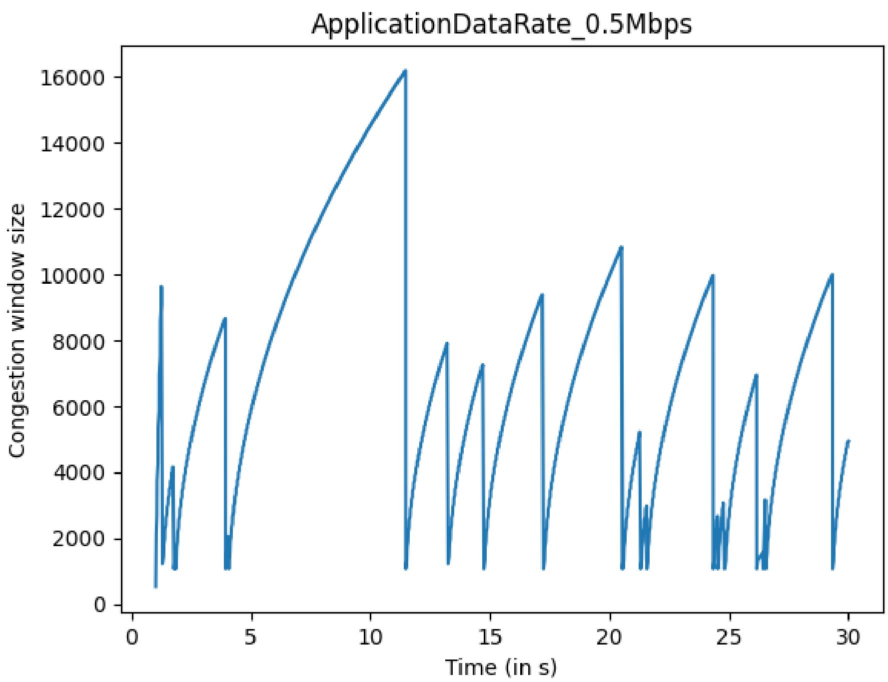

Figure 10: Application Data Rate has been fixed to 0.5Mbps

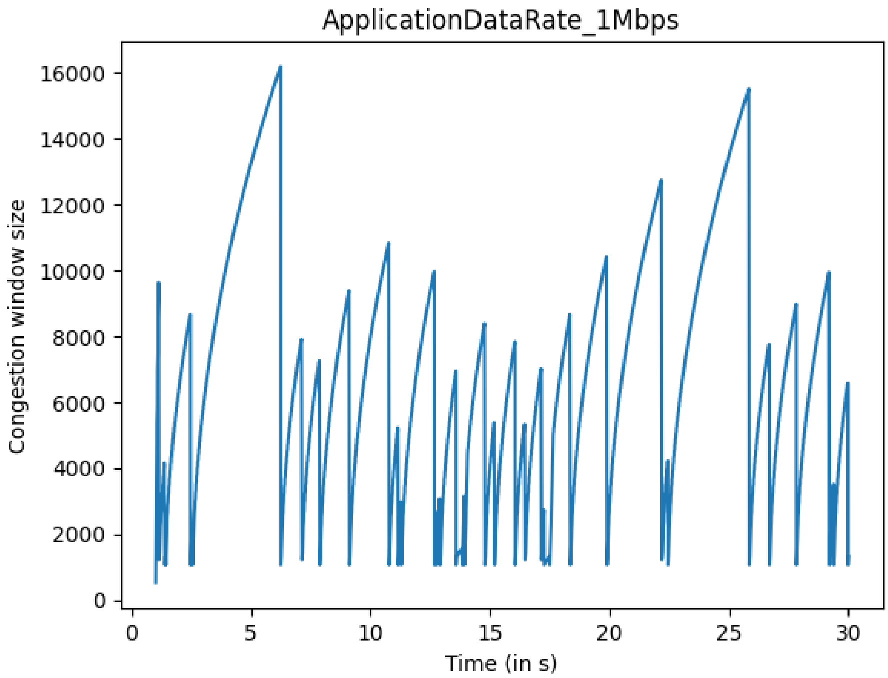

Figure 11: Application Data Rate has been fixed to 1Mbps

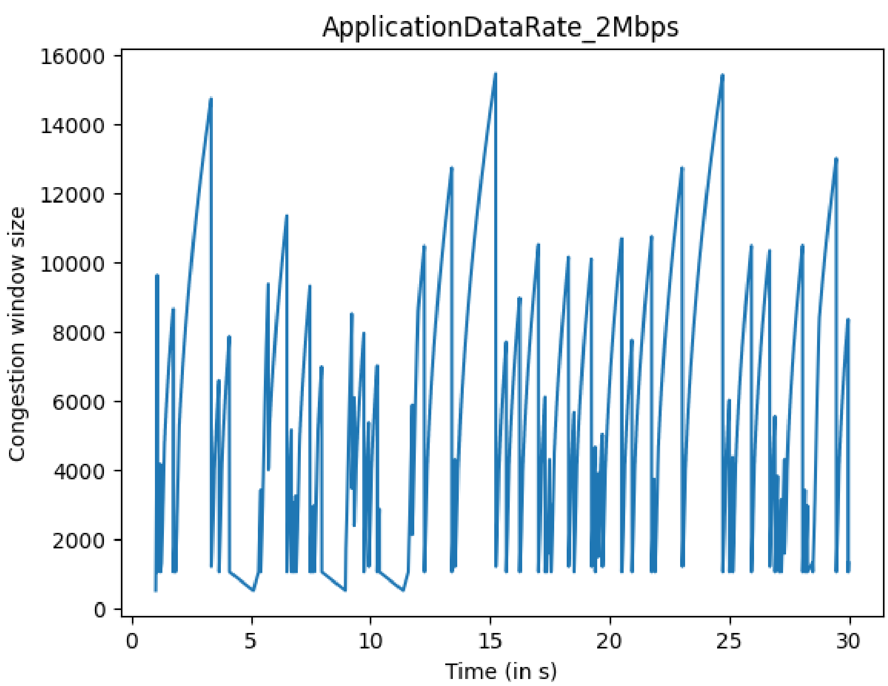

Figure 12: Application Data Rate has been fixed to 2Mbps

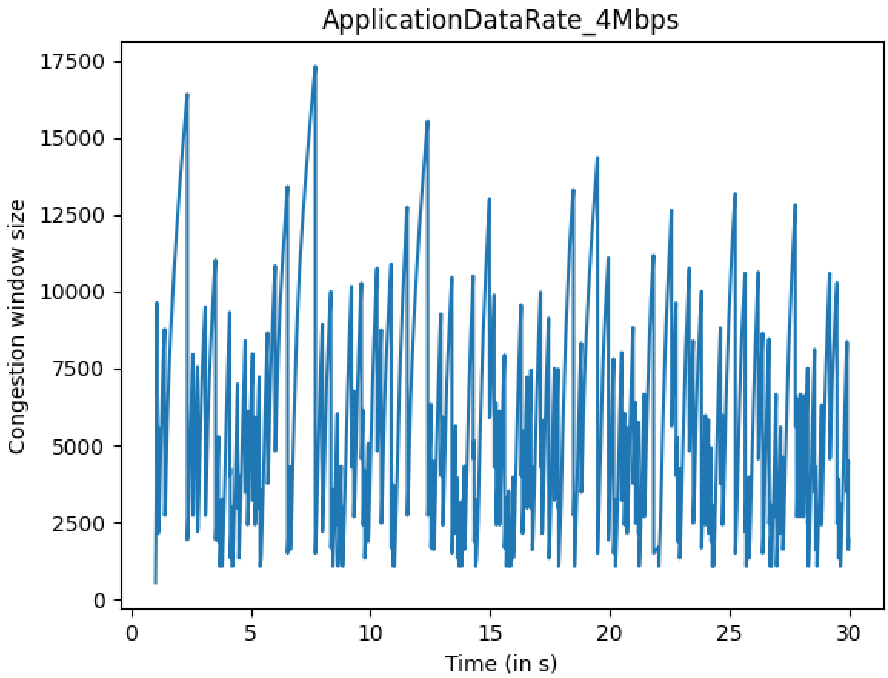

Figure 13: Application Data Rate has been fixed to 4Mbps

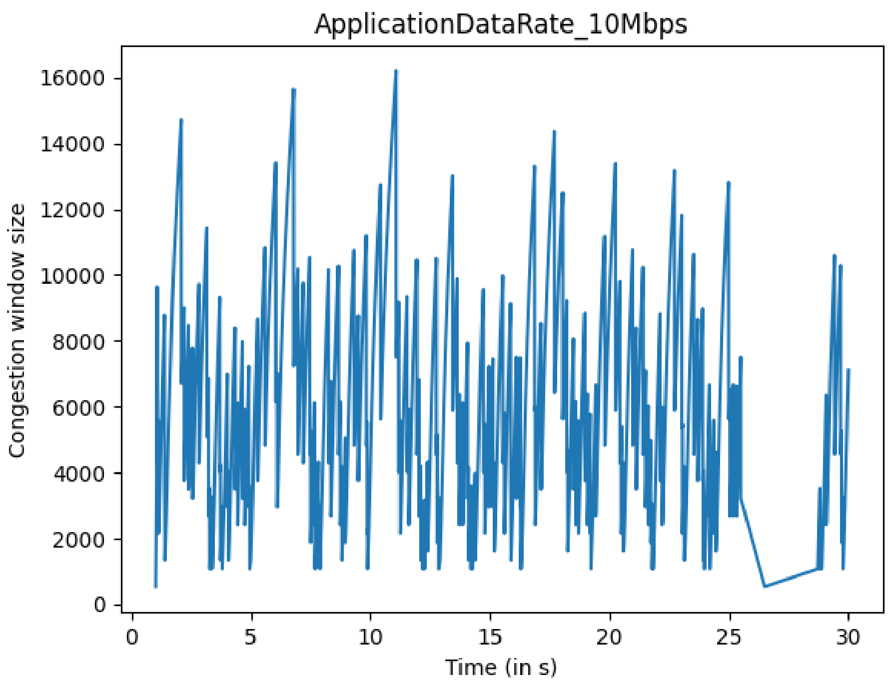

Figure 14: Application Data Rate has been fixed to 10Mbps

Drop rate increases for larger application rates. This can be seen from the following obseravtions. For smaller application rates, the congestion avoidance phase is for a longer time and as the the application rate increases we can see that it shift towards the slow start phase. For the larger application rate there is a region of lesser window size. Thus we can conculde that we need to keep a slower application rate in order to accomodate the the congestion in network.

3 Implementing TcpNewRenoCSE

The files have been included with the submission which contain the code for SlowStart and Congestion Avoidance. How to compile them is given in the README.md

3.1 Plot Congestion window size vs time (from t=1 to t=30 seconds)

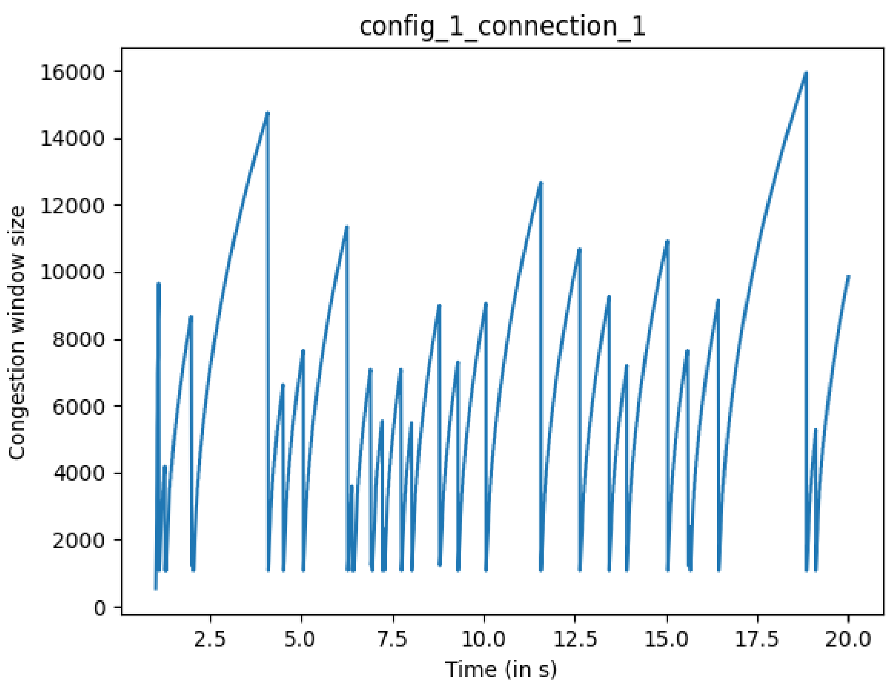

Figure 15: Plot for Configuration 1 and Connection 1

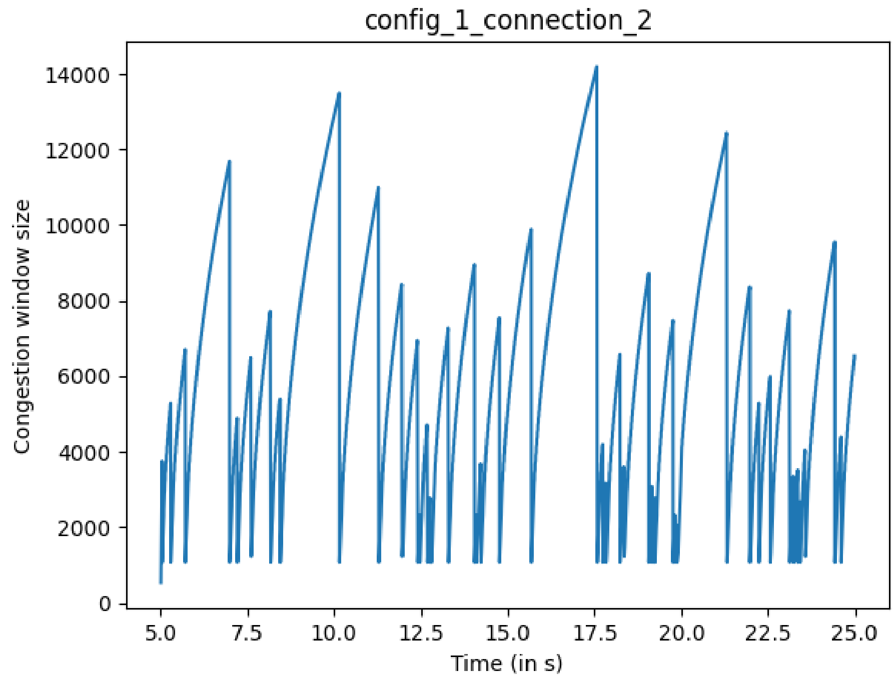

Figure 16: Plot for Configuration 1 and Connection 2

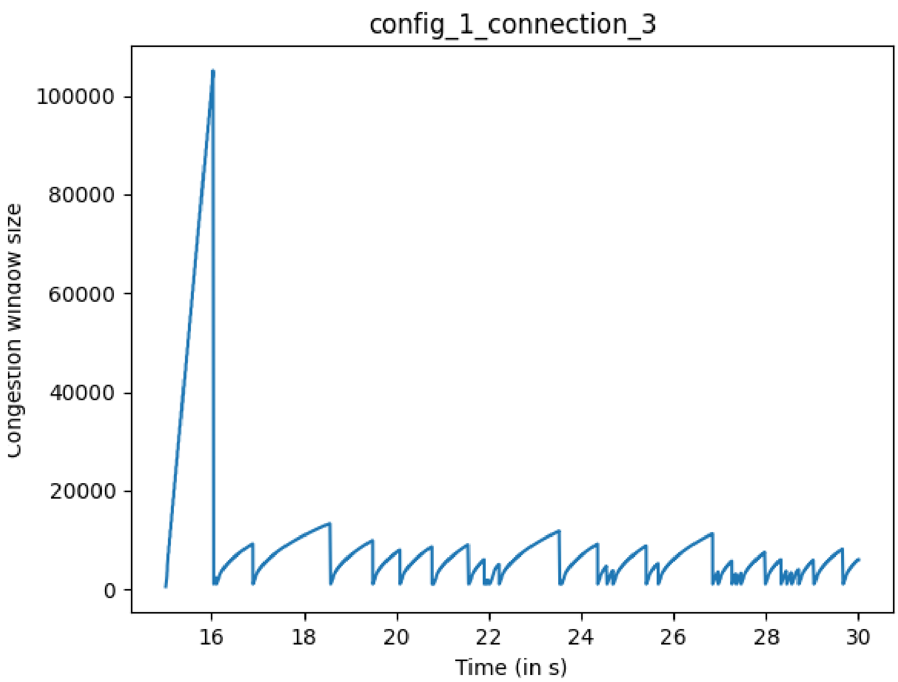

Figure 17: Plot for Configuration 1 and Connection 3

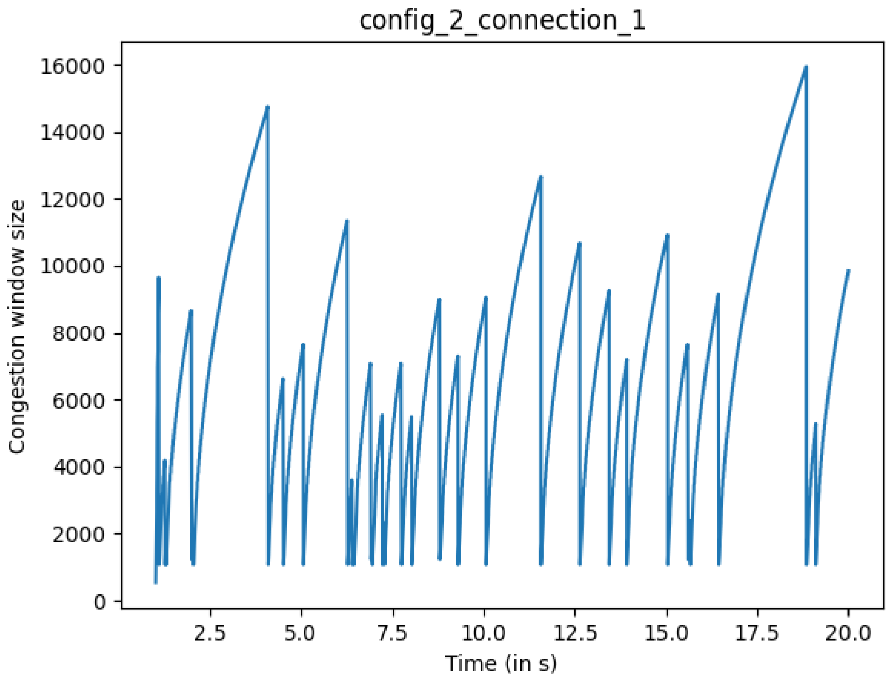

Figure 18: Plot for Configuration 2 and Connection 1

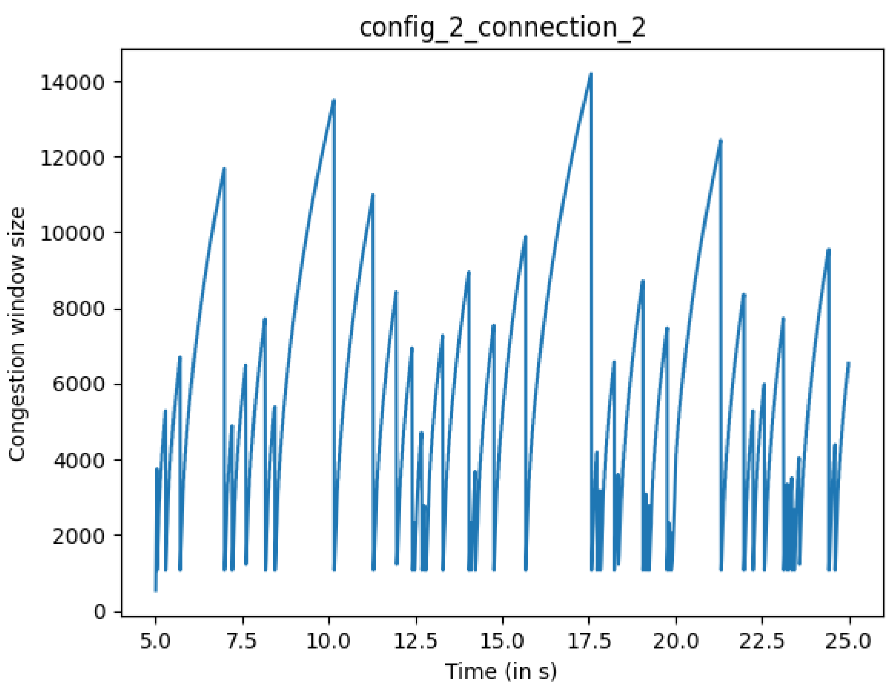

Figure 19: Plot for Configuration 2 and Connection 2

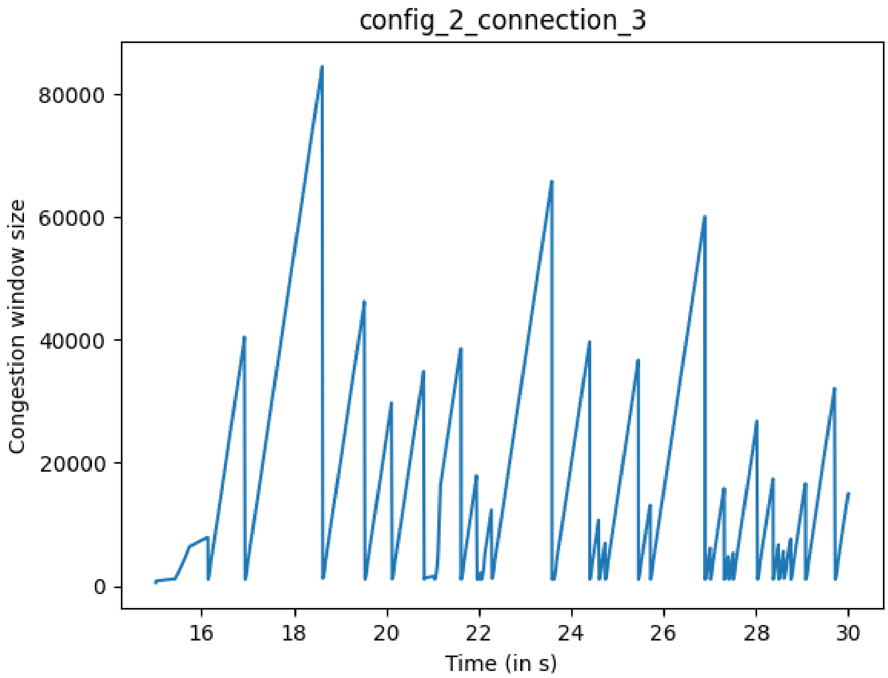

Figure 20: Plot for Configuration 2 and Connection 3

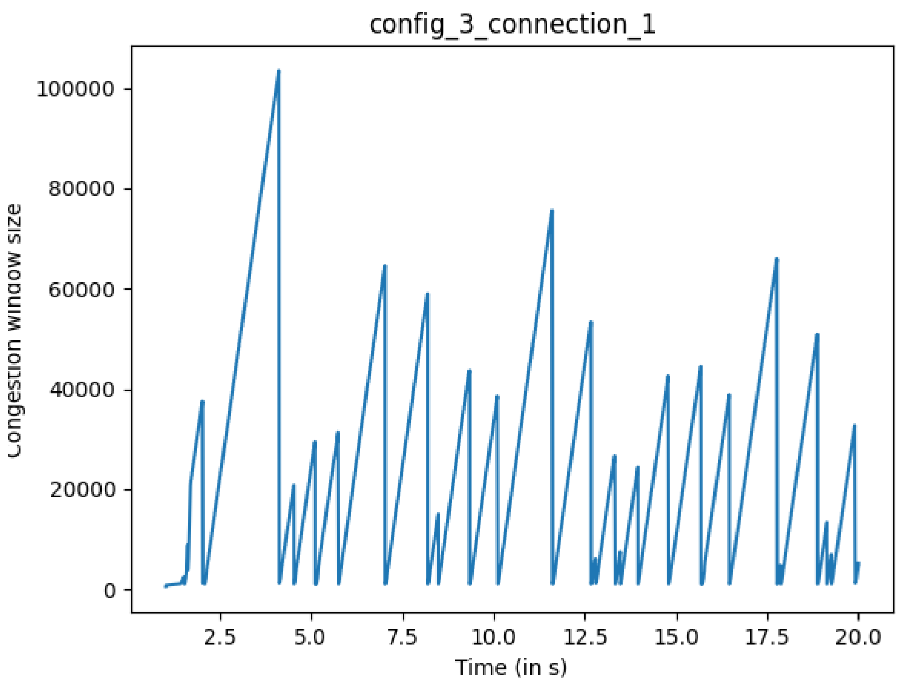

Figure 21: Plot for Configuration 3 and Connection 1

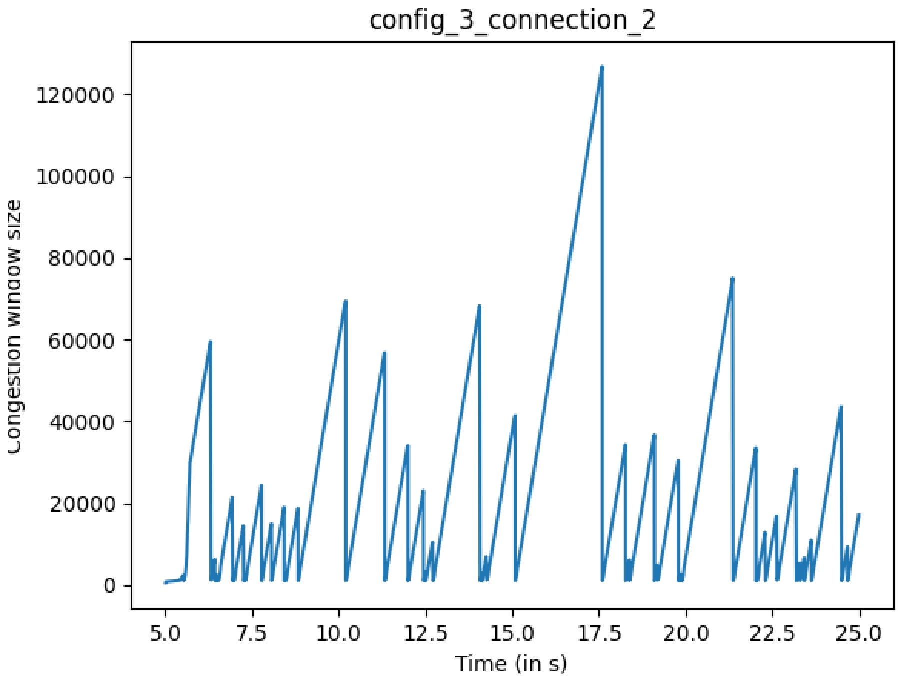

Figure 22: Plot for Configuration 3 and Connection 2

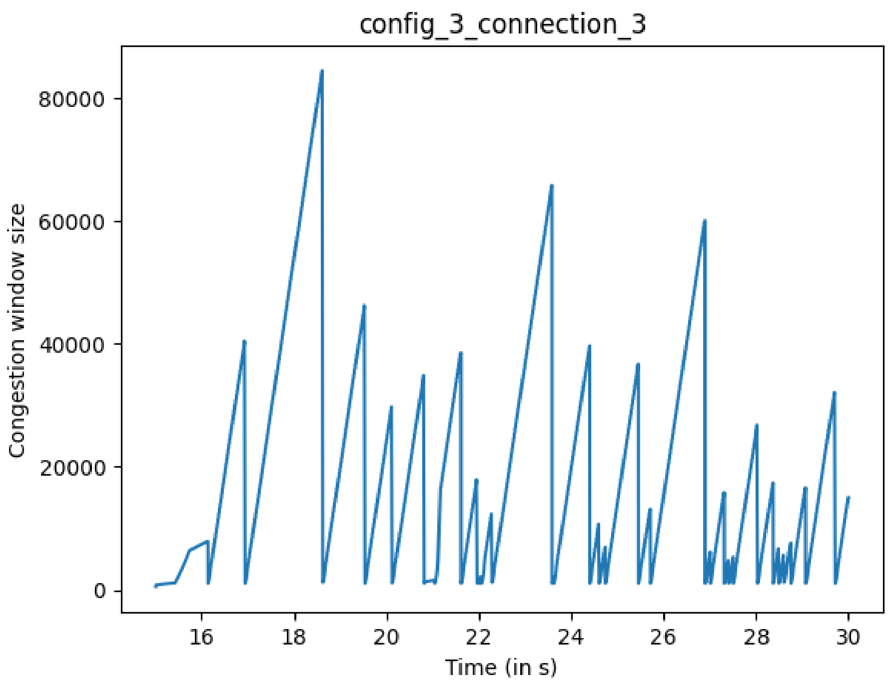

Figure 23: Plot for Configuration 3 and Connection 3

3.2 Report the total number of dropped packets in each scenario.

For this part I have made 2 errror models. In the updated assignment statement it is written that we have to just report only 3 values, one for each Configuration.

1. Configuration 1: 113
2. Configuration 2: 112
3. Configuration 3: 110

3.3 How does the congestion avoidance phase vary on the same sender when using TCPNewRenoCSE vs TCPNewReno? Explain the observed trends. How does it impact the entire network?

- We can notice that the average congestion window size is higher Whenever we use the TcpNewRenoCSE instead of TcpNewReno for a single connection.
- There happens a large number of packet drops on the first link for both of the models and thus we can see that both slow start and congestion avoidance states are achieved.
- In configuration 1 and 3, for the connection 1 and 2, the congestion window size achieved much higher values as comapred to the configuration 2.
- In cinfiguration 1 and 3, the proticol largely reamins in the congestion avoidance phase which leads to relatively large congestion window size.
- In case of configuration 3, for connection 3 the large increase in the congestion window size is due to the slow start phase. When it shifts to congestion avoidance there is a linear increase.

4 Directory Structure

As specified in the problem statement. I have submitted a plot.py, the instructions of how to use it have been given in the README.md. Apart from that the outputs have been stored in the following format, in the outputs folder of each question:

- dropped: It contains the information regarding which how many packets have been dropped for a particular part
- toplot: This contains the time vs the old congestion window size vs the new window size. My plotting script takes all the files present in this and plots them.
- plots: This contains the plots. Look at the titles of the plots for information about them.

Further instructions on how to run have been given in the **README.md** in a well detailed manner.