Assignment-3 Submission

Indian Institute of Technology Delhi

COL334: Computer Networks

Name: Deepanshu Entry Number: 2019CS50427 Group: 04

1 Part-1

1.1 Newreno

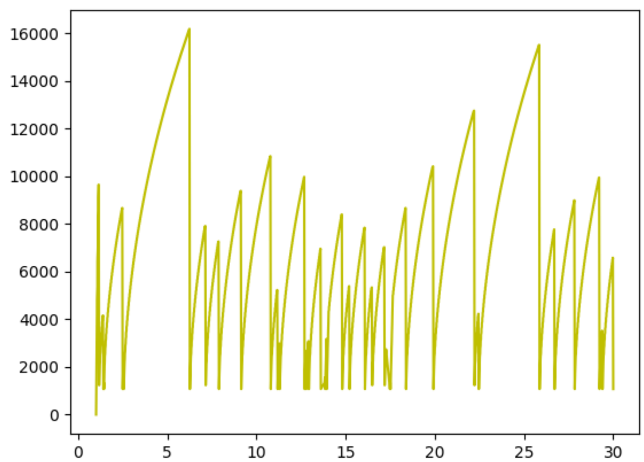

Figure 1: Congestion window size v/s time for Newreno protocol

Following are the key points corresponding to Newreno protocol:

1. NewReno algorithm introduces partial ACKs inside the Reno algorithm.
2. Following is the decision making in various cases:
   - Slow start: TCP increments cwnd by at most SMSS bytes.

- Congestion avoidance: cwnd is increased by 1 fully sized segment per RTT.
- Congestion event: Slow start threshold is halved.

3. **Number of packets dropped: 38**

1.2 Highspeed

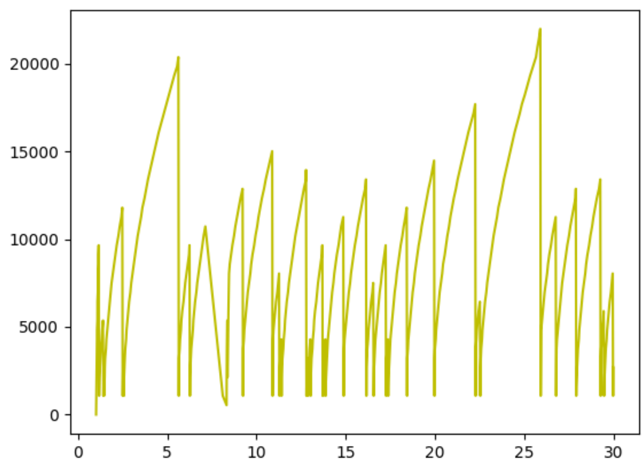

Figure 2: Congestion window size v/s time for HighSpeed protocol

Following are the key points corresponding to HighSpeed protocol:

1. HishSpeed is designed for use cases where we have **large congestion windows.**
2. Following is the decision making in various cases:
   - In probing phases where congestion window grows beyond a certain threshold, the the cWnd grow faster.
   - It thus becomes compatible with normal TCP environments with heavy congestion without any risk of congestion collapse.
   - Congestion event: Slow start threshold is halved.
3. **Number of packets dropped: 38**

1.3 Veno

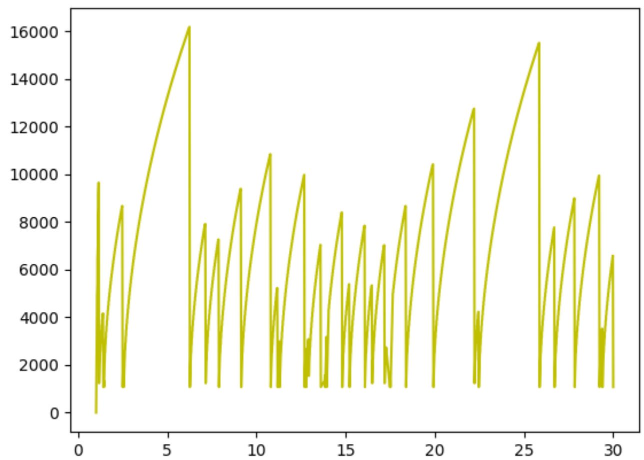

Figure 3: Congestion window size v/s time for Veno protocol

Following are the key points corresponding to Veno protocol:

1. Veno is designed for situations with random packet losses.
2. Following is the decision making in various cases:
   - It builds the algorithm over Reno algorithm.
   - It estimates the backlog at bottleneck queue to distinguish between congestive and non-congestive states.
   - Veno makes decision on cwnd modification based on the calculated N and its predefined threshold beta.
3. **Number of packets dropped: 38**

1.4 Vegas

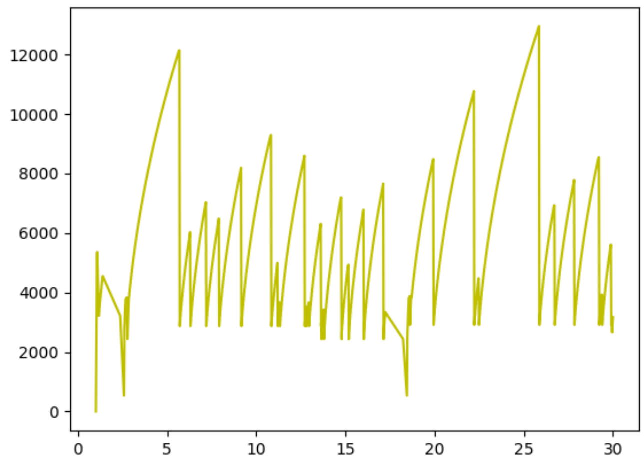

Figure 4: Congestion window size v/s time for Vegas protocol

Following are the key points corresponding to Vegas protocol:

1. Vegas is a proactive scheme that calculates the throughput at intervals to prevent packet losses.
2. Following is the decision making in various cases:
   - It maintains a small backlog at the bottleneck queue.
   - Vegas continuously samples the RTT and computes the actual throughput a connection achieves.
   - Here, we linearly increases/decreases its congestion window so that the diff value is used to take decision.
3. **Number of packets dropped: 39**

2 Part2

2.1 Varying the Channel Data Rates

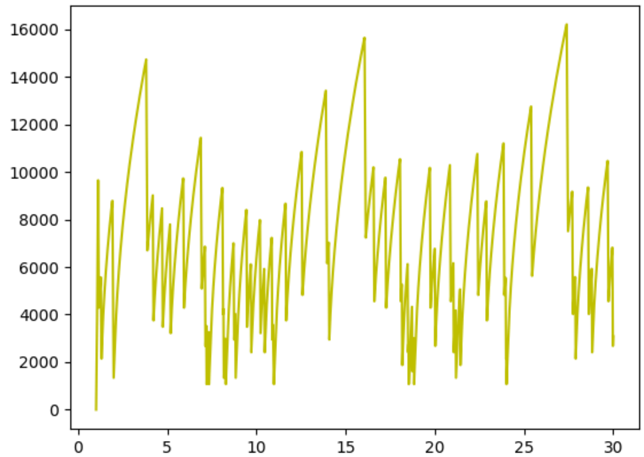

Figure 5: Congestion window size v/s time for Channel Data Rate = 2Mbps

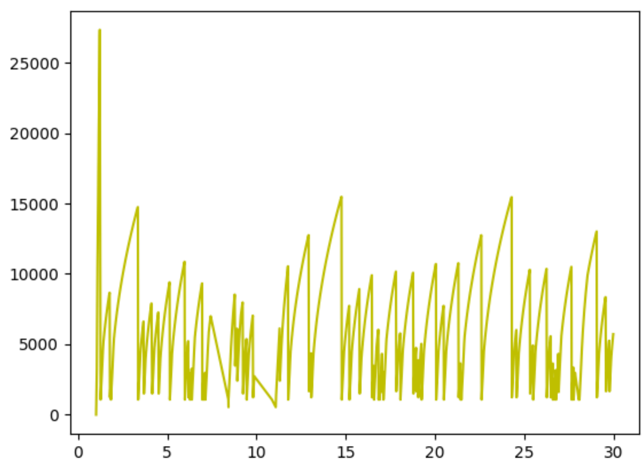

Figure 6: Congestion window size v/s time for Channel Data Rate = 4Mbps

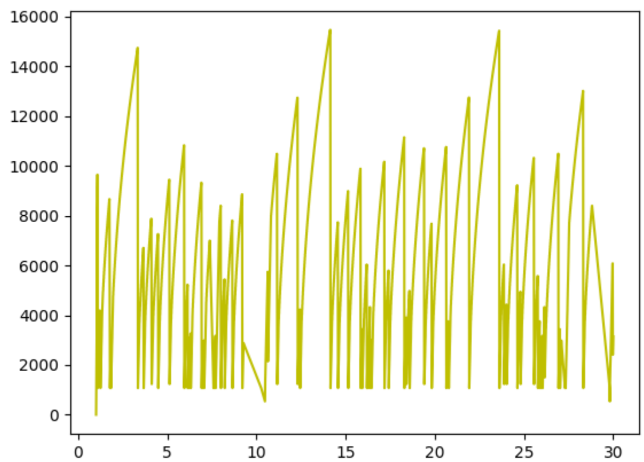

Figure 7: Congestion window size v/s time for Channel Data Rate = 10Mbps

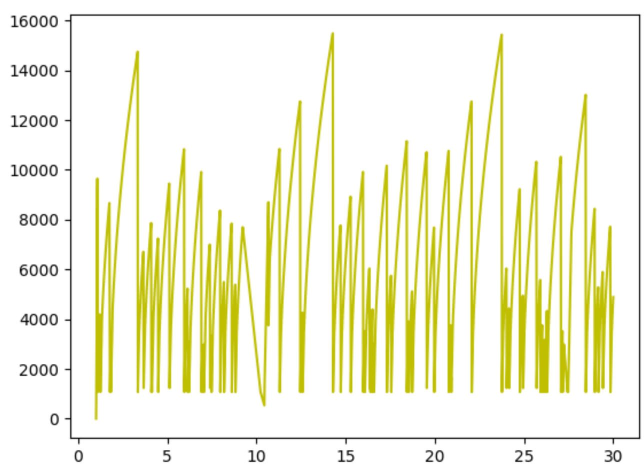

Figure 8: Congestion window size v/s time for Channel Data Rate = 20Mbps

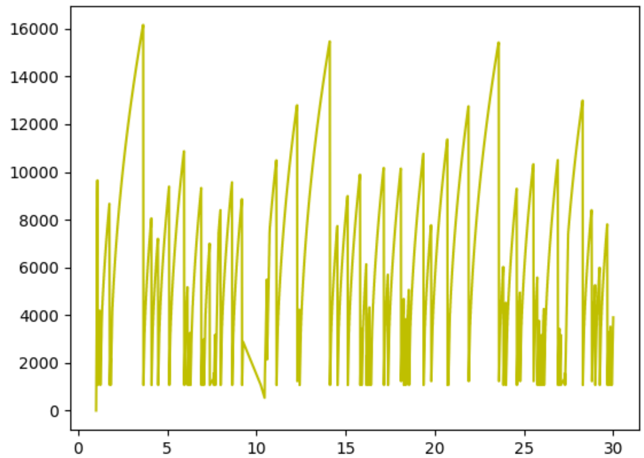

Figure 9: Congestion window size v/s time for Channel Data Rate = 50Mbps

Channel Data Rate	Packets Dropped
2Mbps	62
4Mbps	72
10Mbps	73
20Mbps	74
50Mbps	75

Table 1: Packets Dropped on Varying Channel Data Rate

Explanation for the trend above:

1.

2.2 Varying the Application Data Rates

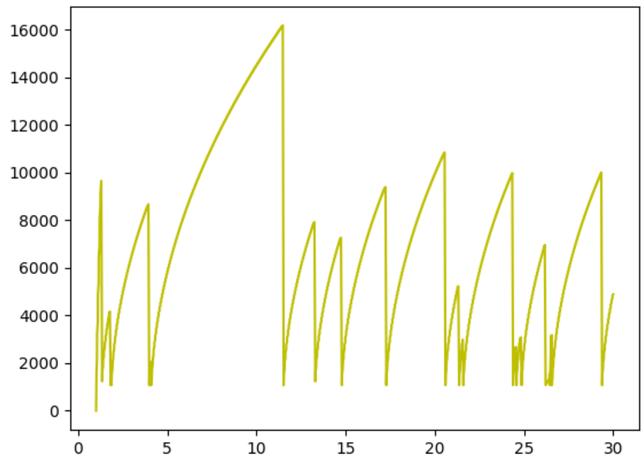

Figure 10: Congestion window size v/s time for Application Data Rate = 0.5Mbps

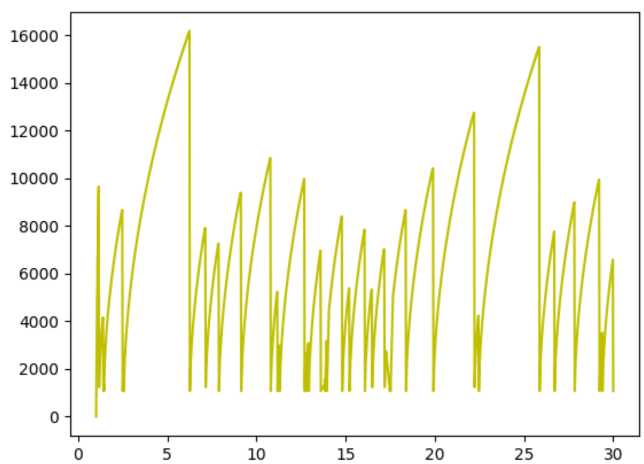

Figure 11: Congestion window size v/s time for Application Data Rate = 1Mbps

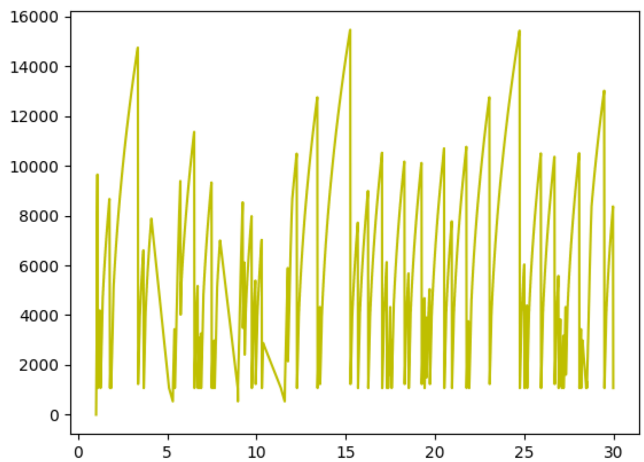

Figure 12: Congestion window size v/s time for Application Data Rate = 2Mbps

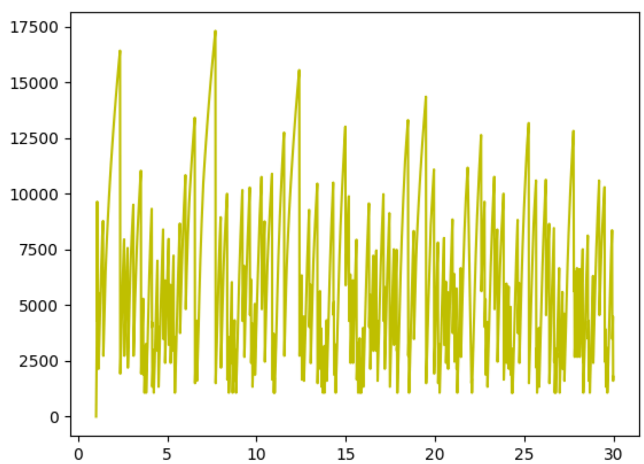

Figure 13: Congestion window size v/s time for Application Data Rate = 4Mbps

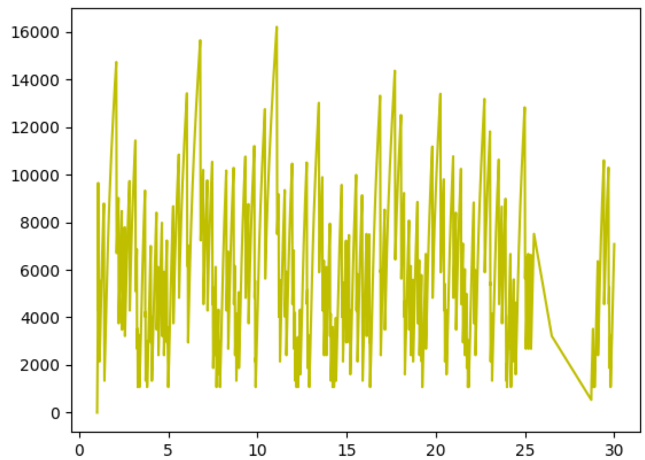

Figure 14: Congestion window size v/s time for Application Data Rate = 10Mbps

Application Data Rate	Packets Dropped
0.5Mbps	4
1Mbps	38
2Mbps	71
4Mbps	156
10Mbps	156

Table 2: Packets Dropped on Varying Application Data Rate

Explanation for the trend above:

1. Increasing the application data rate would lead to more packets being delivered and hence received at the application end.
2. This would lead to faster and frequent filling of congestion window.
3. This would ultimately lead to more packet losses.

3 Part3

3.1 Configuration-1

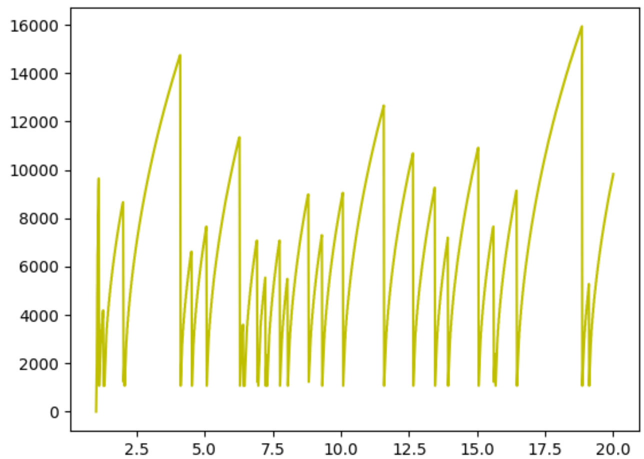

Figure 15: Congestion window size v/s time for node-1

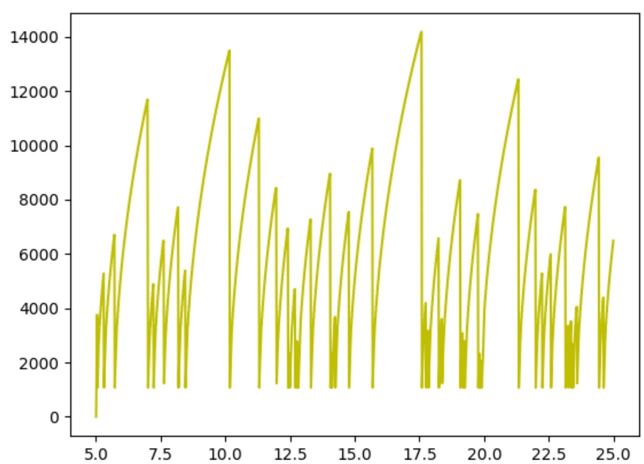

Figure 16: Congestion window size v/s time for node-2

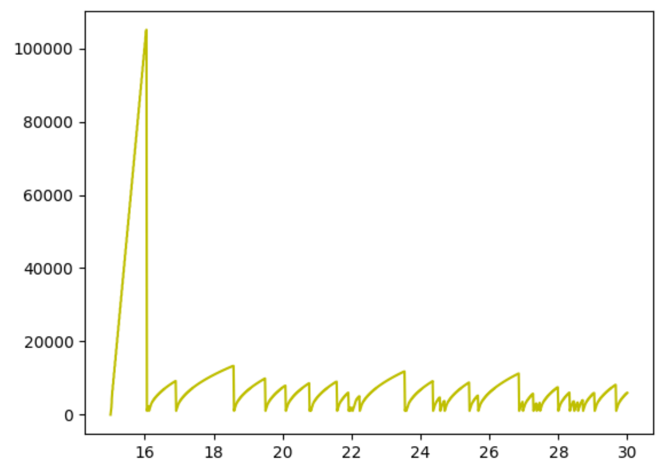

Figure 17: Congestion window size v/s time for node-3

3.2 Configuration-2

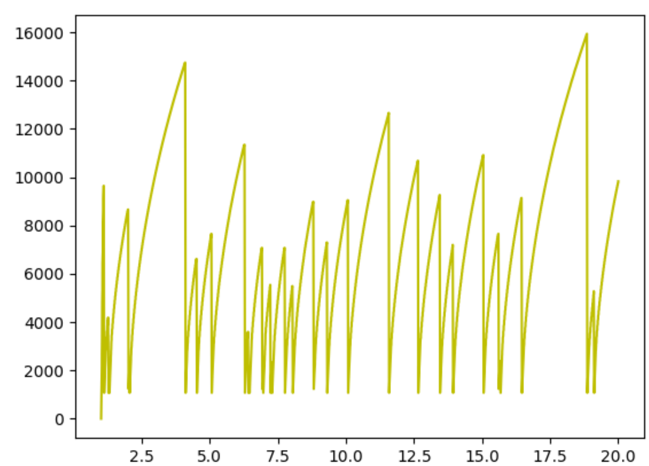

Figure 18: Congestion window size v/s time for node-1

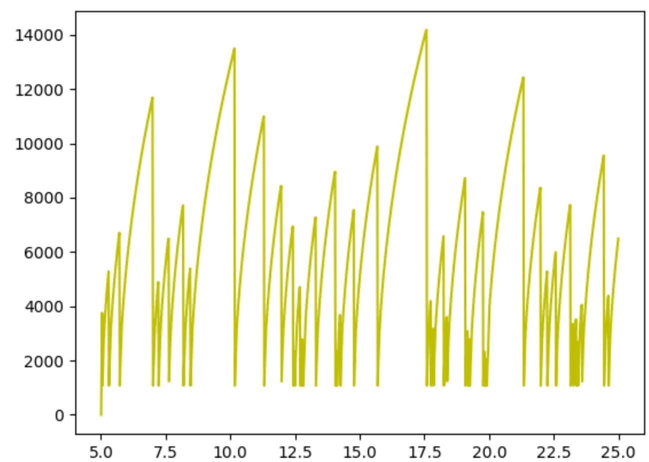

Figure 19: Congestion window size v/s time for node-2

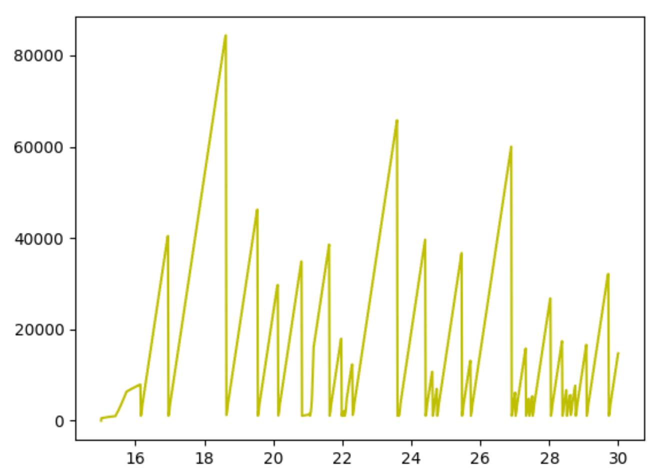

Figure 20: Congestion window size v/s time for node-3

3.3 Configuration-3

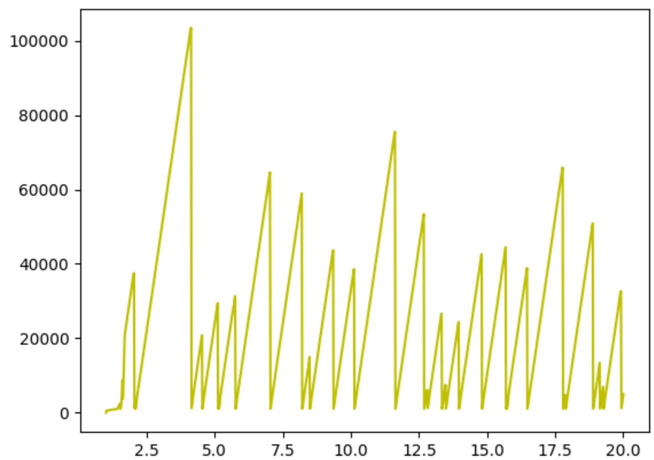

Figure 21: Congestion window size v/s time for node-1

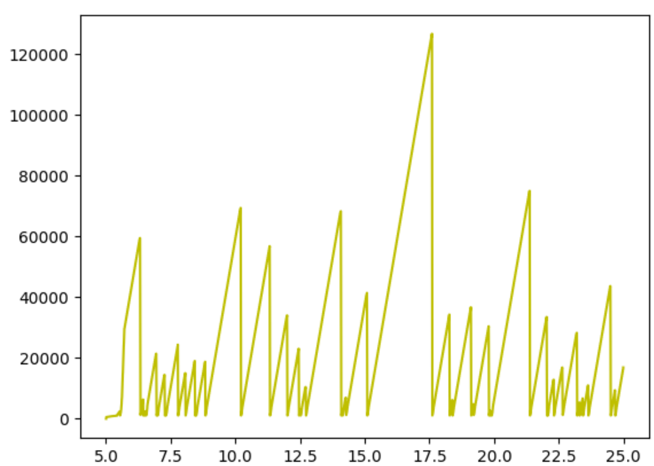

Figure 22: Congestion window size v/s time for node-2

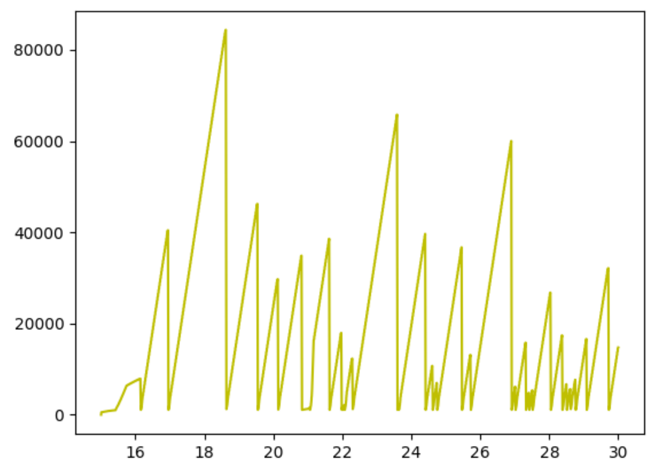

Figure 23: Congestion window size v/s time for node-3

3.4 Analysis and observations

Configuration	Packets Dropped
Configuration1	113
Configuration2	112
Configuration3	110

Table 3: Packets Dropped for different configurations

1. From the graph, the exponential nature of congestion window size can be observed while using TcpNewRenoCSE.
2. This growth based strategy makes the channel bursty in nature which is useful in numerous practical applications.
3. Exponential nature leads to lower packet losses.
4. Since TcpNewRenoCSE extends TcpNewReno, thus TcpNewRenoCSE can show some inherent behaviour similar to TcpNewReno under normal circumstances.

4 Usage of various files

Following is the role/usage of various files submitted in the assignment:

1. Q1 folder contains 4 files having .cc extension. First_1.cc - First_4.cc are used to run NewReno, HighSpeed, Veno and Vegas respectively.

2. Second_1.cc - Second_5.cc are used to run the NewReno code by varying Channel Data Rate.

3. Second_6.cc - Second_10.cc are used to run the NewReno code by varying Application Data Rate.

4. Second_6.cc - Second_10.cc are used to implement the three configurations mentioned in the assignment.

5. Congestion folder inside Q3 folder contains the TcpNewRenoCSE.cc and TcpNewRenoCSE.h files that contains the definition of TcpNewRenoCSE protocol that we customized in accordance to the definition provided in the assignment.

6. plotcode.py is used to plot the graphs using matplotlib library in python.

5 References

- https://www.nsnam.org/docs/models/html/tcp.html
- https://www.sciencedirect.com/topics/computer-science/packet-delivery-ratio