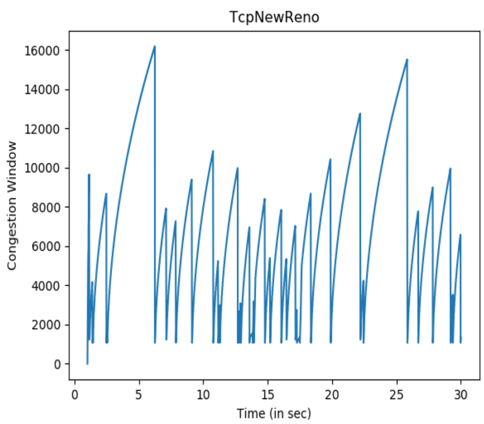
COL334 Assignment3

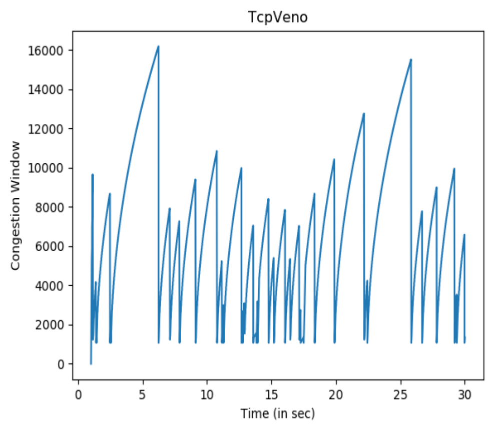
Aniket Gupta
Entry No. - 2019CS10327

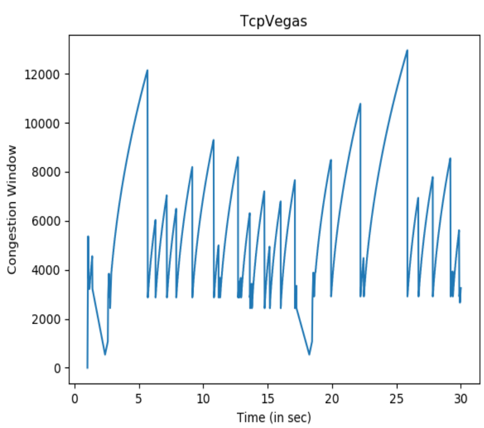
26 October 2021

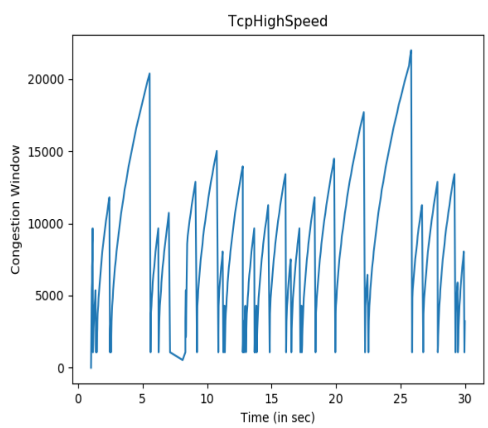
1 Part 1

1.1 Plots of Congestion Window Size vs Time for Different Protocols









1.2 Packets Dropped

- **NewReno**
  No. of packets dropped: 38
- **Highspeed**
  No. of packets dropped: 38
- **Veno**
  No. of packets dropped: 38
- **Vegas**
  No. of packets dropped: 39

  The no. of packets dropped is almost same in all the cases (Vegas has one more). One possible reason is the use of same error rate in the error models used in all the protocols. The no. of packets loss in NewReno, Veno and HighSpeed can be determined by the no. of times the cWnd decreases in the plot. The no. of decreases in cWnd in Vegas may be different from no. of packets droppped as Vegas uses delay based approach to increase/decrease cWnd.

1.3 Observations and Explanations

- **NewReno**
  In NewReno, congestion window increases by 1 MSS for every acknowledgement packet received during the slow start period. During the congestion avoidance phase, it increases by roughly 1 MSS per RTT time. Slow start ends when congestion window exceeds slow start threshold. In the plot of TCP NewReno shown above, the plot is approximately linear at the higher values of Cwnd (when it is in congestion avoidance phase) where cwnd increases by 1 MSS per RTT.
- **Highspeed**
  In Highspeed the cWnd grows faster during the starting period and thus it helps to quickly recover from a packet loss. It is clear from the plot of TcpHighSpeed show above. In TcpHighSpeed, the cWnd has increased to 20000 in approximately 5 secs which is greater than that of other protocols in same amount of time. In this protocol, cWnd increases as:

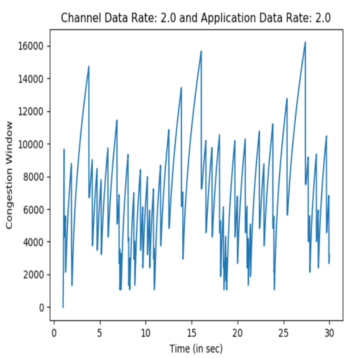
  $$cWnd{+}= \frac{a(cwnd)}{cwnd}$$

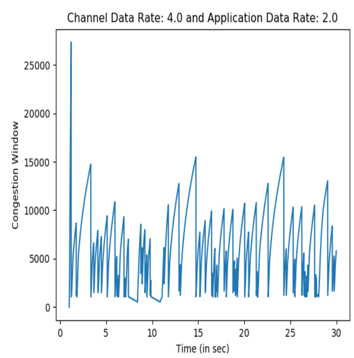
  where the function a() is dependent on current cWnd and is observed from a lookup table.
- **Vegas**
  Tcp Vegas is a delay-based congestion control algorithm. Vegas continuously compares the actual throughput with the expected throughput. The difference between these two is treated as an indicator of the number of packets that are still queued at the bottleneck. It linearly increases and decreases the cWnd such that it lies between a some given thresholds.
  It can be observed from the plot above that the cWnd is smaller in TcpVegas as compared to some other protocols as the other protocols keep increasing cWnd until a packet loss happens but TcpVegas increases/decreases based on delay.
- **Veno**
  Tcp Veno is very similar to Reno but it also uses method similar to Vegas by using the difference between actual and expected throughputs to distinguish between congestive and

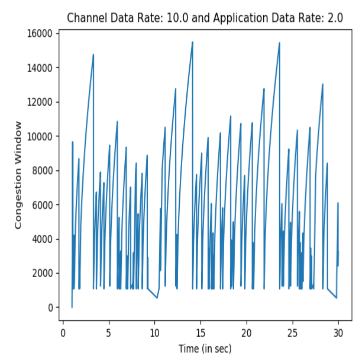
non-congestive states. In the additive increase method, it increases the cWnd by $\frac{1}{cwnd}$ for every acknowledged packed till the bandwidth is not fully utilised. After the full bandwidth utilisation, it increases its cWnd on alternate acknowledgement packets received.
The possible reason for the similarity in plots of TcpNewReno and TcpVeno is that the full bandwidth is not utilised at any time at the given data rates in TcpVeno protocol and thus the increase/decrease in cWnd is similar in both these protocols.

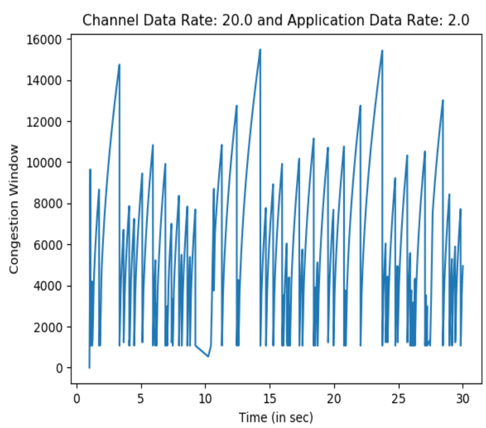
2 Part 2

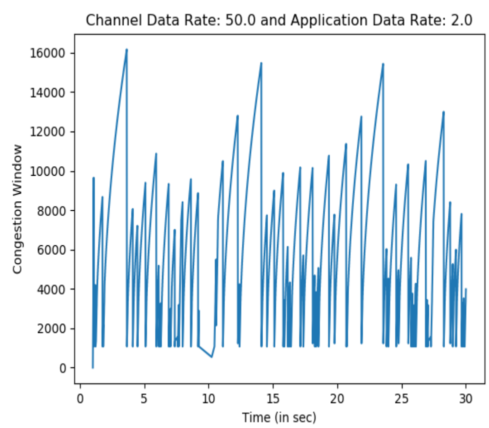
2.1 Congestion Window Size vs Time for different Channel Data Rates









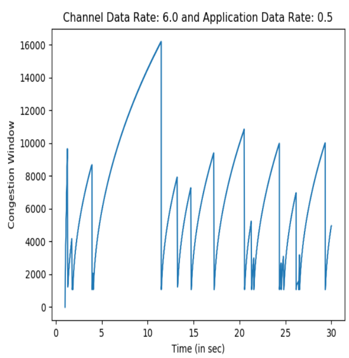


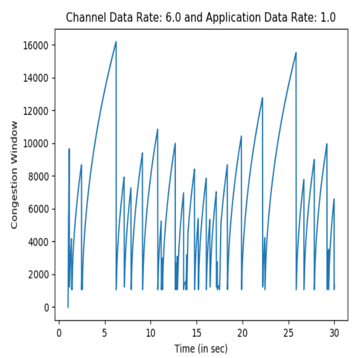
In this part, the application data rate is kept fixed at 2 Mbps and the channel data rate is varied. The plots of the congestion window vs time are shown above and the packets dropped are mentioned in the below table.

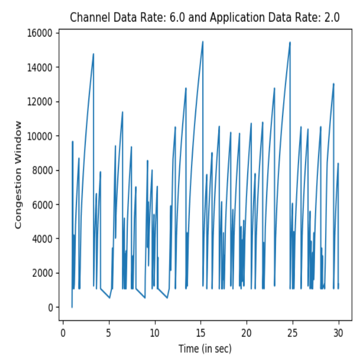
Channel Data Rate	No. of Packets Dropped
2	62
4	72
10	73
10	74
59	75

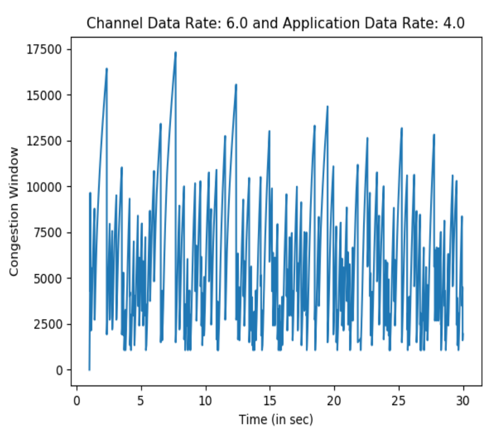
It can be observed that the change in congestion window per unit time increases on increasing the channel data rate. The possible reason is that it becomes possible to transfer more number of packets as we increase the channel rate. Also, this increase in cWnd variation is due to increase in no. of packets dropped.

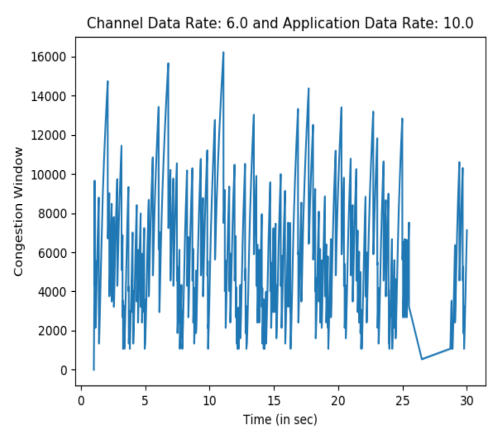
2.2 Congestion Window Size vs Time for different Application Data Rates











In this part, the channel data rate is kept fixed at 6 Mbps and the application data rate is changed. The plots of the congestion window vs time are shown above and the packets dropped are mentioned in the below table.

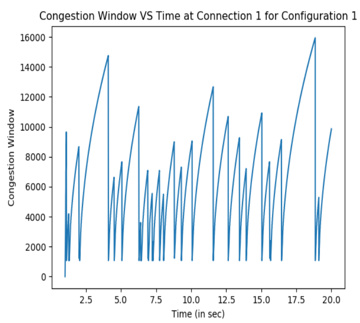
Application Data Rate	No. of Packets Dropped
0.5	22
1	38
2	71
4	156
10	156

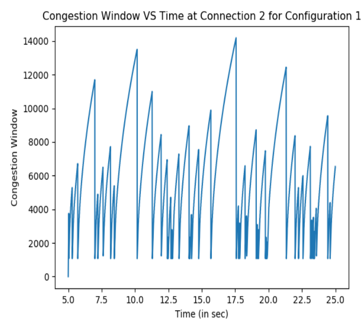
It can be observed that the no. of packets dropped increases with the increase in application data rate. This is in line with what is expected because for a fixed channel rate, increasing the application data rate increases the congestion in the network and thus the no. of packets dropped increases.

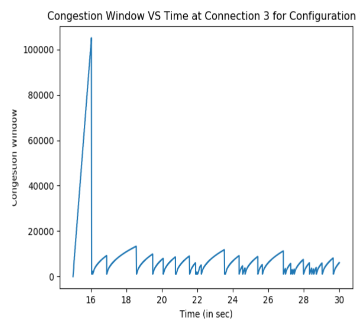
3 Part 3

3.1 Plots of Congestion Window Size vs Time for Different Configurations

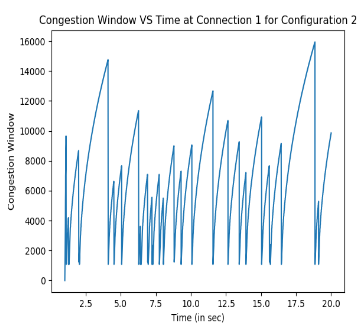
3.1.1 All Connections use TcpNewReno

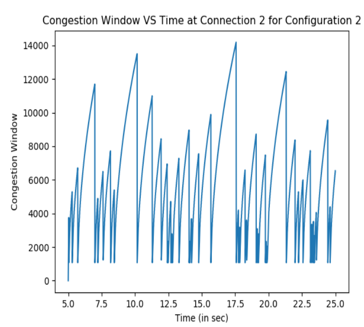


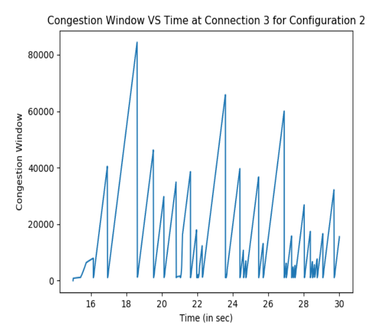




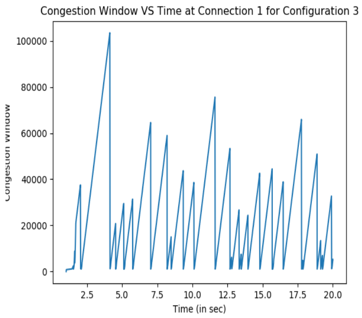
3.1.2 Connection 3 uses TcpNewRenoCSE and other use TcpNewReno

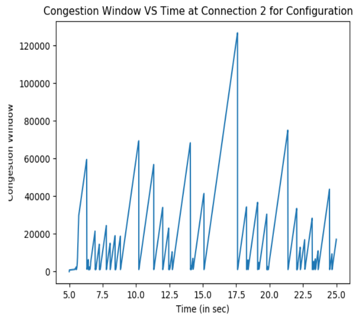


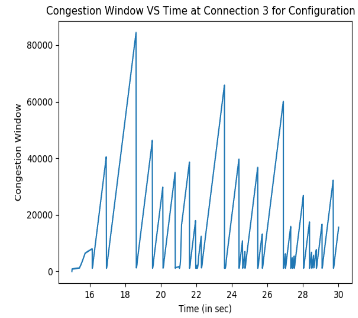




3.1.3 All Connections use TcpNewRenoCSE







3.2 Packets Dropped

Configuration	Total no. of Packets Dropped
1	113
2	112
3	110

3.3 Comparison of congestion avoidance phases of these two protocols and other observations

- It can be observed that the average cWnd is much larger in TcpNewRenoCSE as compared to that in TcpNewReno for a particular node.
- In TcpNewReno, the congestion window is increased by 1 MSS per RTT time whereas in TcpNewRenoCSE, the congestion window in increased by 0.5 MSS per acknowledged packets. Thus, the cWnd in TcpNewRenoCSE increases by higher amount in same amount of given time.
- Because of the large no. of packets dropped on the first link, both slow start and congestion avoidance phase is observed in both the models.
- In case of connection 3 in configuration 1, the protocol largely remains in congestion avoidance phase which leads to lower congestion window.
- In case of connection 3 in configuration 3, the slow start phase increases the cWnd largely and then it changes to congestion avoidance phase.