Computer Networks
Assignment-3

Himanshu Gaurav Singh
2019CS10358

October 26, 2021

Contents

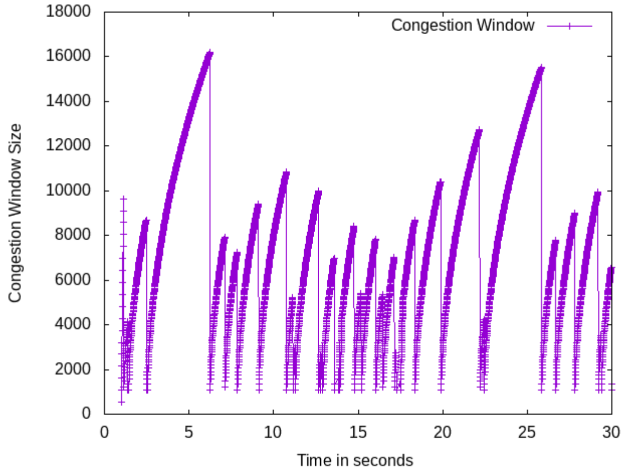
1 Part-1

The code for this part can be run by:
`./waf --run "<file-location> --protocol=(NewReno/Vegas/Veno/HighSpeed)"`
For drawing the plots:
`gnuplot -e "inputfile='<inputfile>' ; outputfile='<outputfile>' plotter.plg`

1.a TCPNewReno

1.a.1 Plot

The following is the plot for `TcpNewReno` variant.



1.a.2 Lost packets

Total dropped packets is 38.

1.a.3 Observation and Description

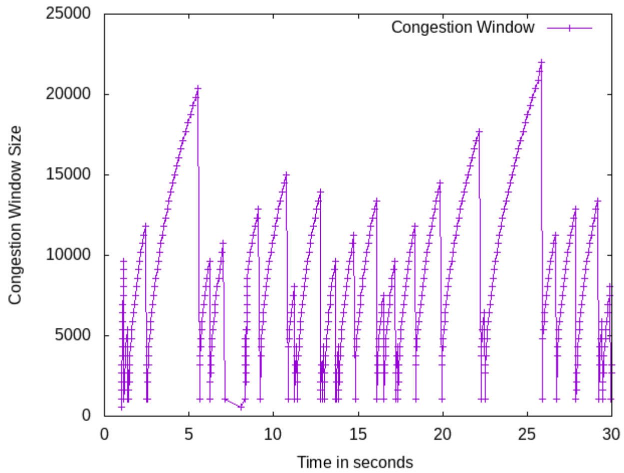
1. The `TcpNewReno` protocol is the default currently most popular variant of TCP. It improves upon the `TcpReno` protocol that uses the standard slow start and congestion avoidance protocol.

2. While in usual TCP protocols, the `cwnd` increases by one MSS on receiving an `ACK`, the NewReno protocol increases it by $min(N, MSS)$ where N is amount of bytes acknowledged by this particular ACK that was unacknowledged before.

1.b HighSpeed

1.b.1 Plot

The following is the plot for `HighSpeed` variant.



1.b.2 Lost packets

Total dropped packets is 38.

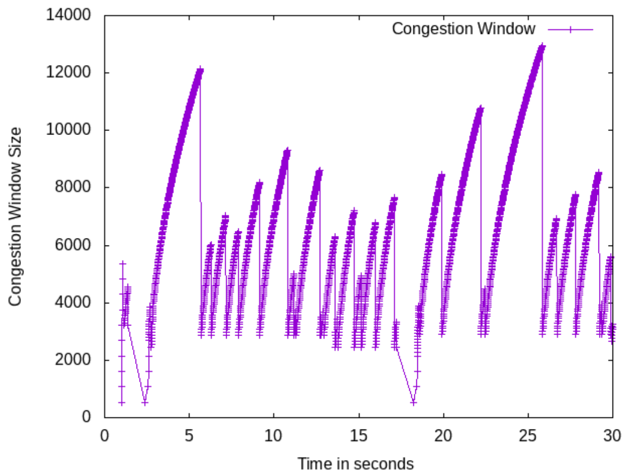
1.b.3 Observation and Description

1. Observe that `HighSpeed` leads to a faster and larger increase in `cwnd` than `TcpNewReno` and also recovers from losses more efficiently. Thr peak values of `cwnd` are also higher.
2. This is because of the "HighSpeed" principle used in the algorithm. According to it, in the initial stage of the congestion avoidance phase(the probing phase), each `ACK` received leads to larger increase in `cwnd` than afterwards.

1.c Vegas

1.c.1 Plot

The following is the plot for `Vegas` variant.



1.c.2 Lost packets

Total dropped packets is 39.

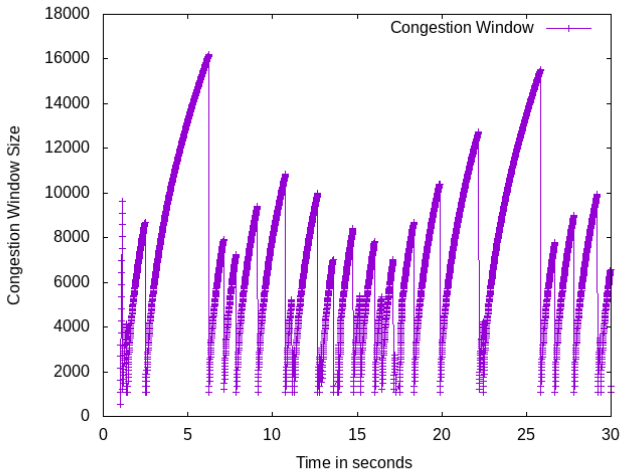
1.c.3 Observation and Description

1. Observe that the average throughput in `Vegas` is better than NewReno although the maximum `cwnd` size is smaller for it.
2. `Vegas` uses change in `RTT` as the measure of congestion in the route. It maintains an expected round trip time variable and compares it with the observed RTT to estimate bottlenecks in the route.
3. Also the congestion window sizes are increased/decreased within an upper and lower threshold and the thresholds are modified with time. The existence of a higher lower bound can be seen from the plot as well.

1.d Veno

1.d.1 Plot

The following is the plot for `Veno` variant.



1.d.2 Lost packets

Total dropped packets is 38.

1.d.3 Observation and Description

1. Observe that there is no noticeable difference between `Veno` and `NewReno` output plots.
2. `Veno` attempts to improve `NewReno` by trying to estimate the congestion in the bottleneck buffer in the complete route rather than just relying on packet loss event or multiple ACKs.
3. This is done by maintaining the number and distribution of the number of sent and unacknowledged packets. However, this improvement does not lead to sufficient gain in our setup.

1.e Comparing packet losses

- From comparing the number of packets dropped, we see the total packets dropped are nearly the same which is reasonable since packet loss is primarily limited by the network architecture and the buffer capacities.

2 Part-2:Effect of Channel Data rate and Application Data Rate

The code for this part can be run by:

```
./waf --run "<file-location> --ChDR=<Channel-Data-Rate> --AppDR=<Appn-Data-Rate>"
```

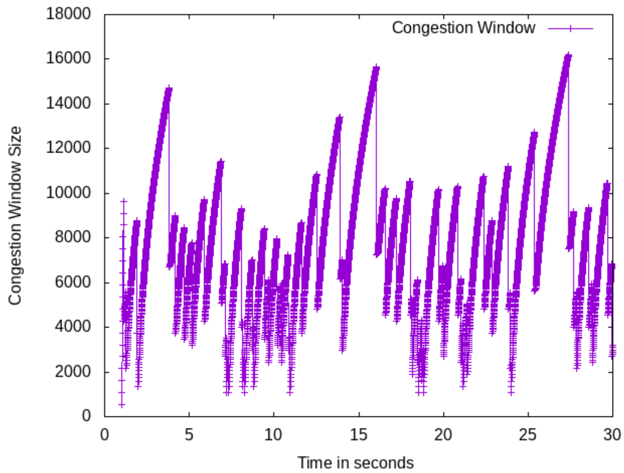
For drawing the plots:

```
gnuplot -e "inputfile='<inputfile>' ; outputfile='<outputfile>' plotter.plg
```

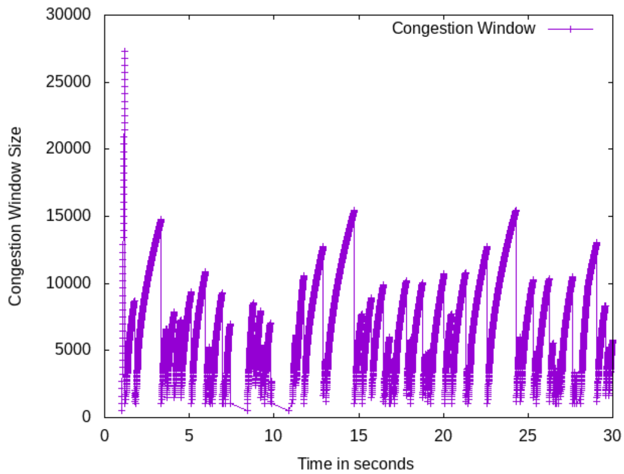
2.a (a)

2.a.1 Plots

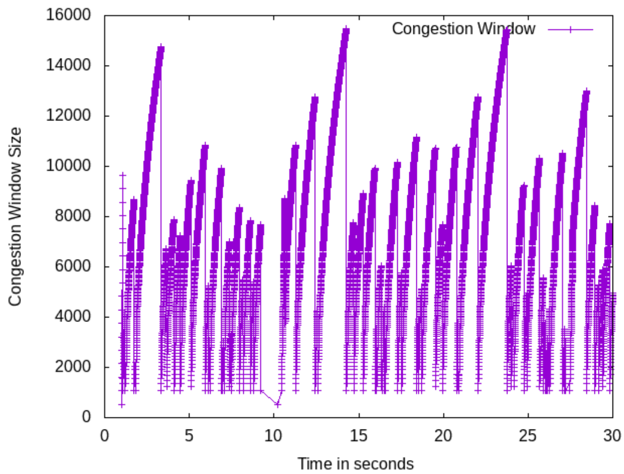
The application data rate is 2Mbps for all plots below.



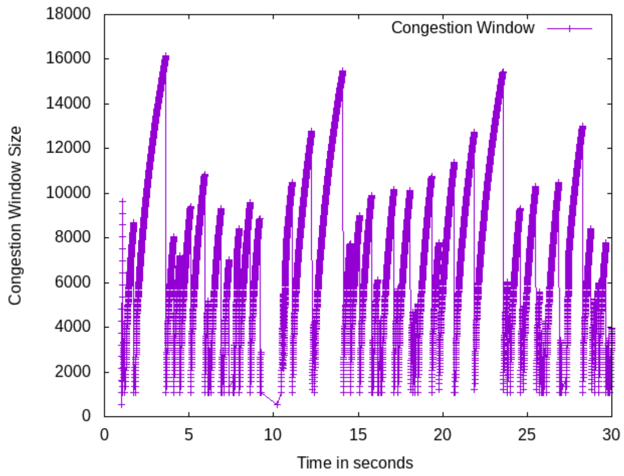
Channel Data rate - 2Mbps



Channel Data rate - 4Mbps



Channel Data rate - 20Mbps



Channel Data rate - 50Mbps

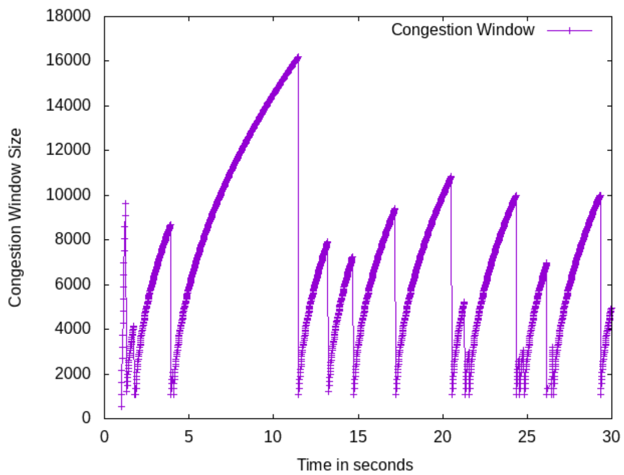
2.a.2 Observation

1. Observe that the average and the peak congestion window size increases as the channel data rate increases. This is reasonable since, increase in channel data rate implies less congestion and hence less packet loss.
2. Also, the rate of increase in `cwnd` also increases with increase in channel data rate. This is reasonable since increase in channel data rate implies decrease in transmission delay which implies faster receiving of `ACK`s by the sending side that causes faster increase of `cwnd` as determined by slow-start/congestion-avoidance algorithms.
3. Interestingly, the number of packet drops also increases with increase in channel data rate. This occurs because of the faster increase in `cwnd`(as mentioned above) due to which the channel gets congested soon too.

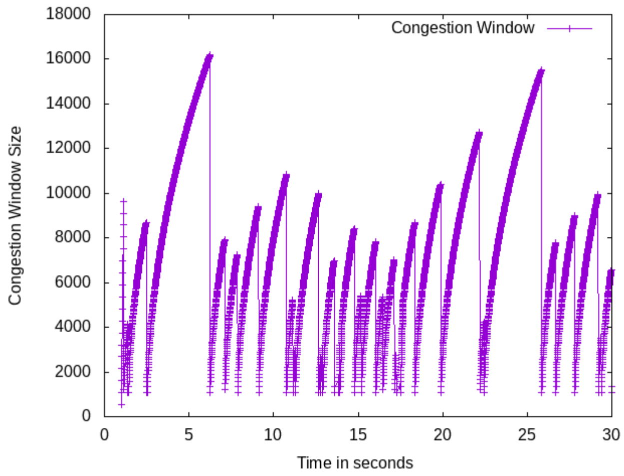
2.b (b)

2.b.1 Plots

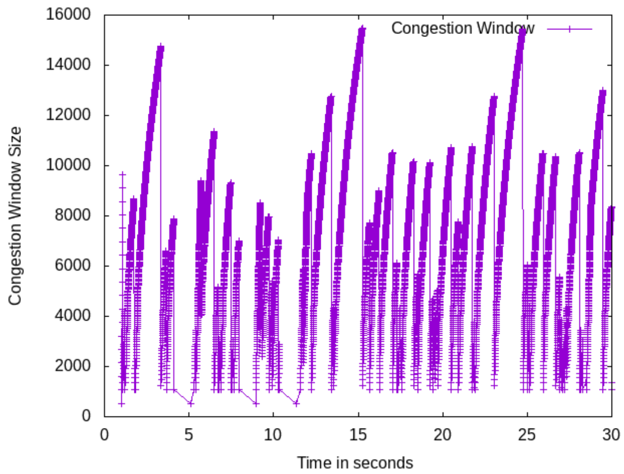
The channel data rate is 6Mbps for all plots below.



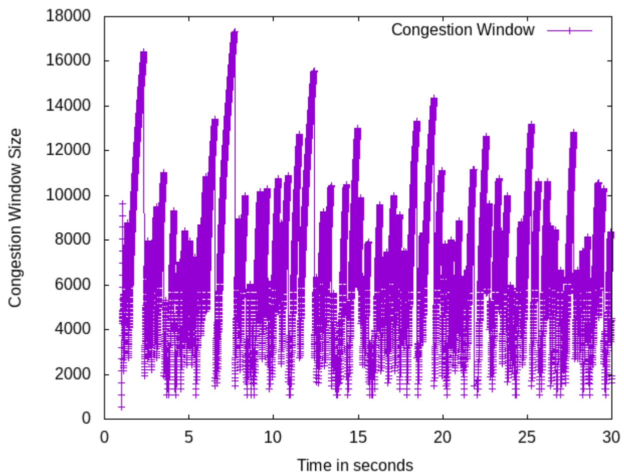
Application Data rate - 0.5Mbps



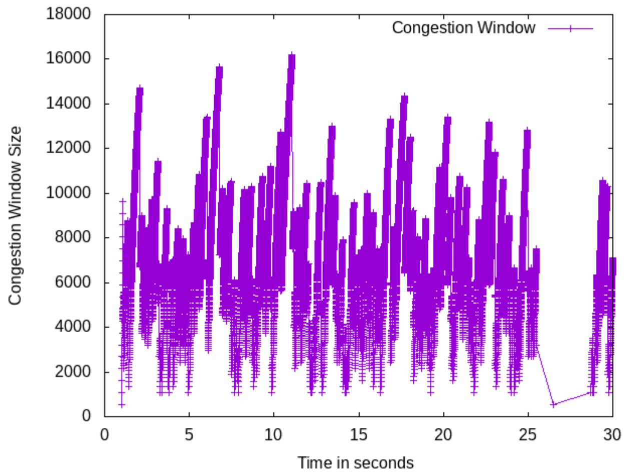
Application Data rate - 1Mbps



Application Data rate - 2Mbps



Application Data rate - 4Mbps



Application Data rate - 10Mbps

2.b.2 Observation

1. Observe that the peak cwnd size remains roughly the same over all values of application data rates. This is reasonable since parameters such as round-trip time and transmission delay are same in all cases and hence the point where congestion starts happening remains same.

2. Observe that the rate of increase of `cwnd` size also increases as application data rate increases. This is reasonable since as application data rate increases, `ACK`s are received more frequently and hence the increase in cwnd caused by the slow-start/congestion-avoidance algorithms occurs more frequently.

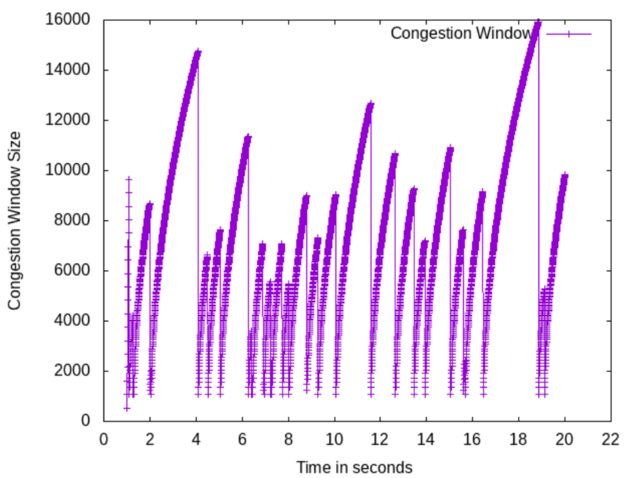
3. The number of packet drops also increases with increase in application data rate. Again, this has similar reason as explained above(for the channel data rate case).

3 Part-3: Analysis of Custom TCP variant

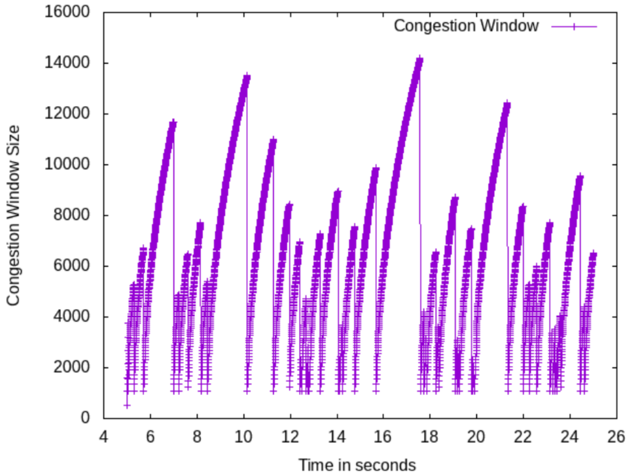
As described in the problem, the protocol `TcpNewRenoCSE` employs a linear increase of `cwnd`(per `RTT`) in slow start phase and an exponential increase in the congestion avoidance phase. The effects of this change have been analysed below.
The code for this part can be run by:
`./waf --run "<file-location> --config=<1/2/3>"`

3.a Configuration-1

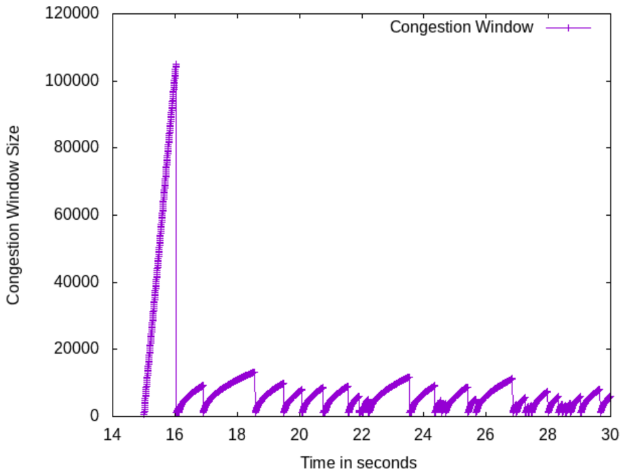
3.a.1 Plots



Connection-1



Connection-2



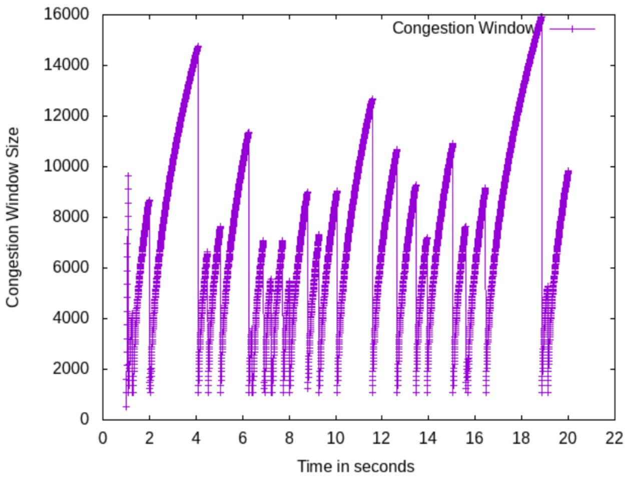
Connection-3

3.a.2 Total dropped packets

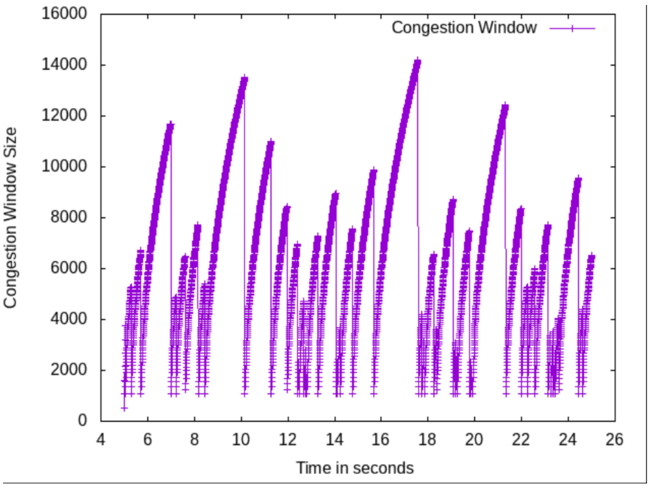
Total dropped packets is 113.

3.b Configuration-2

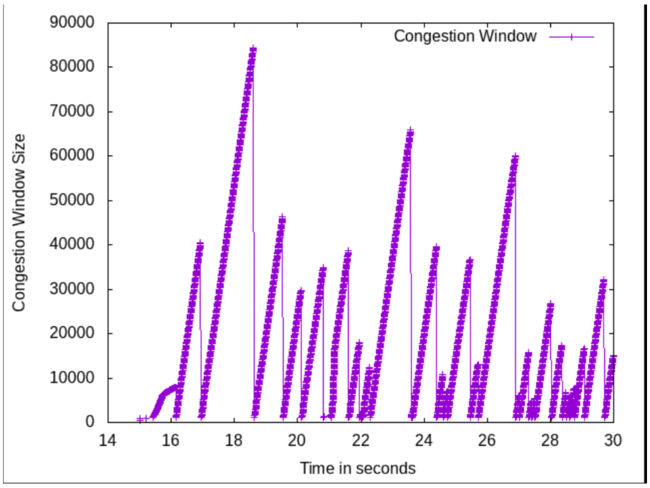
3.b.1 Plots



Connection-1



Connection-2



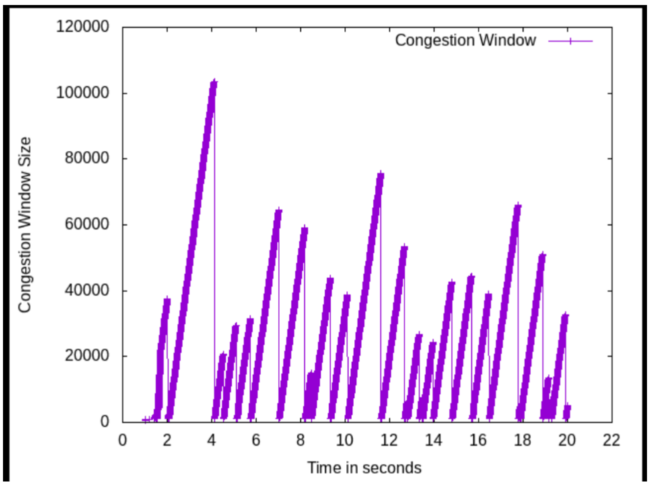
Connection-3

3.b.2 Total dropped packets

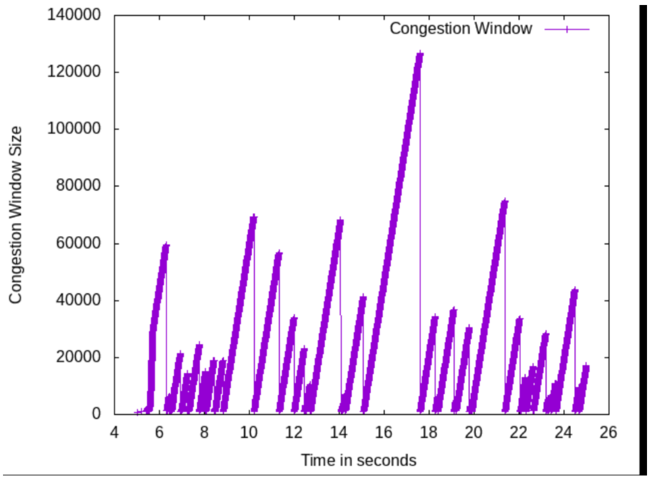
Total dropped packets is 112.

3.c Configuration-3

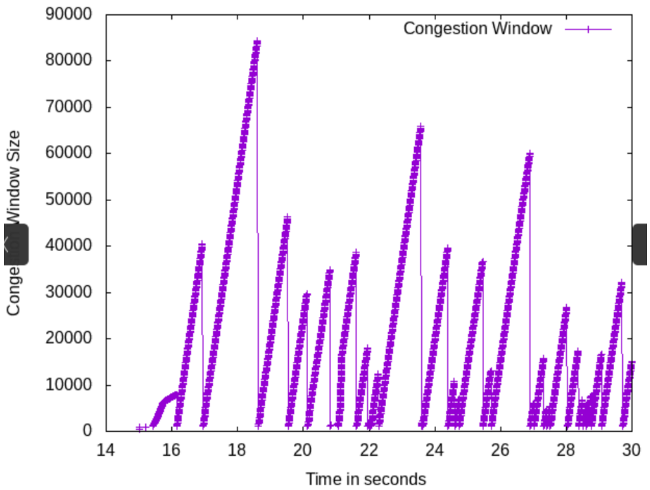
3.c.1 Plots



Connection-1



Connection-2



Connection-3

3.c.2 Total dropped packets

Total dropped packets is 110.

3.d Comparison between `TcpNewReno` and `TcpNewRenoCSE` and its effects

The effect of changing one connection protocol from `TcpNewReno` to `TcpNewRenoCSE` can be observed by comparing the plots of configuration-1 and configuration-2.

1. Observe that switching from `TcpNewReno` to `TcpNewRenoCSE` significantly increases the throughput of Connection-3. This can be explained as follows. In Configuration-1, when Connection-3 starts sending data, the congestion window sizes of Connection-1 and Connection-2 are already high enough. Also, observe that the channel data rate of Connection-3 is also smaller. The buffers(corresponding to the error rate model in ns-3) are occupied with packets from Connection-1 and Connection-2 and hence Connection-3 faces frequent data losses leading to small `cwnd` and hence, small throughput.

2. Intuitively, this suppression of Connection-3 can be countered if the packets from it compete more for space in the buffer at the TCP sink. The new protocol `TcpNewRenoCSE` does this by inducing an exponential(with an exponent of 1.5) increase in `cwnd`. This helps in increasing the thtoughput of Connection-3.

3. In the overall network, the distribution of dataflow has become more equitable.

4 Submission

The submission contains three `.cc` files for the three parts and a `.plg` for plotting using `gnuplot`. The command line commands to use the files have been provided above partwise separately.