COL 334/678

Assignment 3 (NS3 Simulator)

HARIKESH

PART 1: To study how the congestion window size at sender varies with the time for different congestion control protocols.

Topology used: O(n1)------------------O(n2) {n1 is source and n2 is sink}

Data used: Application data rate used = 1Mbps

Data rate of link between n1 and n2 = 8 Mbps

Propagation delay = 3ms

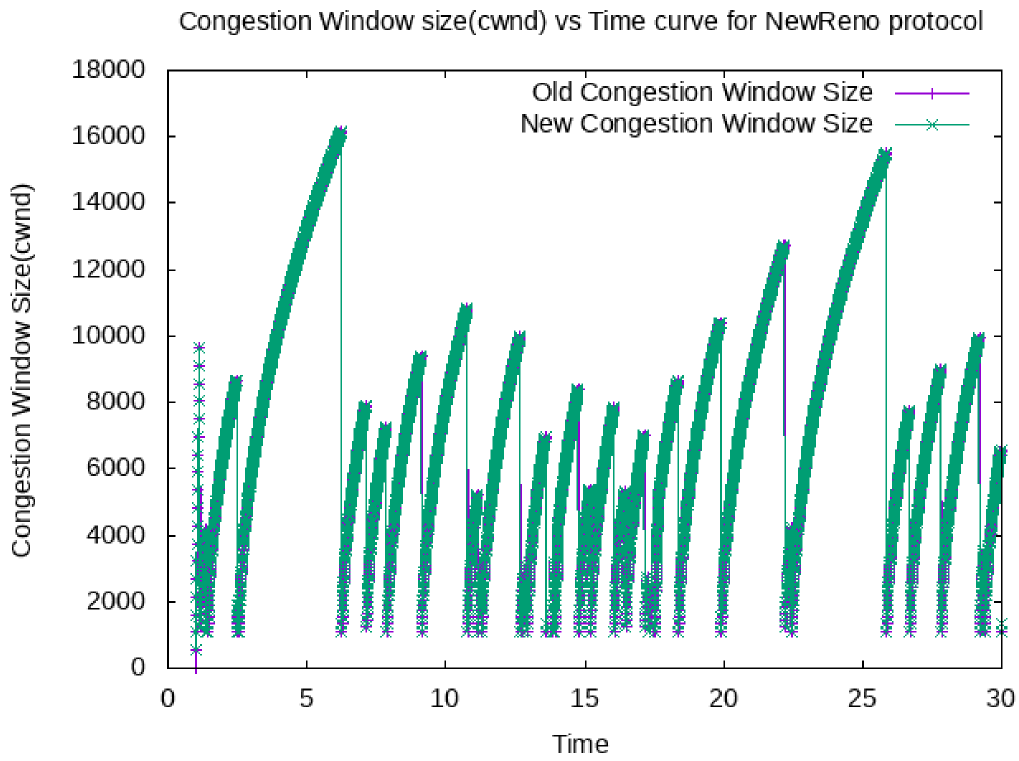
Rate error model with error rate = 0.00001

Packet size = 3000 bytes

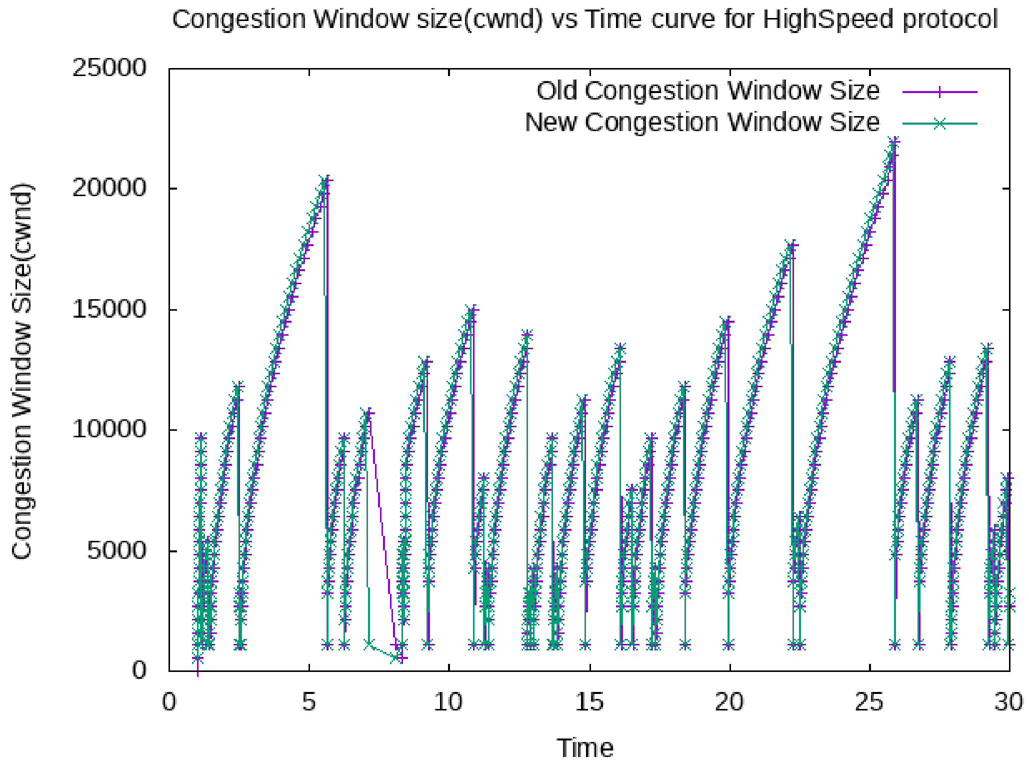
Connection will start at t = 1 sec and end at 30 sec

1 Below are the plots with congestion window size on the y-axis and time on the x-axis (till 30 secs):

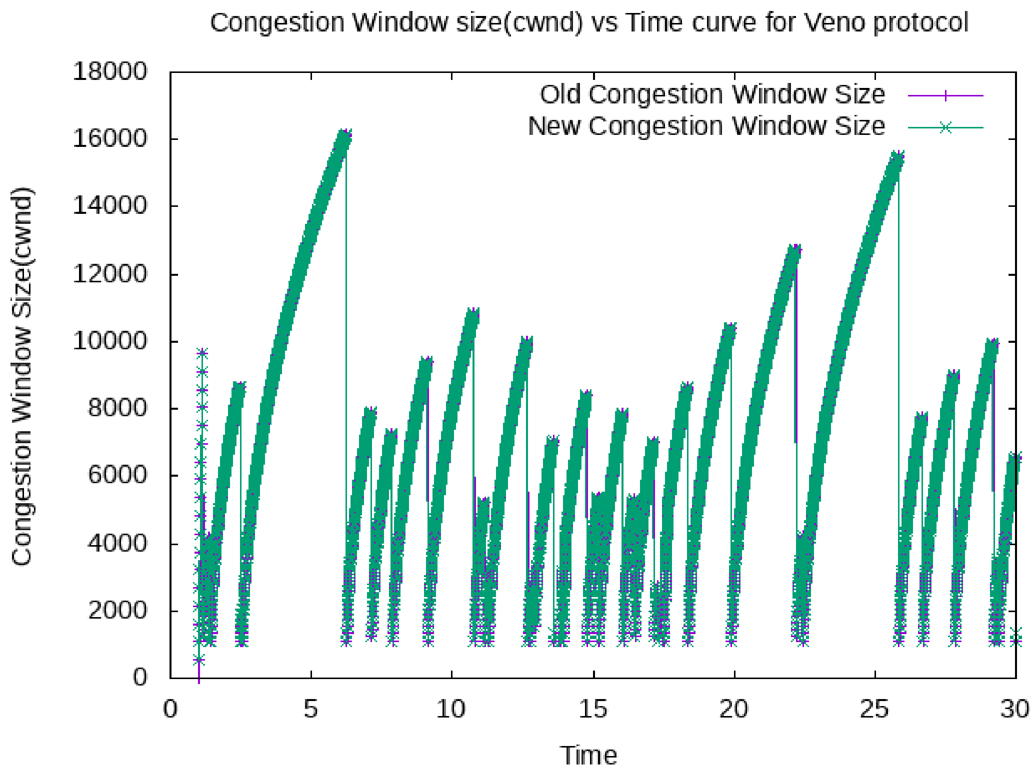
(a) TCP Newreno:



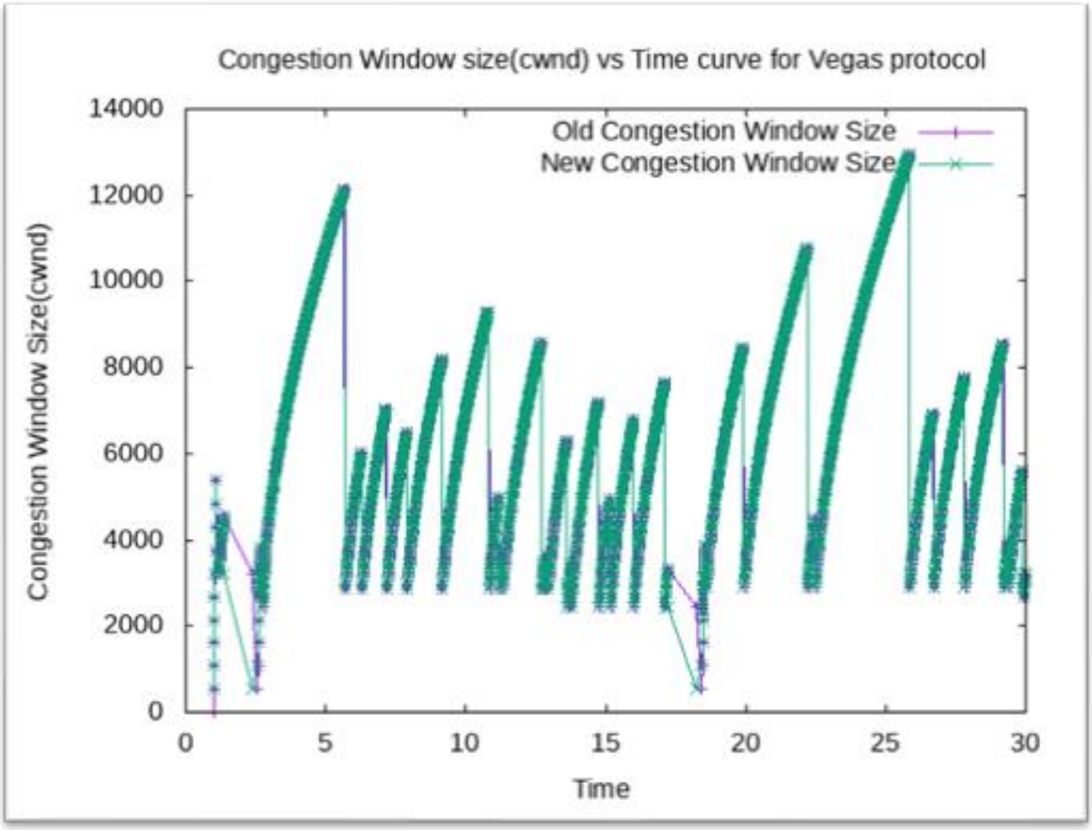
(b) TCP HighSpeed:



(c) TCP Veno:



(d) TCP Vegas:



All of these plots are generated from the cwnd files of corresponding protocols which are generated during execution of our first.cc file.

2. Now, for each of the protocols we have to find the total number of packets dropped and it can be founded by declaring a global variable in our script and then incrementing the total count every time our function RxDrop is called and it then gives us the total number of dropped packets. The below snippets of the output shows the same for different protocols:

```
29.8094 5403
29.8312 5456
29.8334 5508
RxDrop at 29.8534
29.8552 5560
29.857  5611
29.8787 2910
29.9032 3008
29.9097 3103
RxDrop at 29.9242
29.9278 3195
29.929  2659
29.9512 2767
29.9524 2870
29.9583 2970
29.9752 3066
29.9817 3159
29.9992 3249
The number of packets dropped are: 39hari@DESKTOP-NOATFN0:/mnt/c/Users/HARIKESH/Downloads/ns-allinone-3.29/ns-3.29$
```

TCP NewReno:

The number of packets dropped are: 38.

TCP HighSpeed:

The number of packets dropped are: 38.

TCP Veno:

The number of packets dropped are: 38.

TCP Vegas:

The number of packets dropped are: 39.

Inference: We can see that almost in al the protocols the number of packets dropped are nearly same and this is due to the fact that all of these have all the three almost same procedures i.e. slow start then congestion avoidance phase and then congestion detection phase. In these the number of packets send increases exponentially first and then after reaching the threshold they increase by 1 and then finally we go back to slow start phase because of the multiplicative decrease. The number of packet drops are almost same because of the fact that packet drop itself is an indicator of having congestion avoidance.

3. In this part, we have to basically explain the trends obtained in the plots of these protocols and also briefly explain the algorithms used by the different protocols in different phases.

TCP NewReno: We know that new reno doesn't have ssthresh immediately. During slowstart the TCP increments command by atmost SMSS. So the formula for increment of window size is: cWnd += min(N, SMSS)

TCP HighSpeed: In HighSpeed MAthematiclly the formula for incrementing the cWnd size is as below : cWnd = cWnd + a*(cWnd)/cWnd

Which is set to such a value cWnd = (1 – b(cWnd))*cWnd

TCP Veno: In veno the amount of extra packets at queue are given by cWnd/RTT - cWnd/BaseRTT. Veno makes decision on its Cwnd modifications based on calculated n and threshold beta.

TCP Vegas: In vegs the amount of extra packets at queue are given by cWnd/BaseRTT - cWnd/RTT.

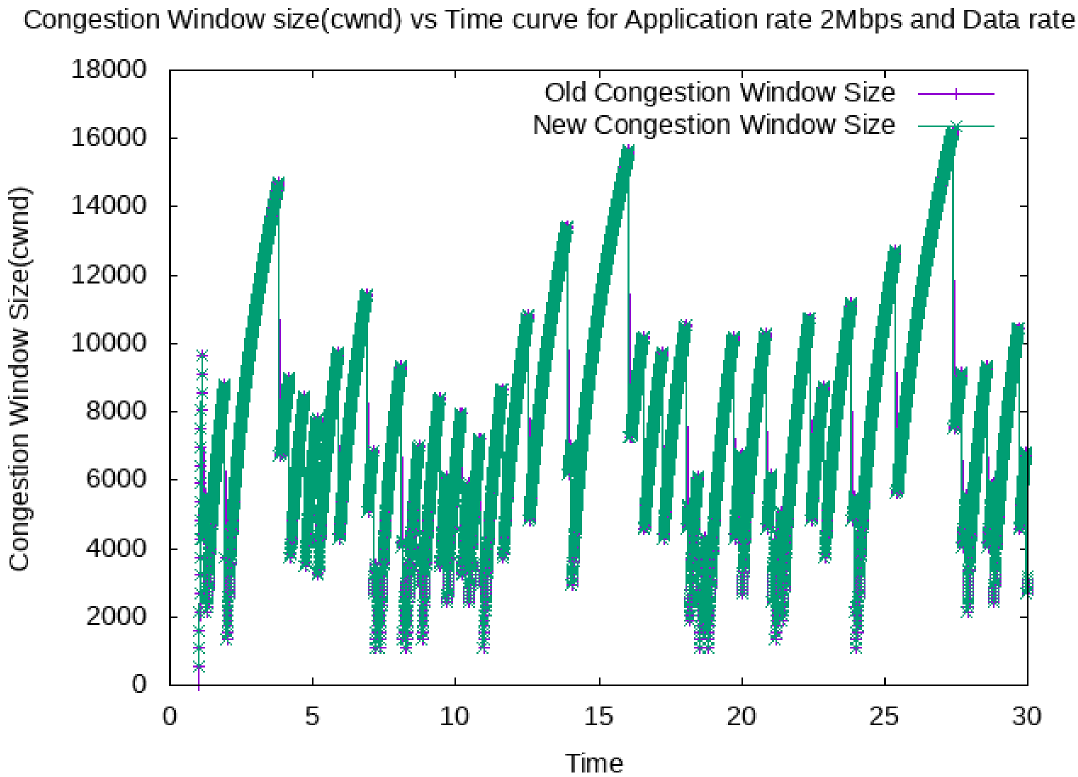
It is a pure delay based congestion control algorithm.

PART 2: In this part we are going to analyse the effect of the bandwidth/application data rate on congestion window.

Protocol used: TCP NewReno, Same topology, Channel delay 3ms, Error rate = 0.00001, Packet size = 3000 bytes, Time will be between t = 1 sec to t = 30 sec.

Part (a) Here we are going to plot the congestion window size vs time graph for TCP connection using fixed Application rate of 2Mbps.

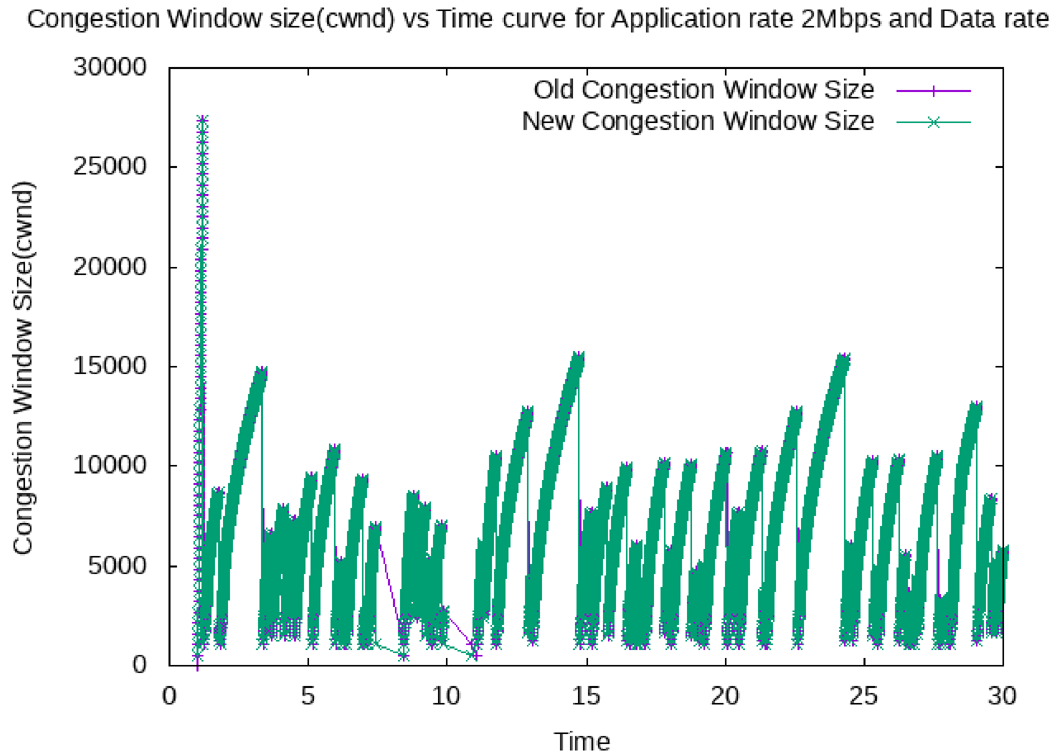
Plot 1: Channel data rate = 2Mbps



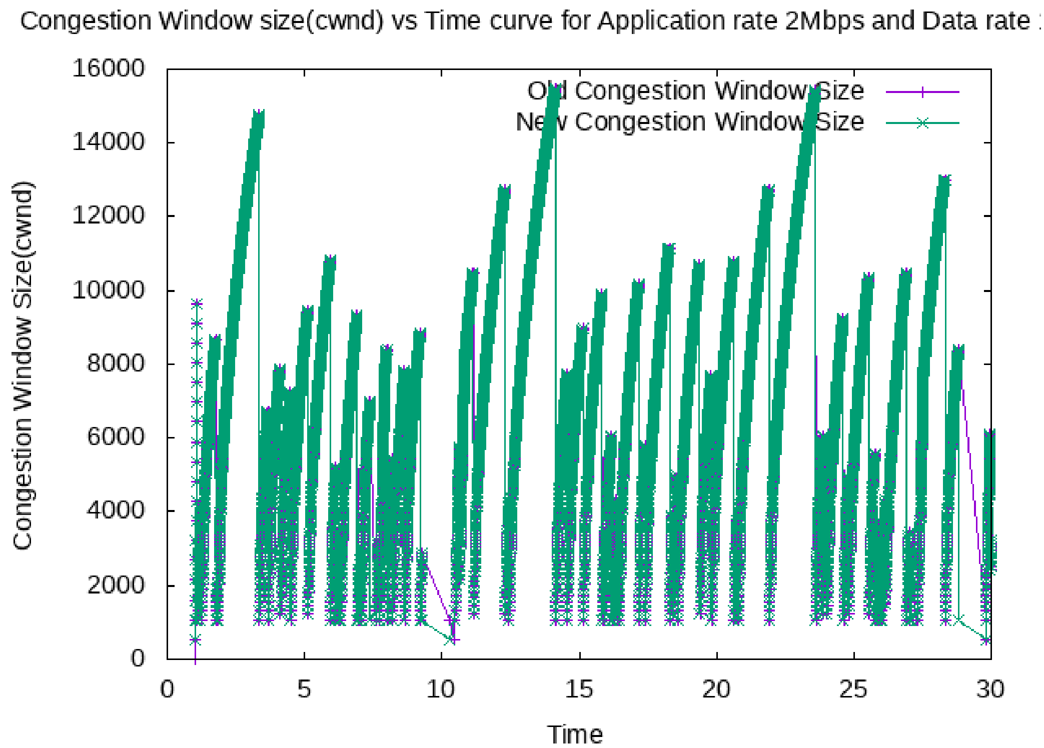
The number of packets dropped in this case is 62.

Plot 2: Channel data rate = 4Mbps

The number of packets dropped in this case is 72



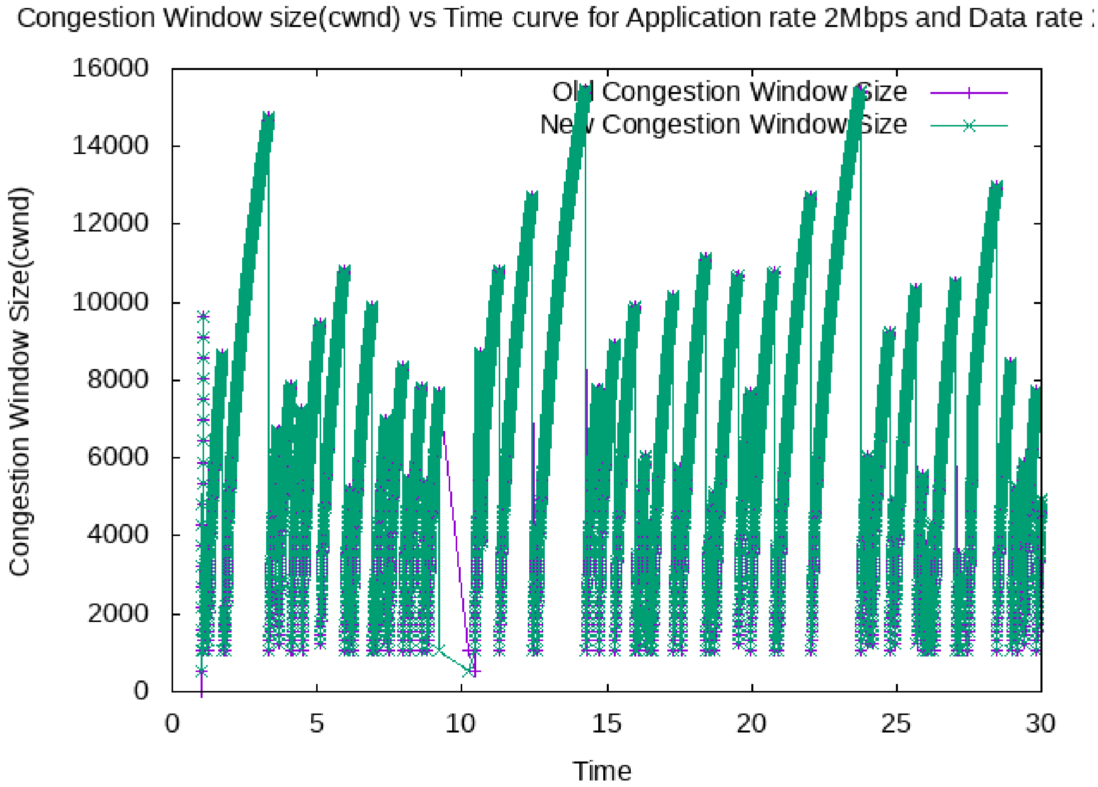
Plot 3: Channel data rate = 10 Mbps



The number of packets dropped in this case are 73.

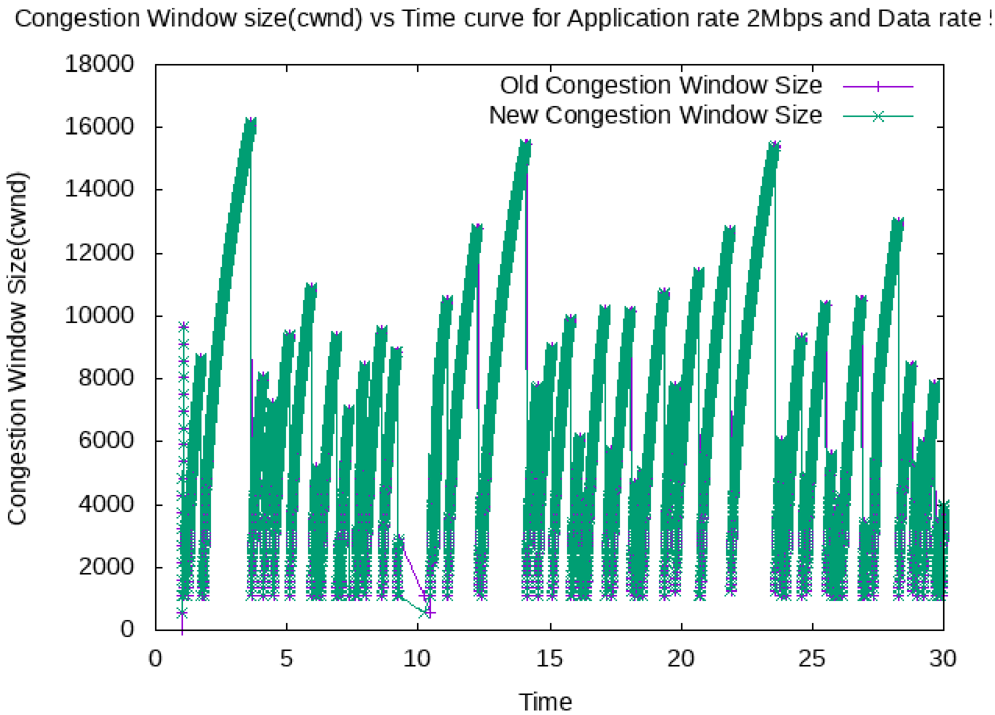
Plot 4: Channel data rate = 20Mbps

The number of packets dropped in this case are 74.



Plot 5: Channel Data rate = 50 Mbps

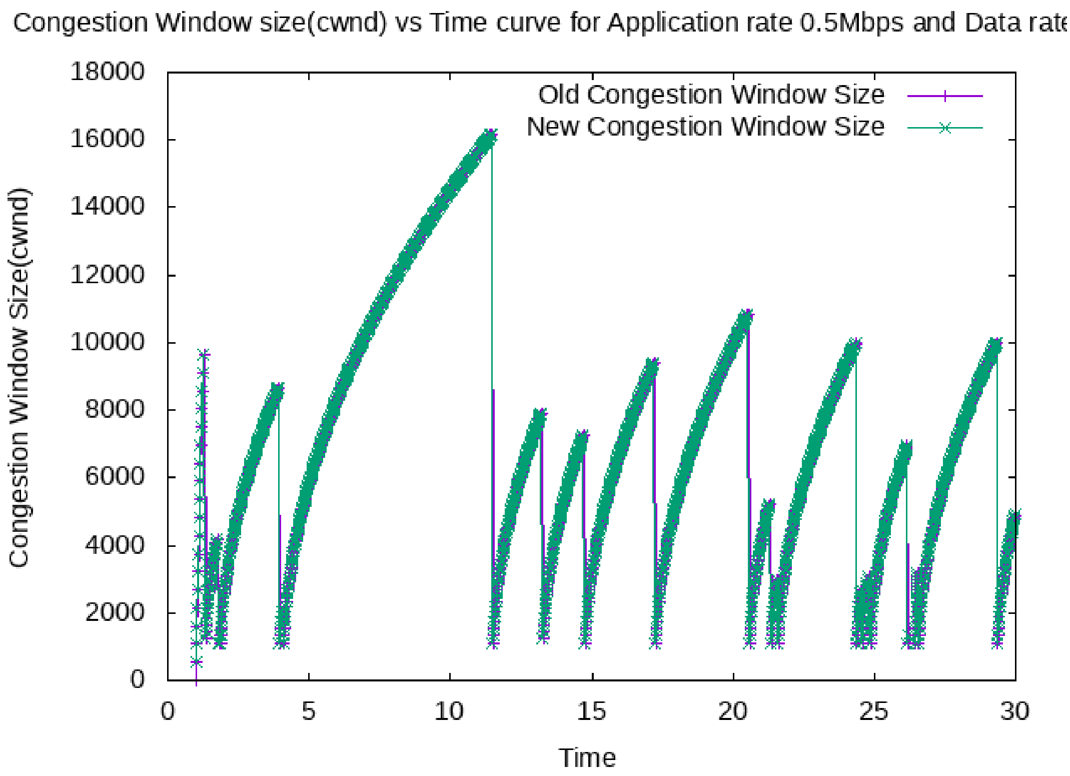
The number of packets dropped in this case is 75.



In the above part we can see that on increasing the

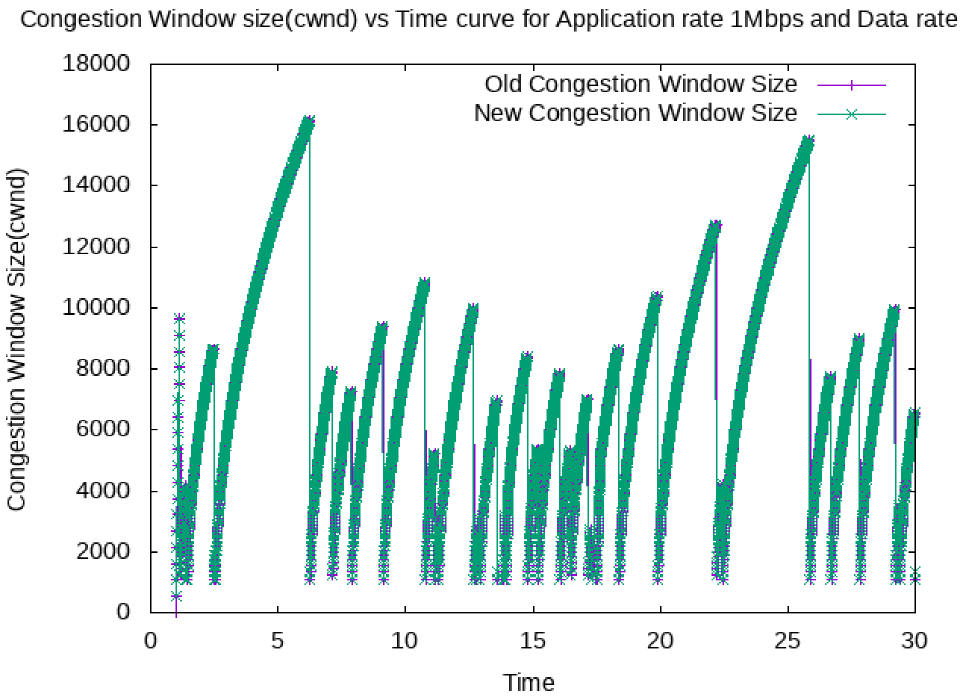
Part (b) In this part we have to plot the congestion window vs time graph for TCP connection using fixed data rate of 6Mbps between the nodes.

Plot 1: Using application rate of 0.5Mbps



The number of packets dropped in this case are 22.

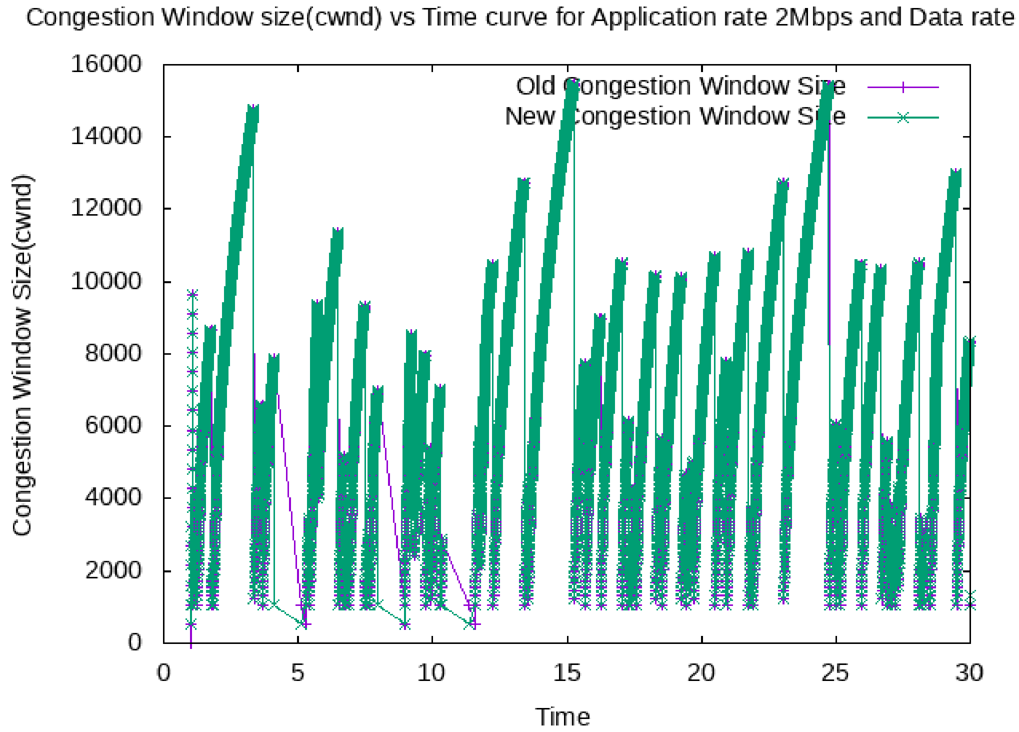
Plot 2: Using Application rate of 1Mbps



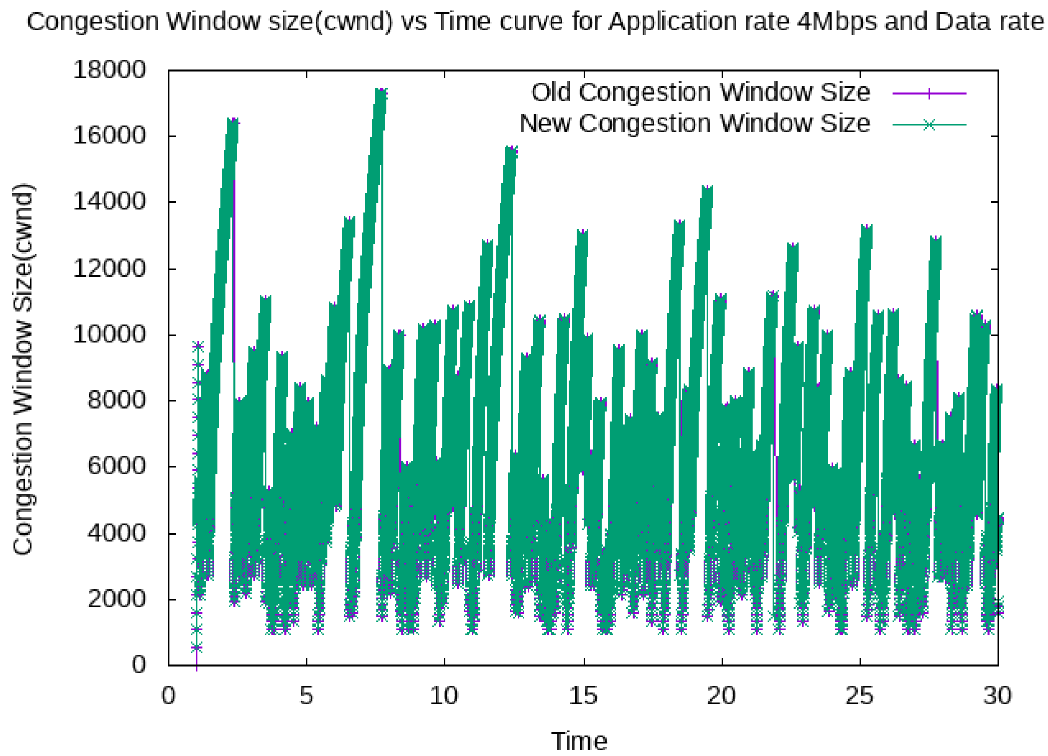
The number of packets dropped in this case are 38.

Plot 3: Using application rate of 2Mbps

The number of packets dropped in this case are 71.



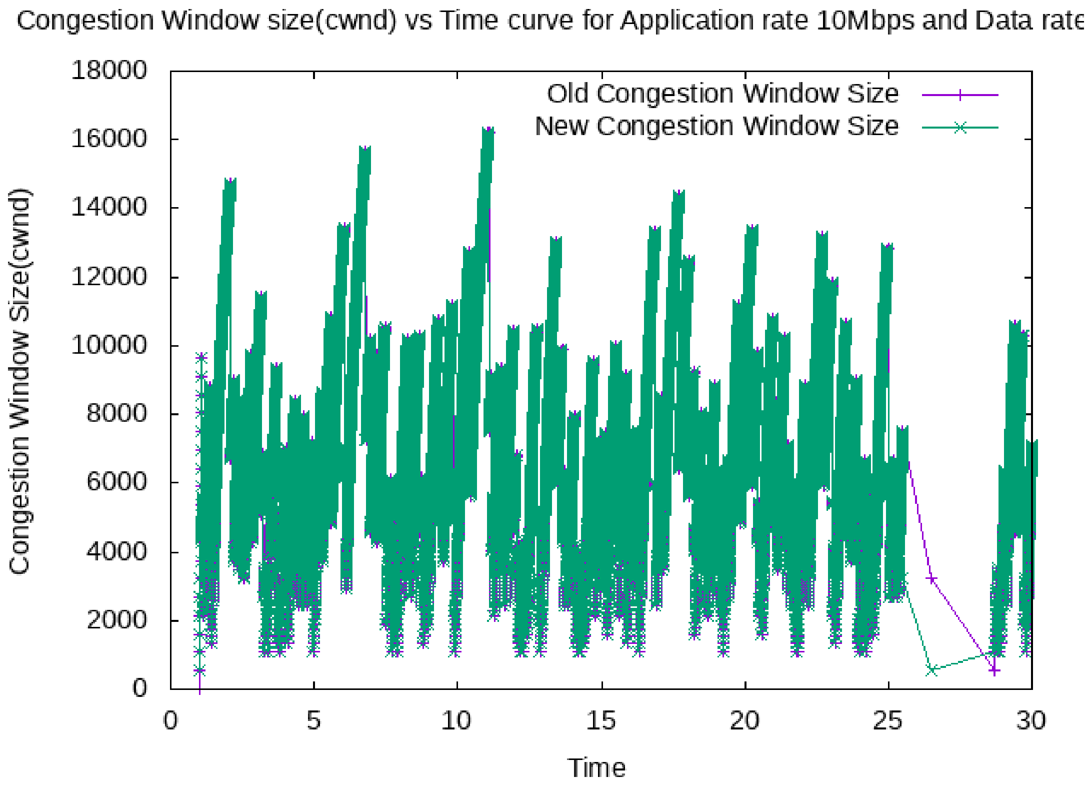
Plot 4: Using application rate of 4Mbps



The number of packets dropped in this case is 156.

Plot 5: Using Application rate of 10Mbps:

The number of packets dropped in this case is 156.



From the plots of part 2a we can see the three processes going on clearly i.e. processes of congestion control mechanism. We can see that initially the congestion window size is increasing and after that the congestion window size is decreasing by a multiplicative factor. One more thing to notice here is that as the channel data rate is increasing the number of packet drops are also increasing and this is due to the fact that with increasing data rates the queues of the forwarding nodes overflow which results in more loss of packets.

For the plots of part2b we can see that as we are increasing the application data rate the graph is becoming more and more denser because now it reaches all the values quickly and after that decreases quickly in time. Due to these quick transitions i.e. leaps there are a lot of packet drops going on as we are going to increase the application data rate of this network. This is simply due to the fact that as the number of packets are going to increase we are going to have a lot of packet drops due to less space for queuing them.

Part 3:

Here we have to form our own congestion control protocol. Our new control which we have to form should be named TcpNewRenoCSE. There are only a few changes we have to do which are:

In the slow start phase of TcpNewRenoCSE, increase the congestion window size as:

Cwnd = Cwnd + (segmentsize)^1.9/Cwnd

In the congestion avoidance phase of TcpNewRenoCSE, we have to update the congestion window as:

Cwnd = Cwnd + (0.5)*Segmentsize

Topology: we have three nodes with N1-N3 and N2-N3 connected via point to point link.

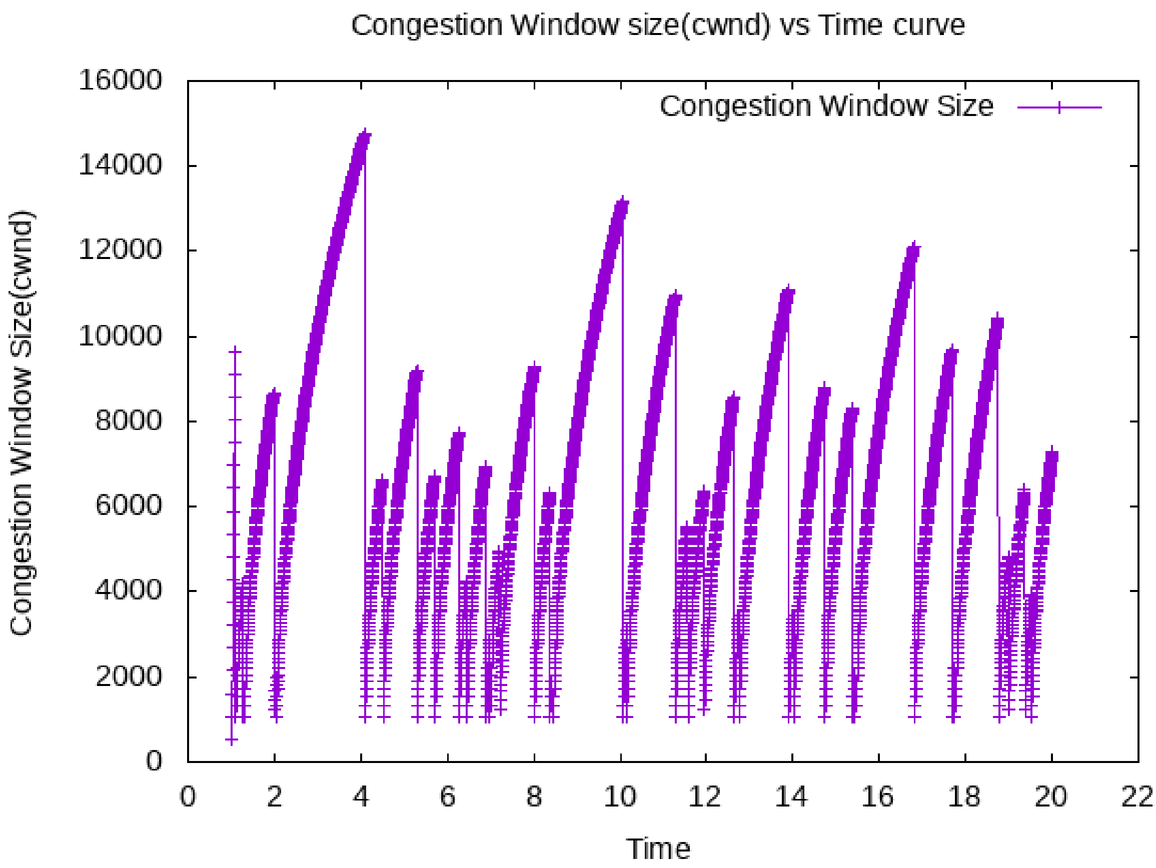
Application data rate = 1.5Mbps, package size = 3000 bytes

The respective time of the starting and ending of all the connections is also given.

1. Congestion window size vs time plots (from time t = 1 to t =30)

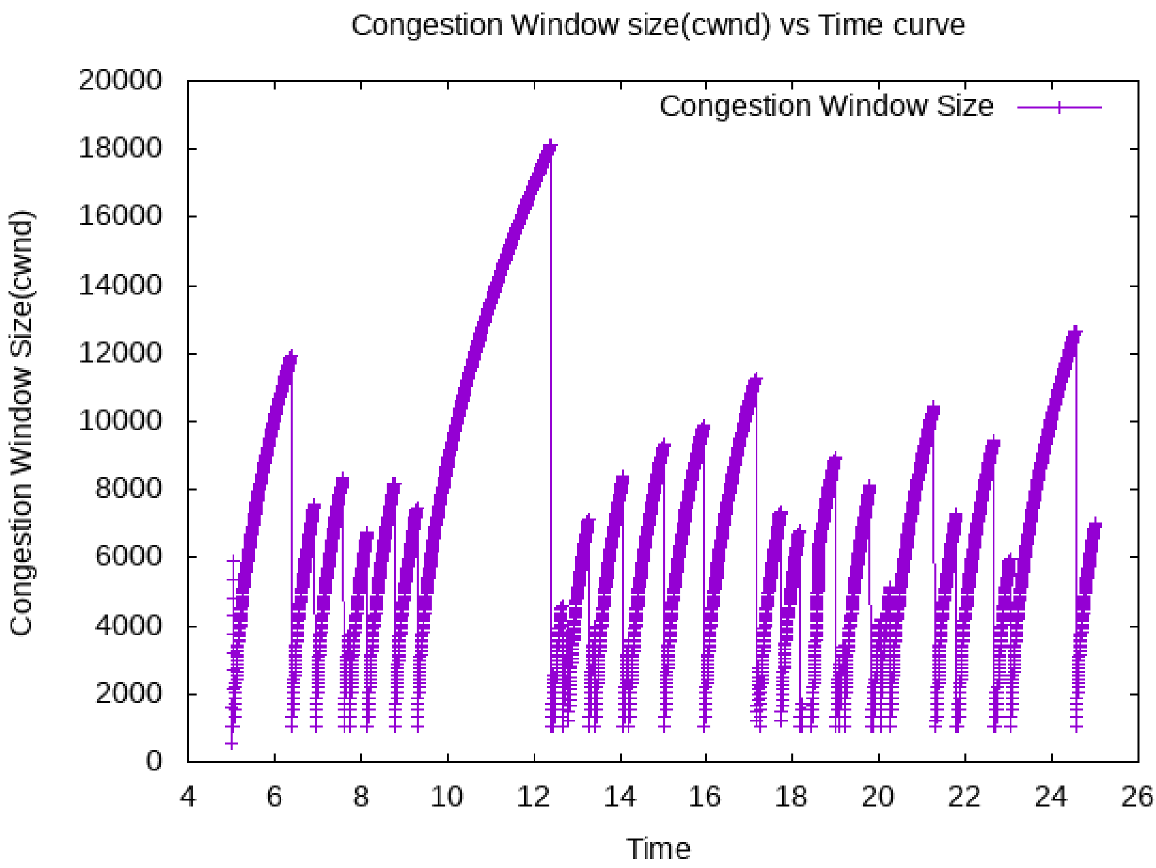
   Configuration 1:

   Sender 1 :



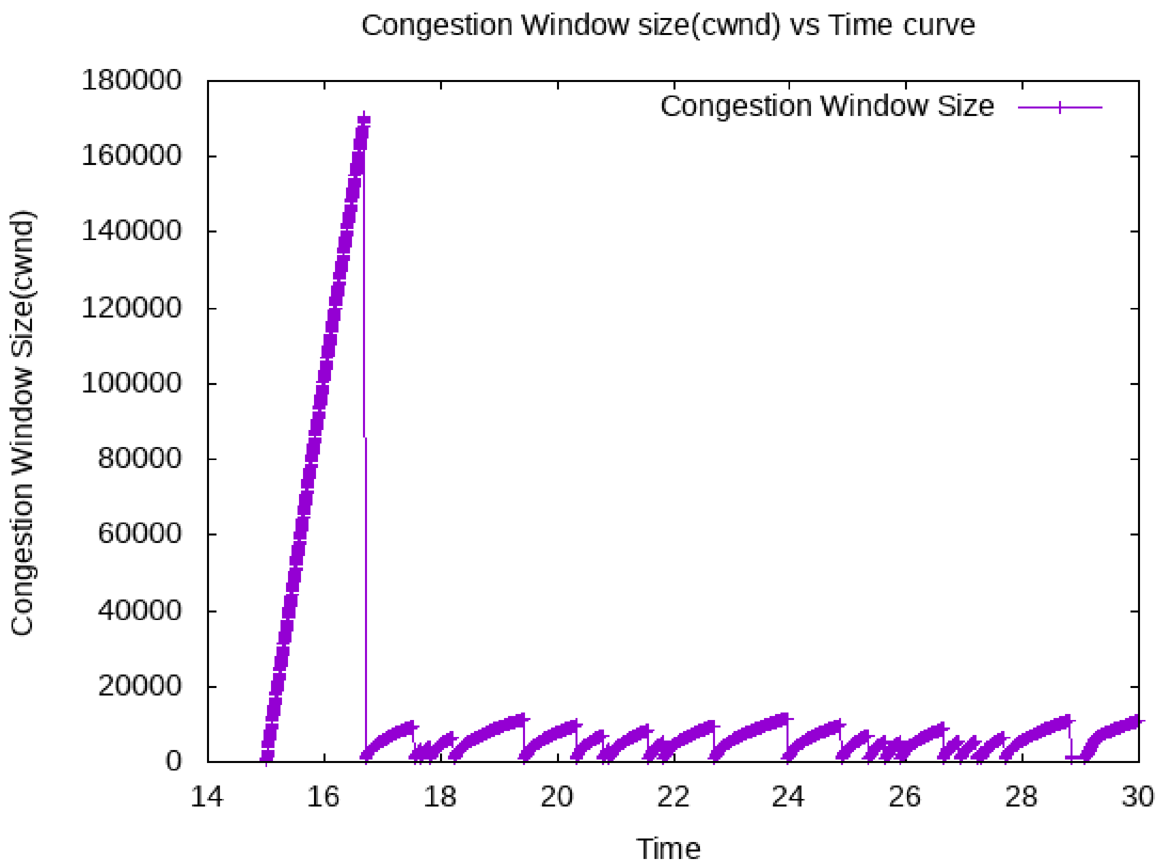
Congestion Window size(cwnd) vs Time curve
Congestion Window Size
Congestion Window Size(cwnd)
Time
0
2000
4000
6000
8000
10000
12000
14000
16000
0
2
4
6
8
10
12
14
16
18
20
22

Sender 2:



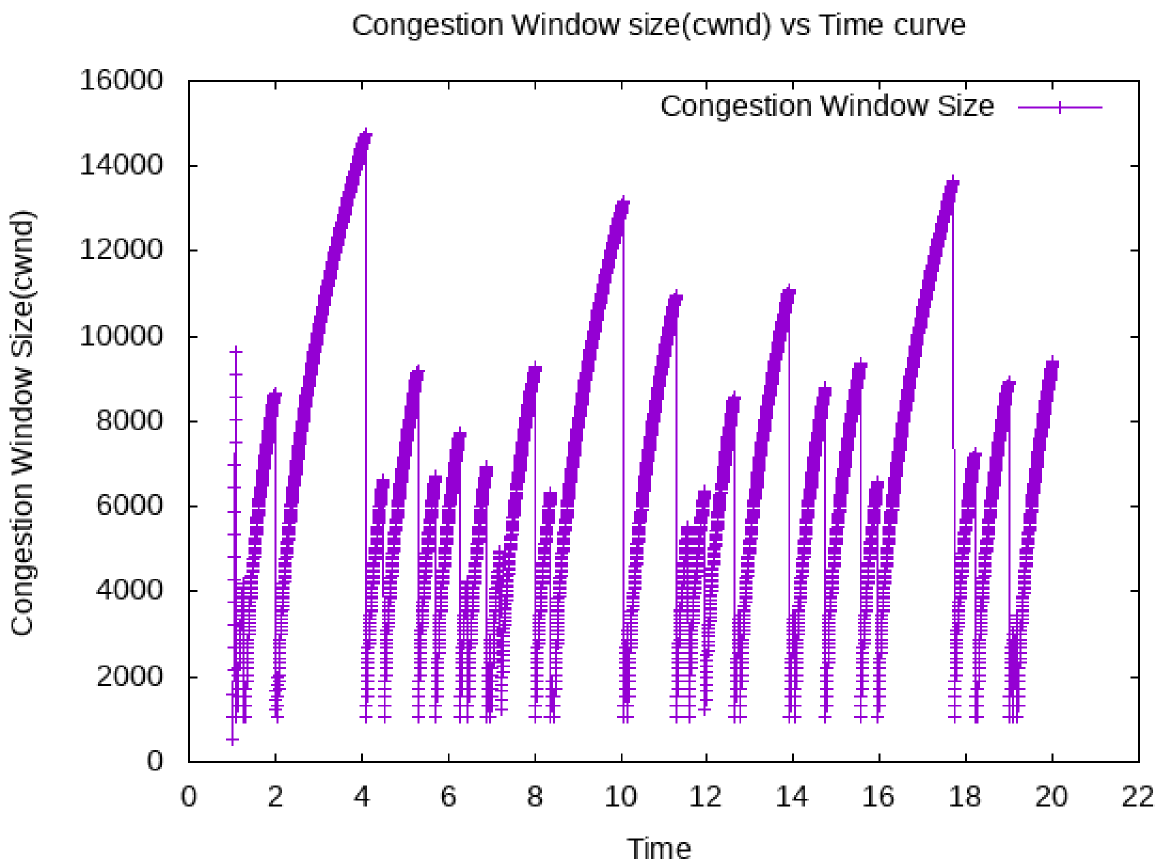
Congestion Window size(cwnd) vs Time curve
Congestion Window Size
Congestion Window Size(cwnd)
20000
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
4
6
8
10
12
14
16
18
20
22
24
26
Time

Sender 3:

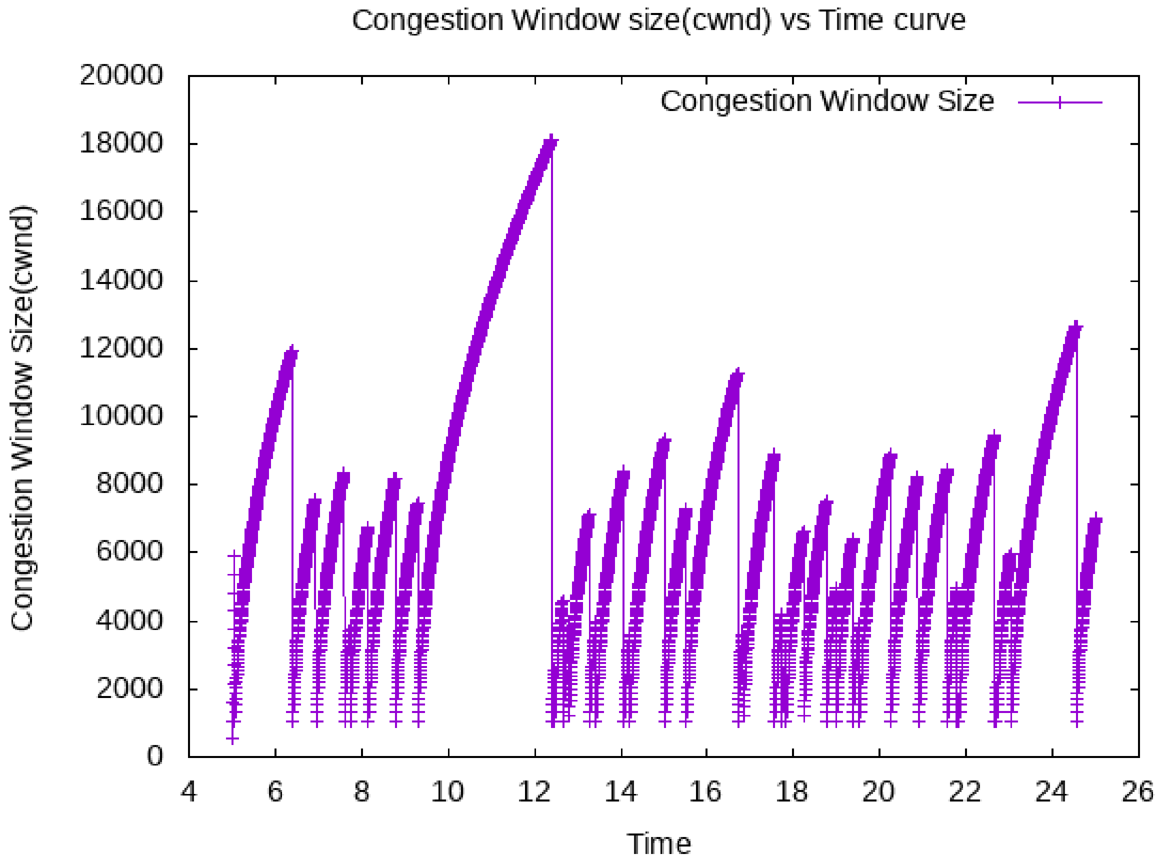


Comfiguration 2:

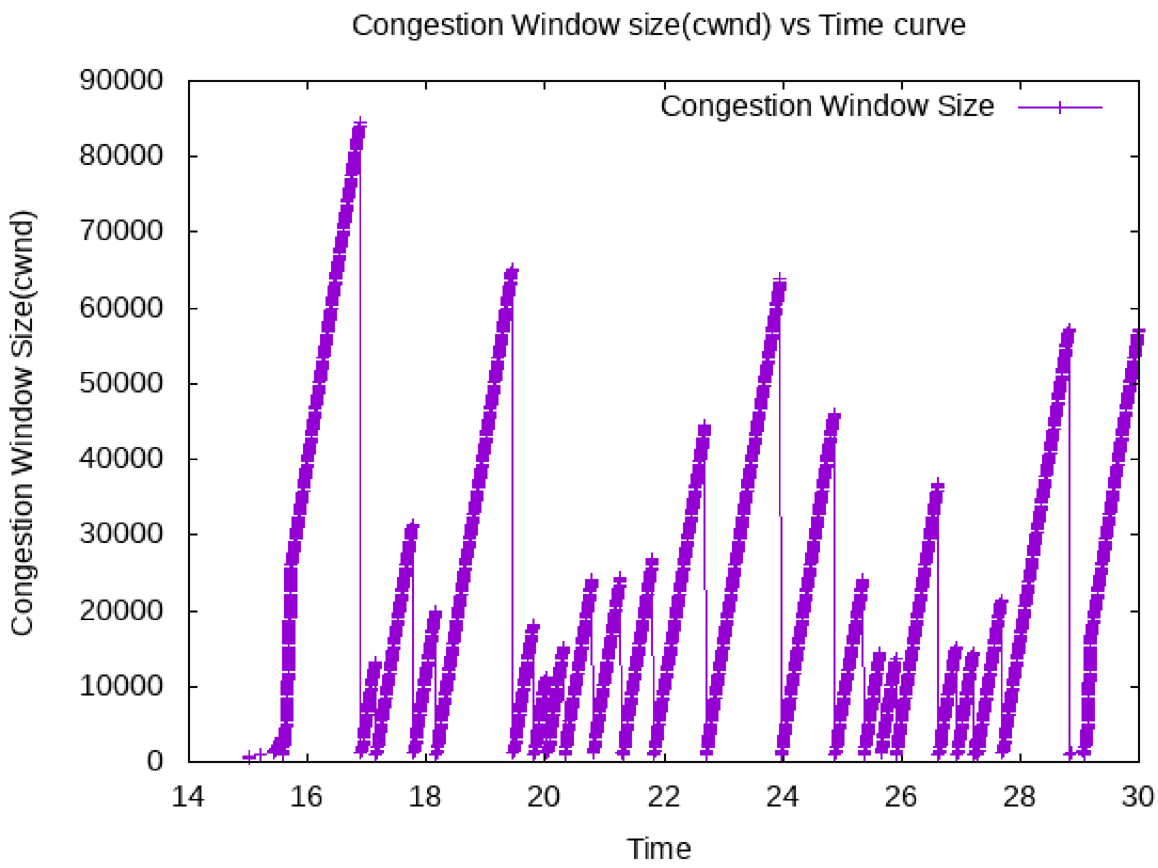
Sender 1:


Congestion Window size(cwnd) vs Time curve
Congestion Window Size
Congestion Window Size(cwnd)
16000
14000
12000
10000
8000
6000
4000
2000
0
0
2
4
6
8
10
12
14
16
18
20
22
Time

Sender 2:

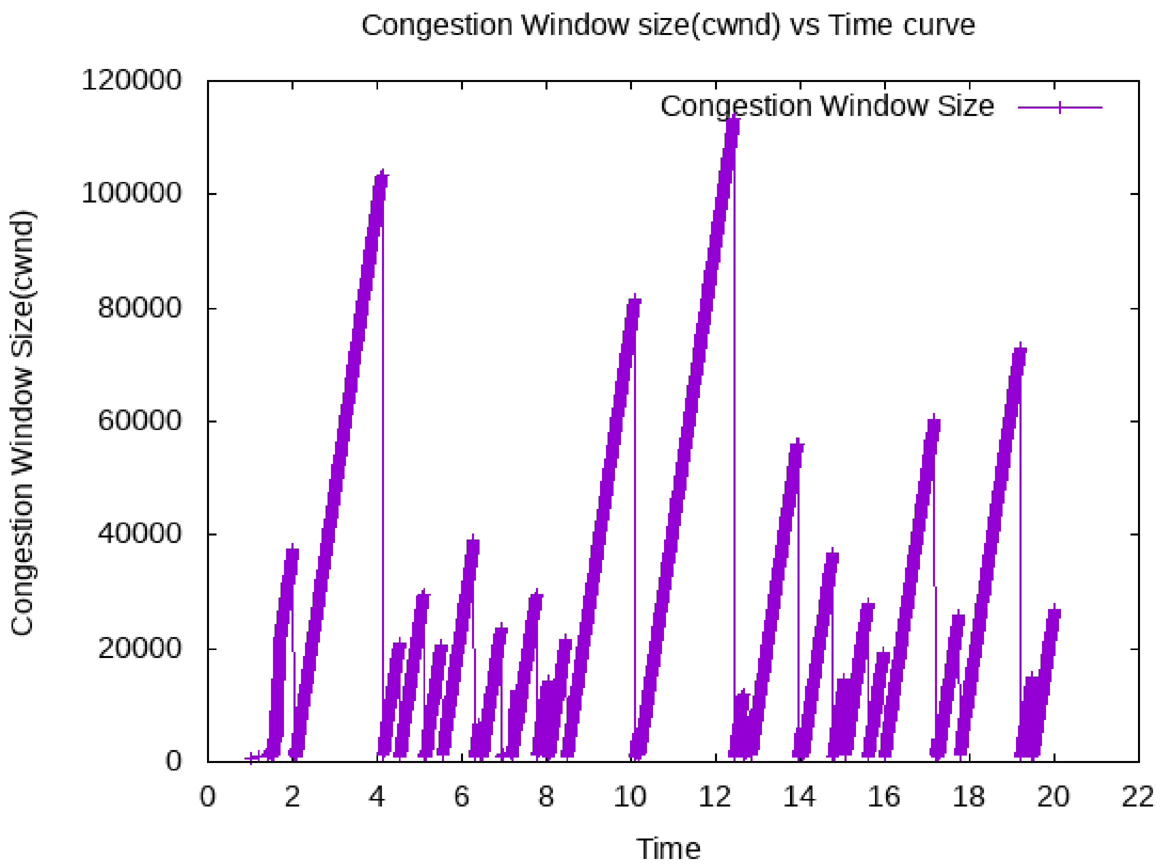

Congestion Window size(cwnd) vs Time curve
Congestion Window Size
Congestion Window Size(cwnd)
Time
20000
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
4
6
8
10
12
14
16
18
20
22
24
26

Sender 3:

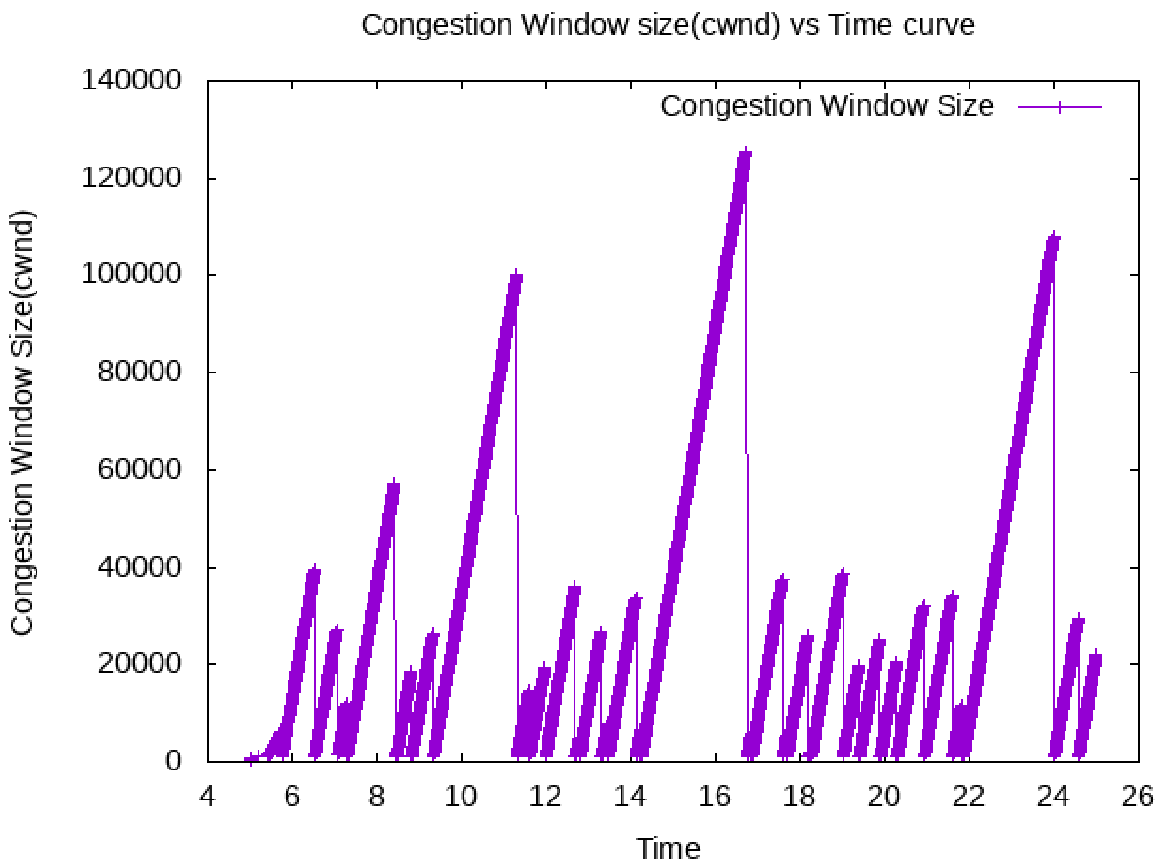


Configuration 3:

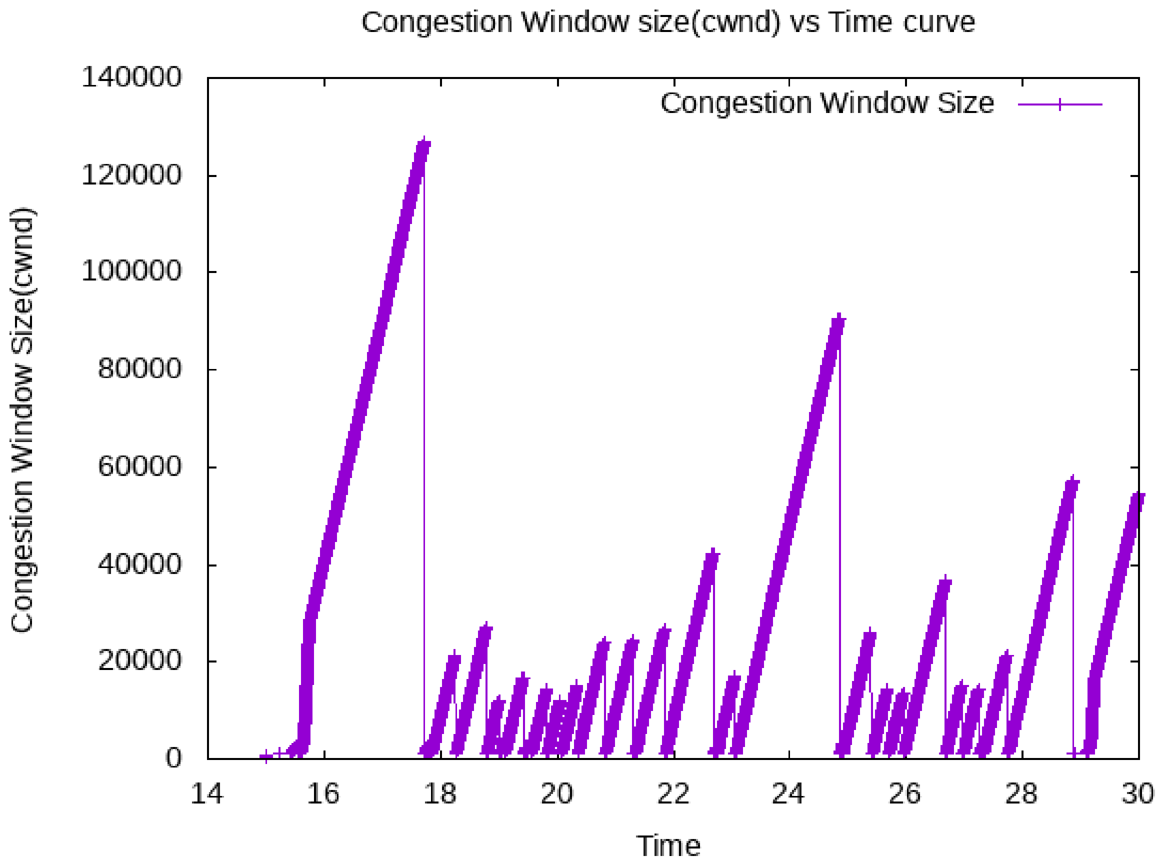
Sender 1:


Congestion Window size(cwnd) vs Time curve
Congestion Window Size
Congestion Window Size(cwnd)
Time
0
20000
40000
60000
80000
100000
120000
0
2
4
6
8
10
12
14
16
18
20
22

Sender 2:


Congestion Window size(cwnd) vs Time curve
Congestion Window Size
Congestion Window Size(cwnd)
Time
0
20000
40000
60000
80000
100000
120000
140000
4
6
8
10
12
14
16
18
20
22
24
26

Sender 3:



(b) In configuration 1 the total number of packets dropped are 105. Packets dropped from connection1 are 76, from connection2 are 19 and connection3 are 10.

In configuration 2 the total number of packets dropped are 103. Packets dropped from connection1 are 74, from connection2 are 19 and connection3 are 10.

In configuration 3 the total number of packets dropped are 108. Packets dropped from connection1 are 78, from connection2 are 20 and connection3 are 10.

(c) On using the two different prtocols i.e. TCPnewreno and TCPnrewRenoCSE the congetion avoidance phase is varying at the sender because the congestion avoidance phase is afte the slow start and we have used a different formula for calculating window size in congestion avoidanace phase. Basically half of segment size is added In our new Algorithm compared to the previous newreno protocol where full segmentsize was added. Also we can se from the graphs that due to this there are less number of multiplicative decrease in our new protocol formed there. This impact the entire network due to the fact that as the congestion avoidance phase will be near the packets sent by that sender will decrease which will affect the data rate of that connection and hence network topology.

References: https://youtu.be/Y37DTIF-kRQ