Q1

|  |  |  |
| --- | --- | --- |
| Protocol | Packets dropped | Plot |
| NewReno | 38 |  |
| HighSpeed | 38 |  |
| Veno | 38 |  |
| Vegas | 39 |  |

Observations:

We see almost similar packet losses in all the 4 protocols. We find that HighSpeed is with largest congestion window and hence has fastest recovery phase. Veno is derived from NewReno and hence we see almost similar graphs with around 10 different congestion window sizes. We also a stop in packet transmission at t=17 seconds in vegas

The SMSS is the size of the largest segment that the sender can transmit.

**Newreno:**

* Congestion avoidance – congestion window is increased by 1 full sized segment
* Slow start – congestion window is increased by min(number of bytes unacknowledged, maximum segment size)
* We can see that till ssthresh value the points are separated by much larger distance that is jump is higher. But when ssthresh is reached points are close as the increment is by 1 – fixed.

**HighSpeed:**

* Designed for high capacity channels, with large congestion windows.
* Cwnd grows much faster and accelerates from recovery faster.
* We see much higher jumps in the plot.
* The average window size is higher compare with any other protocol

**Vegas:**

* Delay based congestion control protocol. Expected and actual throughputs are calculated to get the congestion queue at bottleneck and accordingly cwnd is adjusted
* Linear increase and linear decrease
* Plots show lowest average cwnd size

**Veno:**

* Uses prediction method similar to vegas.
* Refines additive increase algo of reno to increase duration of connection in stable state by increasing cwnd by 1/cwnd for every new ack after bandwidth is fully utilized
* Multiplicative decrease is by 1/5 as loss is assumed to be more likely due to corruption.
* The average window size is higher compare with any other protocol
* Plot is similar to newreno except for fully utilization part

Q2

A) Different channel rate

|  |  |  |
| --- | --- | --- |
| Channel data rate | Plot | Packet dropped |
| 2 Mbps |  | 66 |
| 4 Mbps |  | 72 |
| 10 Mbps |  | 73 |
| 20 Mbps |  | 74 |
| 50 Mbps |  | 75 |

Channel rate is varied but the rate at which application is sending data is constant.

The total packets loss increases as we increase the channel rate

The increase in channel rate at a constant application rate improves the transmission. Increased channel rate makes channel more lossy but the tradeoff is seen with overall increase in congestion window size

B) Different application data rate

|  |  |  |
| --- | --- | --- |
| Channel data rate | Plot | Packet dropped |
| 0.5 Mbps |  | 22 |
| 1 Mbps |  | 38 |
| 2 Mbps |  | 71 |
| 4 Mbps |  | 158 |
| 10 Mbps |  | 100 |

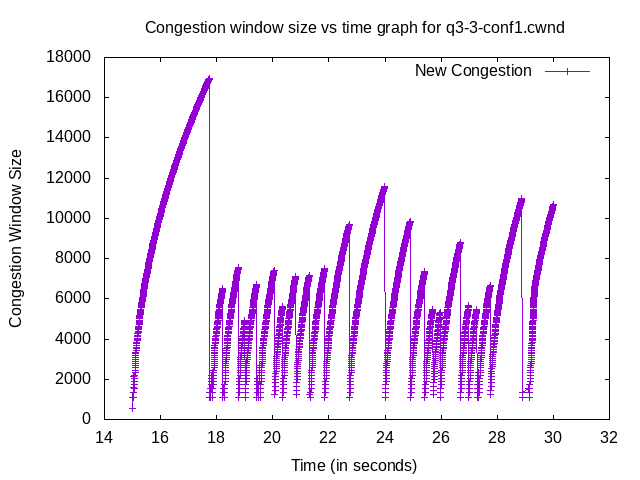
Here channel rate is fixed but application rate is varied.

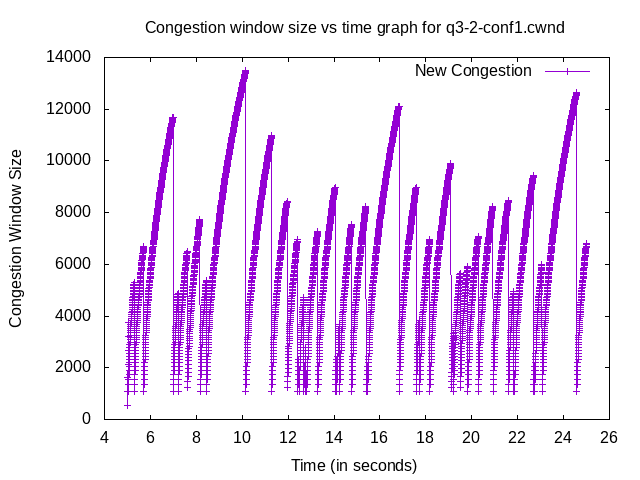
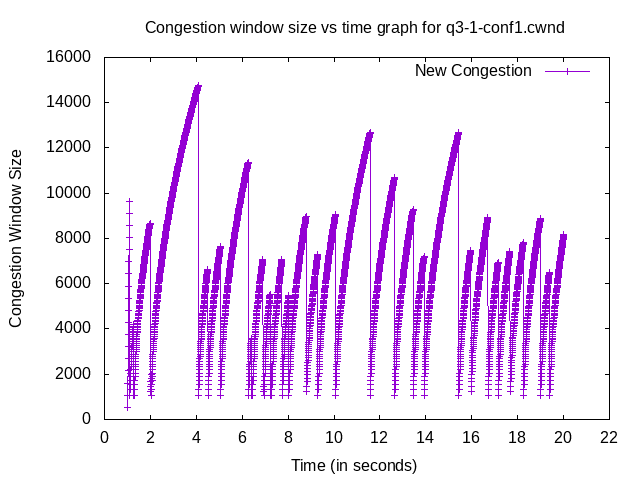
Total number of packets lost increases with the increase in application data rate. Large number of packets lost increases the noise in the graph. This is expected as the channel gets over occupied for more time than less application rate.

Q3

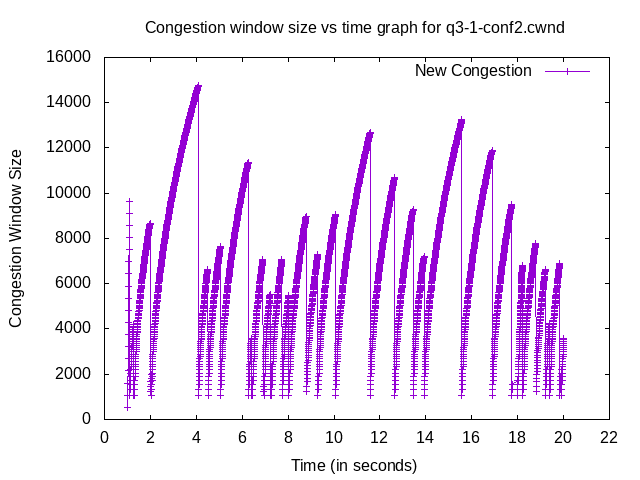
Implementation: By inheritance

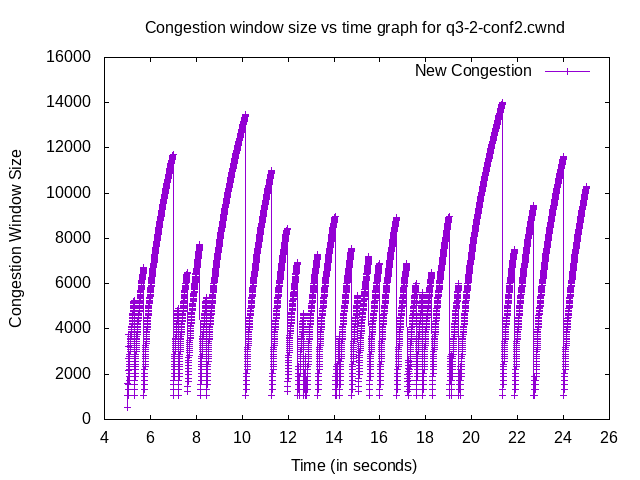
./waf --run "scratch/Third –conf\_num=3"

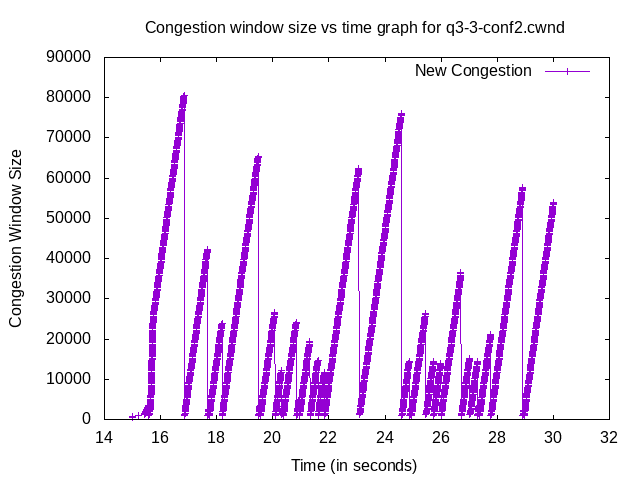




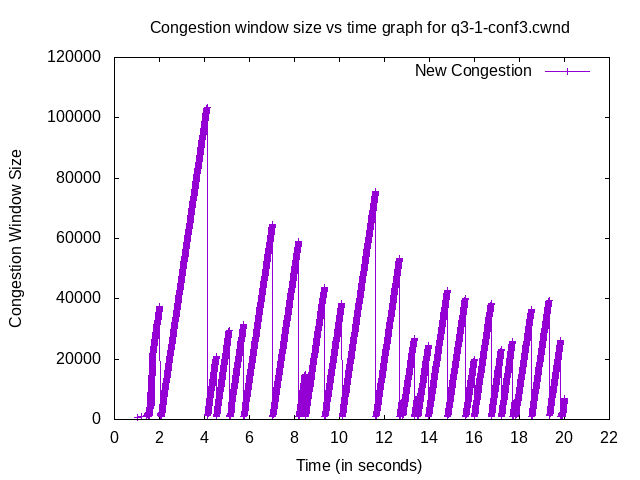
**Configuration 2:**

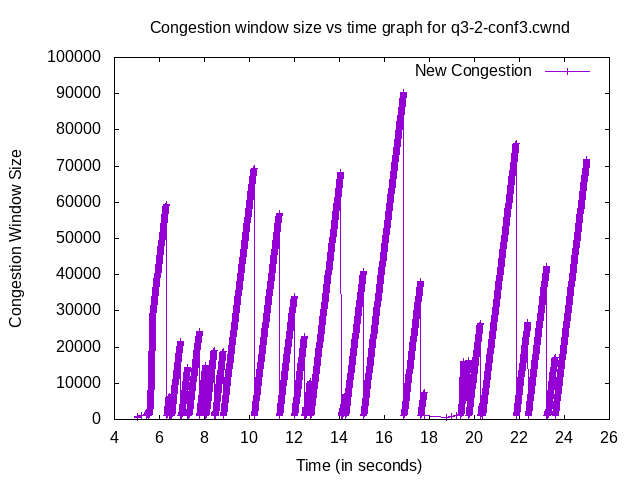


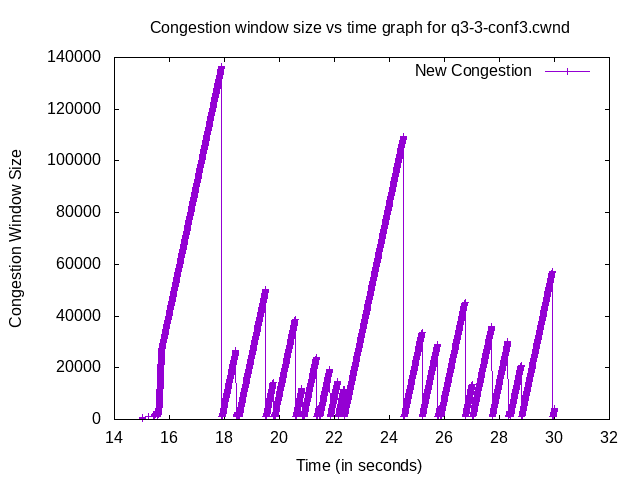




**Configuration 3:**

****

****

****

**Q2.**

Packet Loss

|  |  |  |  |
| --- | --- | --- | --- |
| Config | Channel 1-3 loss | Channel 2-3 loss | Total Loss |
| 1 | 76 | 30 | 106 |
| 2 | 80 | 30 | 110 |
| 3 | 78 | 27 | 105 |

**When connection 3 started to use TCPNewRenoCSE the packet loss in channel 1-3 started to increase. Channel 2-3 loss remained constant as there was not any diference. But when all the nodes used TCPNewReno the overall packet loss decreased. Channel 1-3 got higher number of packet losses but channel 2-3 got reduced number of packet losses.**

**Channel with single connection had lower packet losses than connection with multiple connections.**

**Q3.**

**We see congestion avoidance in TCPNewReno is linear and hence we see a straight line in Conf2 after t=15 seconds where node with new protocol starts whereas we see non linear behaviour in earlier congestion avoidance phases.**

**The overall**