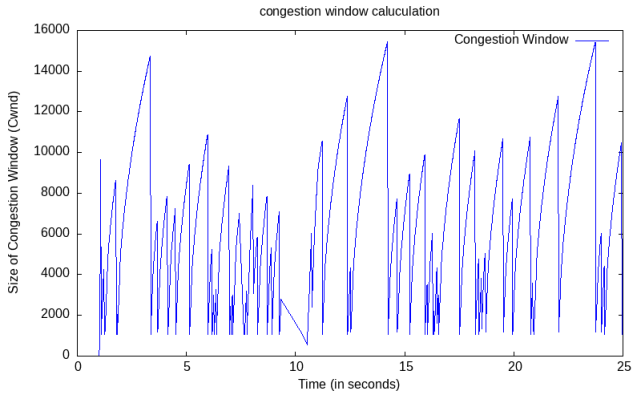
COL 334 Assignment-3 3-11-2022

Madaka Jaya Prakash 2020CS10356

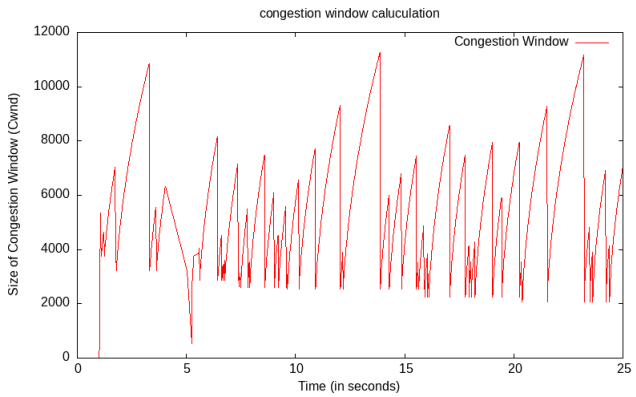
1 Task 1:

Protocol Name	Max Window Size	No. Packet drop
NewReno	15462	58
Vegas	11252	58
Veno	15462	59
WestWood	15471	60

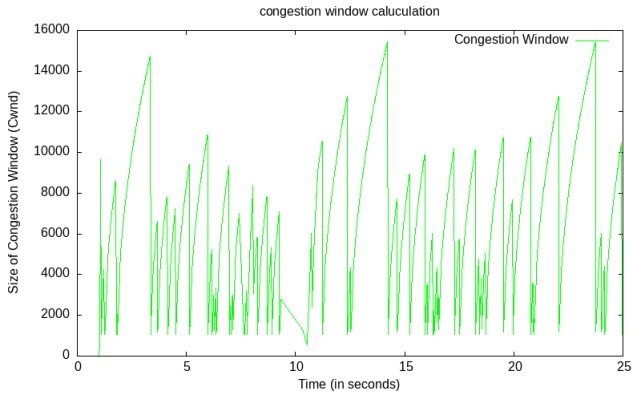
Graph of congestion window for NewReno protocol



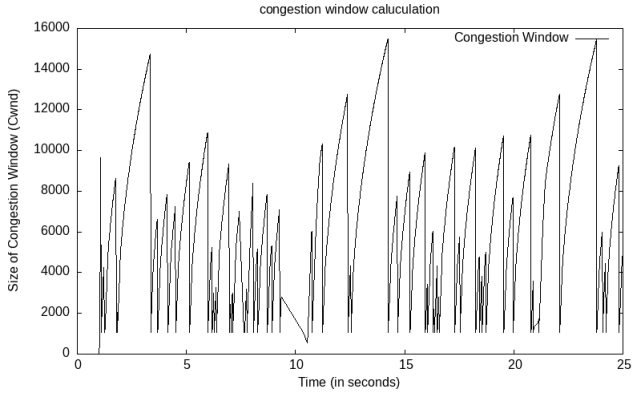
Graph of congestion window for Vegas protocol



Graph of congestion window for Veno protocol



Graph of congestion window for WestWood protocol



TCP NewReno : TCP NewReno is a modified version of TCP Reno congestion protocol. Similar to TCP Reno, TCP NewReno uses a slow start phase with exponential increase in congestion window size.
As we can see it reduces the congestion window size by a large factor when it encounters a packet drop.
TCP Vegas : Compared to all other protocols, the maximum congestion window size is smallest for this protocol.
The reason behind this is TCP Vegas doesn't continue to increase cwnd size till a packet is dropped. Instead, it enters congestion avoidance phase when it encounters lower throughput compared to expected throughput.
So the packet loss is also less compared to other protocols.
TCP Veno : TCP Veno is optimized for wireless networks with larger number of packet drops due to bit errors. For this wired topology, there is no visible difference between TCP Veno and TCP NewReno and they behave almost similarly.
TCP WestWood : TCP Westwood is a sender-side-only modification to TCP New Reno that is intended to better handle large bandwidth-delay product paths, with potential packet loss due to transmission or other errors, and with dynamic load.
Due to only sender-side-only modification TCP WestWood is almost similar to TCP NewReno as observed by graphs.

Bbr Congestion Algorithm :
Bbr refers to Bottleneck Bandwidth and Round-trip propagation time.
Bbr is a new congestion control algorithm developed at google.
All the congestion protocols discussed has primarily used loss-based congestion control, relying only on indications of lost packets as the signal to slow down the sending rate. but Bbr uses latency, instead of lost packets as a primary factor to determine the sending rate.
Latency is the time it takes for data to pass from one point on a network to another.
Bbr is more advantage than other protocols because BBR can cause 100x more packet retransmissions
BBR uses an estimate of the available bottleneck link bandwidth and RTT to govern its pacing rate. BBR has a good performance on Google's internal network and gains widespread attention.
Cwnd of Bbr protocol is caluculated by :

$$\boldsymbol{BDP = BtlBW \times RTprop}$$

where, BDP is Bandwidth Delay Product, and BtlBw isbottleneck bandwidth (BtlBw) and round-trip propagation delay (RTprop).

$$\boldsymbol{Cwnd = G \times BDP}$$

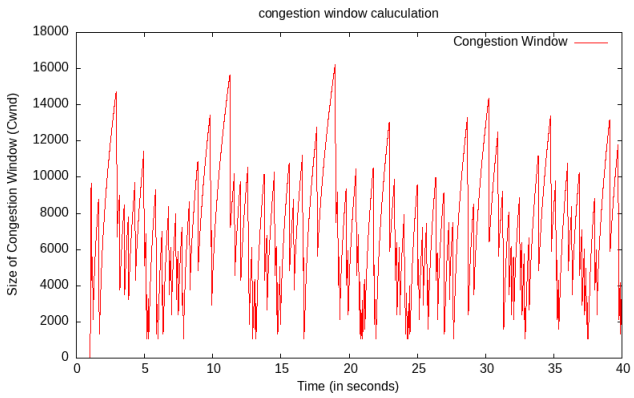
G refers to Gain Coeffcient.
When G > 1, BBR increases the transmission rate; when G < 1, BBR converges to reduce the transmission rate; when G = 1, keeps the current rate to send packets.
When the link is congested, the BBR adjusts the gain coefficient while keeping the BDP unchanged. Once the congestion is relieved, the BBR could be restored to the original state in the first time. The BBR enters Probe RTT phase without a decrease in RTprop within 10s, and the send window is fixed to 4 MSS and the new RTprop value is estimated.

I used existing sixth.cc file in examples to generate all the data and tables and gnuplot for generating all the tables.
Encountered new variables to get max window size and number of packets dropped.

2 Part 2:

2.1

Keeping application data rate as 5Mbps, vary channel data rates (3Mbps, 5Mbps, 10Mbps, 15Mbps, 30Mbps) between N1 and N2.



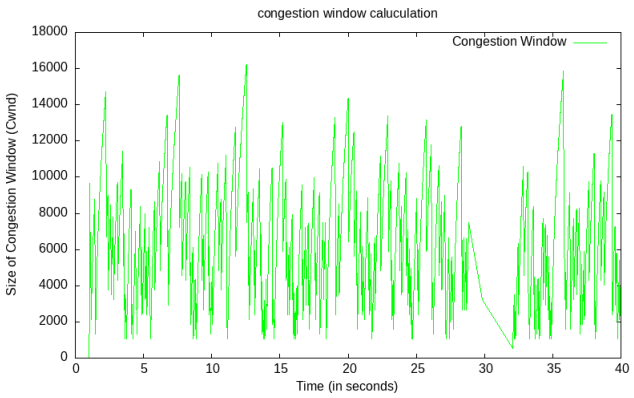


Figure 1: application rate 5Mpbs , channel rate 3Mbps

Figure 2: application rate 5Mpbs , channel rate 5Mbps

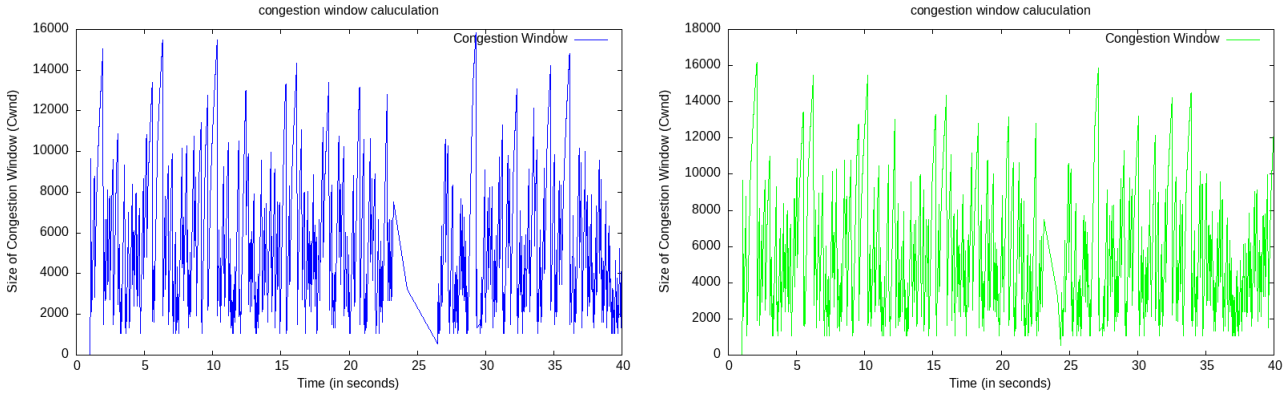


Figure 3: application rate 5Mpbs , channel rate 10Mbps

Figure 4: application rate 5Mpbs , channel rate 15Mbps

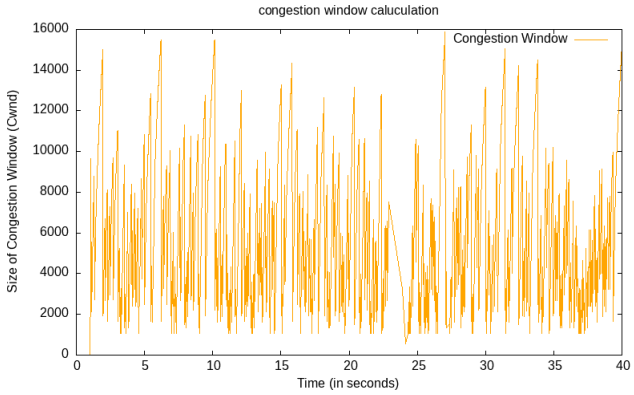


Figure 5: application rate 5Mpbs , channel rate 30Mbps

Observations :

The max window size are 16206,16206,15849,16162,15867

The packet loss count are 131,194,246,260,259

We can see that keeping application data rate as constant and increasing the channel data rate leads to more packet loss.

as more number of packets are transfering.

The plots for channel data rates 15 and 30 are almost same , as they are much larger than application data rate.

For channel rates 3Mbps , 5Mbps the max window size is same , this is because both have channel rates less than application data rates.

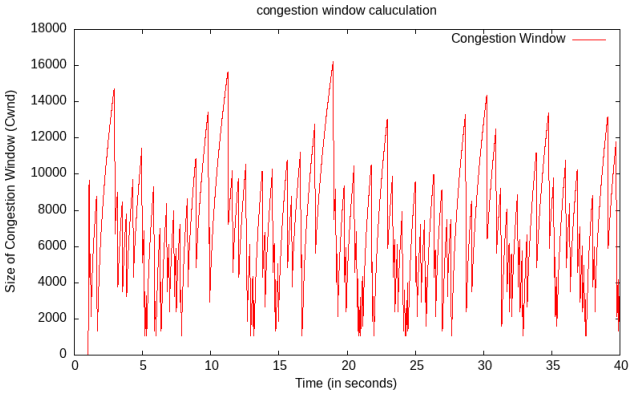
We can see the graph as channel data rate increases the graph of congestion window size is getting more dense and narrowing.

For channel data rate 15Mbps the maximum window size is observerd at 2sec and it is maximum , After that, the peak window size is small which means that the protocol starts at a very large window size and after some packet drops, it

adapts to the topology and the maximum window size is reduced.

2.2

Keeping channel data rate as 4Mbps, vary application data rates (1Mbps, 2Mbps, 4Mbps, 8Mbps, 12Mbps) between N1 and N2.



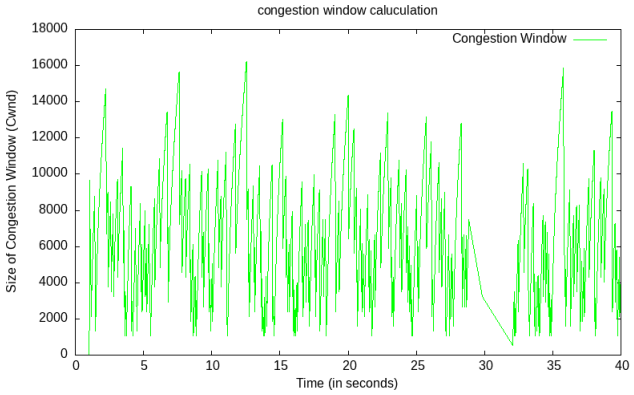
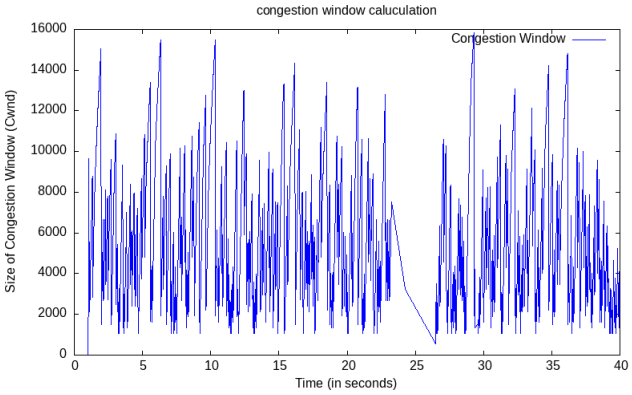


Figure 6: channel rate 4Mpbs , application rate 1Mbps

Figure 7: channel rate 4Mpbs , application rate 2Mbps



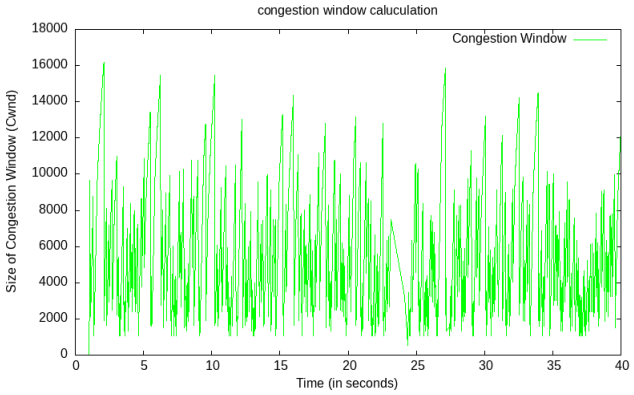


Figure 8: channel rate 4Mpbs , application rate 4Mbps

Figure 9: channel rate 4Mpbs , application rate 8Mbps

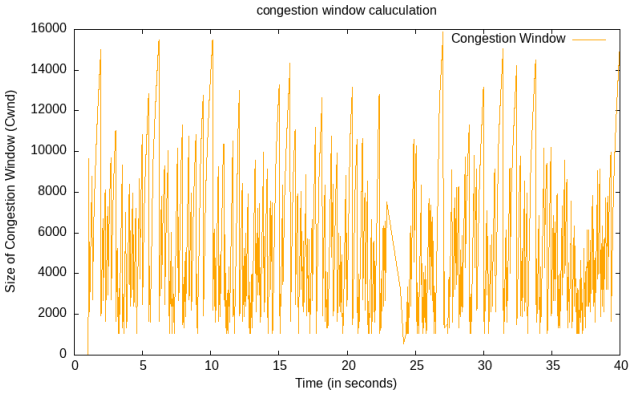
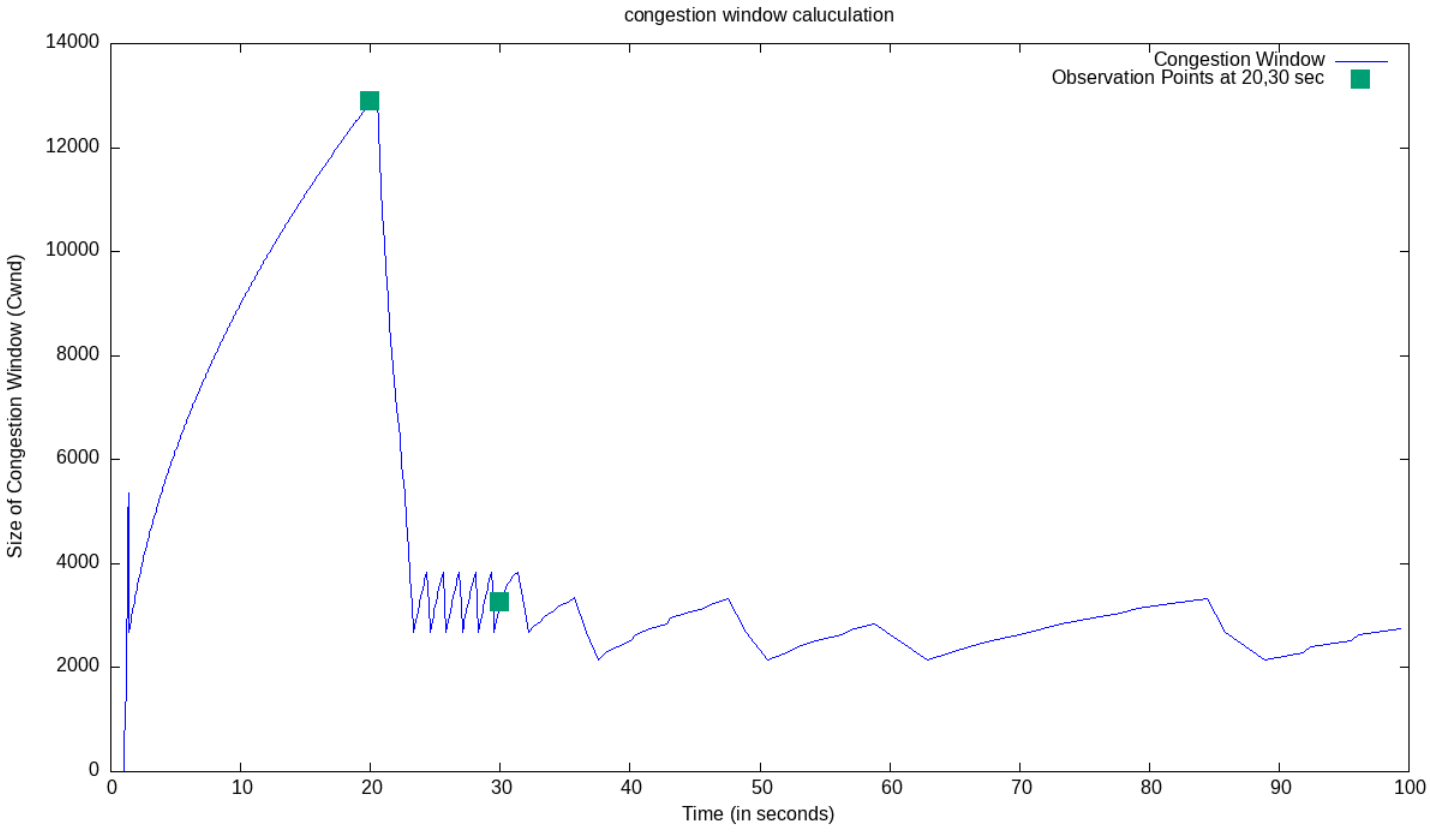


Figure 10: channel rate 4Mpbs , application rate 12Mbps

Observations :
The max window size are 15507 at 25.9 , 27336 at 1.2 , 16206 at 14.8 , 16206 at 14.8 , 16206 at 14.8.
Packet loss count are 51,100,162,162,162.
As we can see the last 3 graphs are identical their max window size , packet drop count are also similar , this is because the application can only send the data at the rate of bottleneck size which is channel data rate in this case. so no matter how much we increase application data rate beyond channel data rate the graphs come similar.
But , increasing the application data rate from 1Mbps to 2Mbps causes the max window size to increase ,and the maximum is at initial stage of the simulation.
As we see packet loss increases as we increase the application data rate as expected if application data rate increases which causes to increase congestion window more fastly which causes more packet loss.

3 Task 3



For Taks 3, I took the reference of simple routing example in examples provides in ns3 and created TCP and UDP sockets and also used fifth.cc file as areference for transmission of UDP and TCP packets.

We can see the sudden drop of congestion window after UDP process starts at 20 seconds.
After 20 seconds at 30 seconds also there is a drop of congestion window size as the UDP arrives 500Kbps application rate which almost clogs the N2-N3 link.
The plot is indicated with the points where the change is observerd.
The pcap file named task3.pcap is submitted along with the report.

For Task 1:
NewReno.cc,NewReno.cwnd,NewReno.pcap
Vegas.cc,Vegas.cwnd,Vegas.pcap.
Veno.cc,Veno.cwnd,Veno.pcap.
WestWood.cc,WestWood.cwnd,Westwood.pcap.

For Task 2:
task2.cc.
task2a1.cwnd,task2a2.cwnd,task2a3.cwnd,task2a4.cwnd,task2a5.cwnd.
task2b1.cwnd,task2b2.cwnd,task2b3.cwnd,task2b4.cwnd,task2b5.cwnd.

For Task 3:
task3.cc.
task3.cwnd,task3.1.pcap,task3.2.pcap,task3.3.pcap,
task3.4.pcap,task3.5.pcap,task3.6.pcaptask3.7.pcap,task3.8.pcap.

These files are submitted along with reprt.