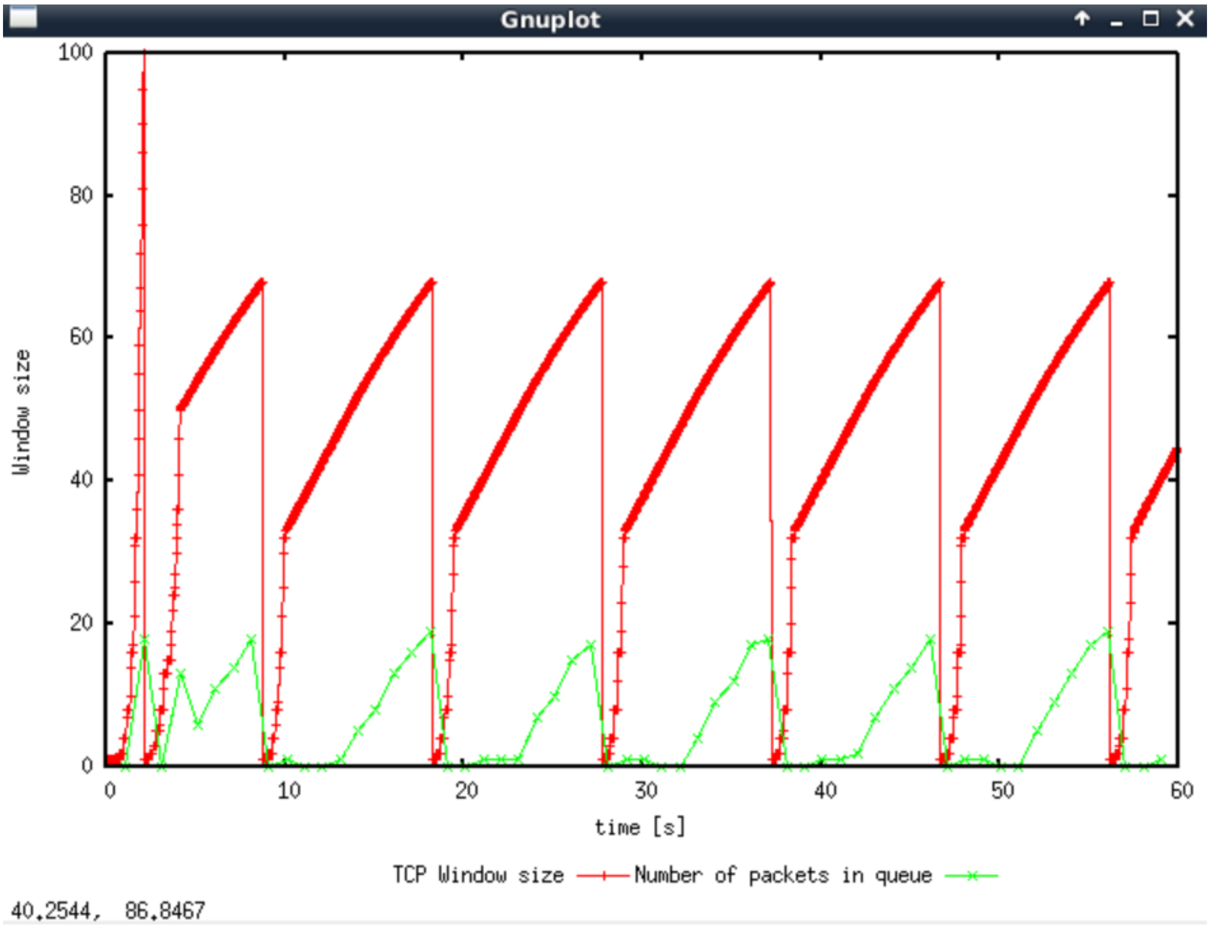
LAB 5

Exercise 1: Understanding TCP Congestion Control using ns-2

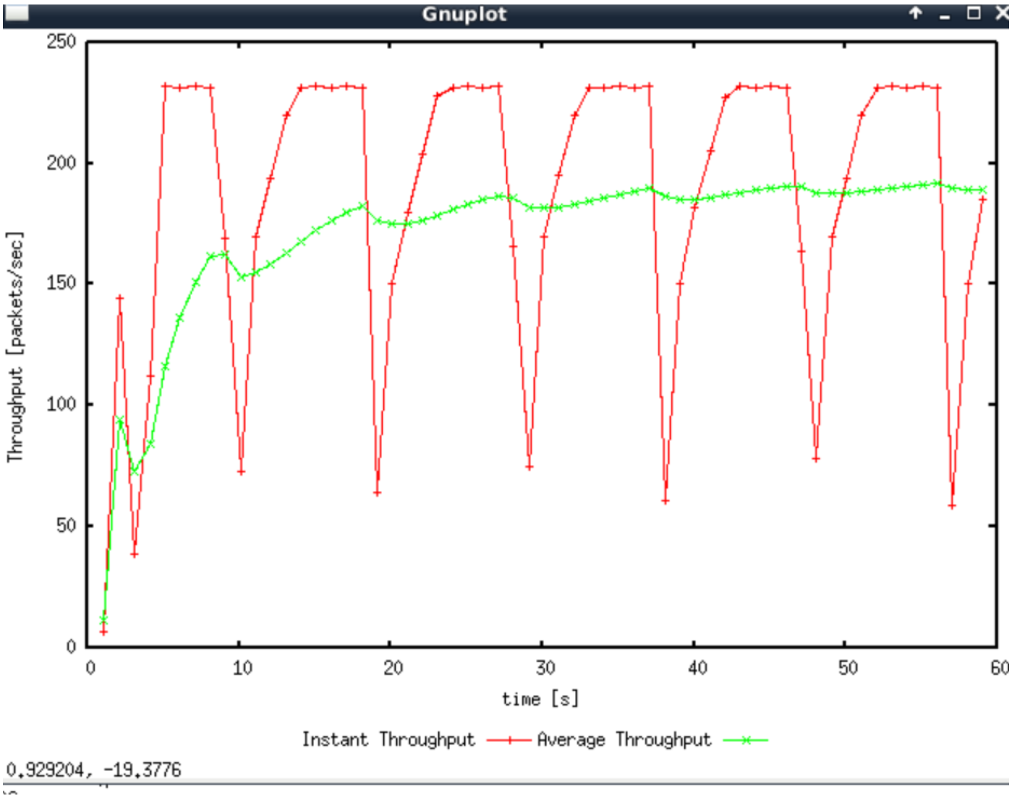
1. **What is the maximum size of the congestion window that the TCP flow reaches in this case? What does the TCP flow do when the congestion window reaches this value? Why? What happens next? Include the graph in your submission report**.



- The maximum congestion goes up to 100 packets with the slow-start mechanism. This is observed at the beginning of the graph
- Some packets are dropped. This is because the congestion of 100 packets is greater than the size of the queue of 20 packets. This causes the queue to be full and the TCP reduces current congestion size to 1MSS.
- The process seems to alternate from the slow-start phase to the packet congestion and dropping of packets phase.

2. **From the simulation script we used, we know that the payload of the packet is 500 Bytes. Keep in mind that the size of the IP and TCP headers is 20 Bytes, each. Neglect any other**

headers. What is the average throughput of TCP in this case? (both in number of packets per second and bps) Include the graph in your submission report.



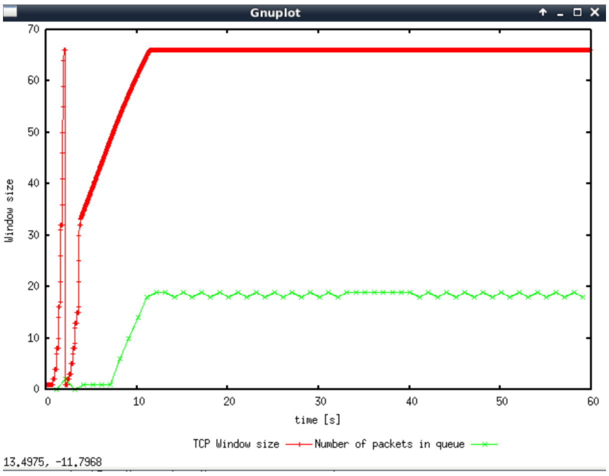
- $Packets\ per\ second\ throughput:\ 190\ pps$
- $Bytes\ per\ second \in throughput:$
  - $IP+TCP\ headers=20+20=40\ bytes$
  - $Payload: 500\ bytes$
  - $Total\ payload: 500+40=540\ bytes$
- $Throughput=Total\ Payload*8\ bits\ (bits\ per\ byte)*pps$

= $540*8*190=820{,}800\ bps$

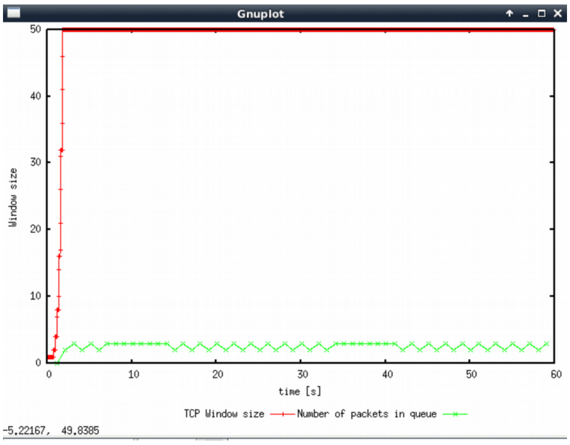
Therefore, average throughput is 820,800 bps

3. **Rerun the above script, each time with different values for the max congestion window size but the same RTT (i.e. 100ms). How does TCP respond to the variation of this parameter? Find the value of the maximum congestion window at which TCP stops oscillating (i.e., does not move up and down again) to reach a stable behaviour. What is the average throughput (in packets and bps) at this point? How does the actual average throughput compare to the link capacity (1Mbps)?**

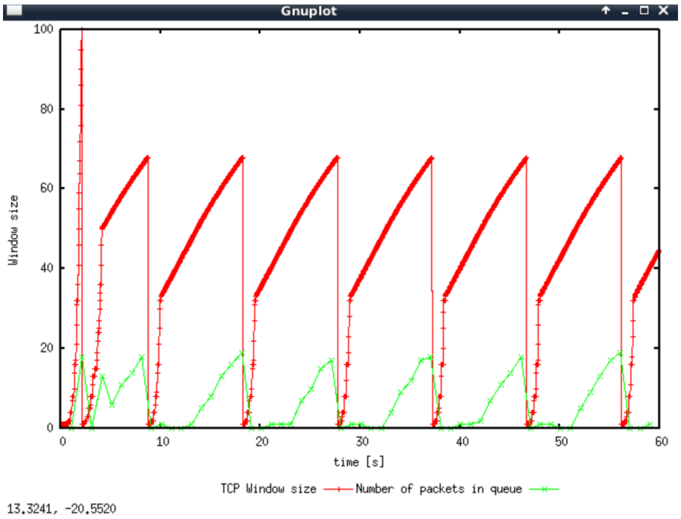
   We can see that when the initial max window size is 66 packets, the oscillations stop after the second slow start.



When the initial max window size is 50 packets(Graph C), the window size stops oscillating when it reaches 50.

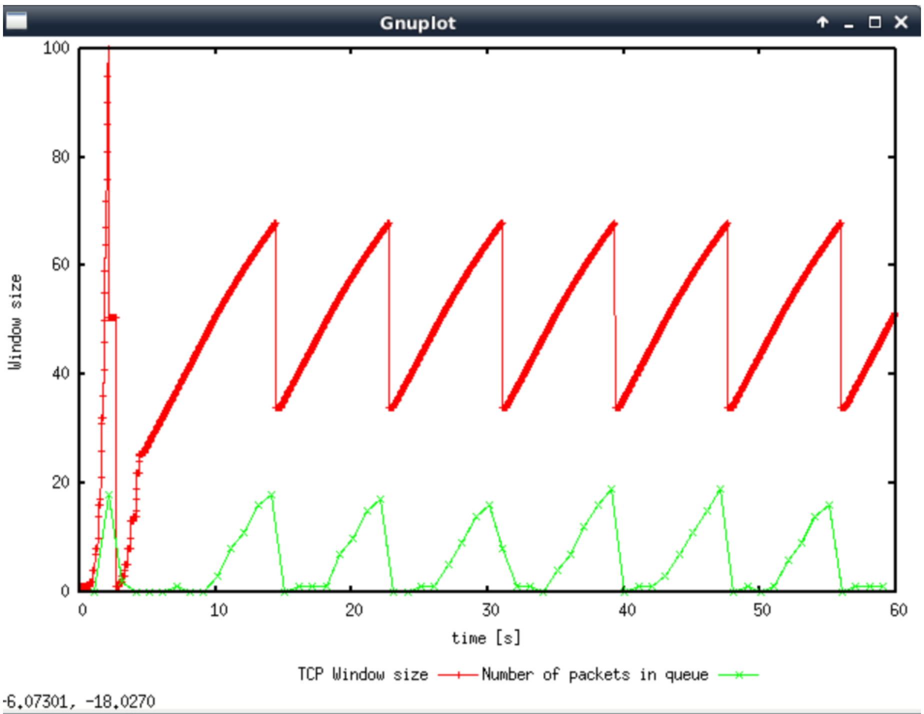


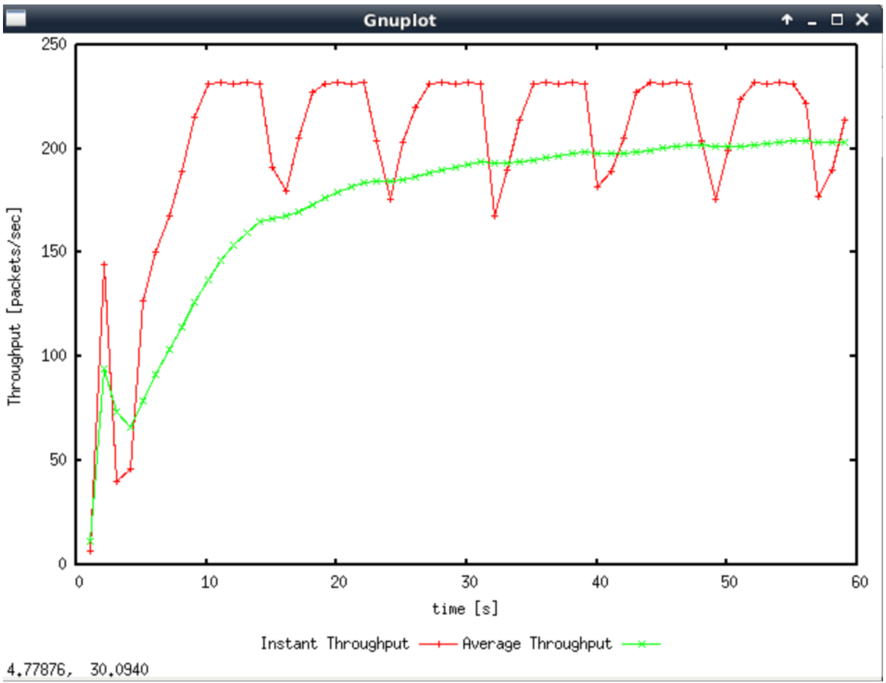
In the graph below, the av the average packet throughput is close to 225 packets per second. This, in bps is 225*8*540 = 97200 bps, which is almost the link capacity of 1Mbps.



4. **Repeat the steps outlined in Question 1 and 2 (NOT Question 3) but for TCP Reno. Compare the graphs for the two implementations and explain the differences. (Hint: compare the number of times the congestion window goes back to zero in each case). How does the average throughput differ in both implementations?**

<u>TCP Reno</u>





As seen from the graphs above, there is no slow-start phase with TCP Reno. It does not reduce the window size to 1MSS when a packet-loss event occurred. Instead, it dropped it window size to half and entered the congestion avoidance phase.

Furthermore, the throughut of TCP Reno is:

TCP Reno pps: 200 pps

$$Throughput = (500 + 40) * 8 * 200 = 820{,}200\, bps$$

As the throughput of TCP Reno is better than that of TCP Tahoe, TCP Reno performs better.

Exercise 3: TCP competing with UDP

1. **How do you expect the TCP flow and the UDP flow to behave if the capacity of the link is 5 Mbps ?**
   I expect the UDP throughput to be higher than the TCP throughput.

2. **How does one flow achieve higher throughput than the other? Try to explain what mechanisms force the two flows to stabilise to the observed throughput.**

   The UDP throughput should be higher than TCP throughput, as it has *no congestion control feature*. This means that it transfers the packet at relatively constant speed regardless of dropping packets during transmission.

   However, TCP has congestion control, and therefore it will detect congestion in the network and reduce the congestion window size (& the sending rate).

TCP’s ‘fast recovery’ can make the TCP throughput relatively stabilized, however, UDP always sends is at a relatively constant speed and we can say it is stabilized per se.

3. **List the advantages and the disadvantages of using UDP instead of TCP for a file transfer, when our connection has to compete with other flows for the same link. What would happen if everybody started using UDP instead of TCP for that same reason?**

Using UDP would not be preferred in a shared link.

As UDP has more aggressive flow rate when compared with TCP, it would be better.

However, the downside is that using UDP can easily cause congestion because of its lack of congestion control mechanism. This would result in a congestive collapse. This is when the network is congested and this results in a large amount of packet loss.