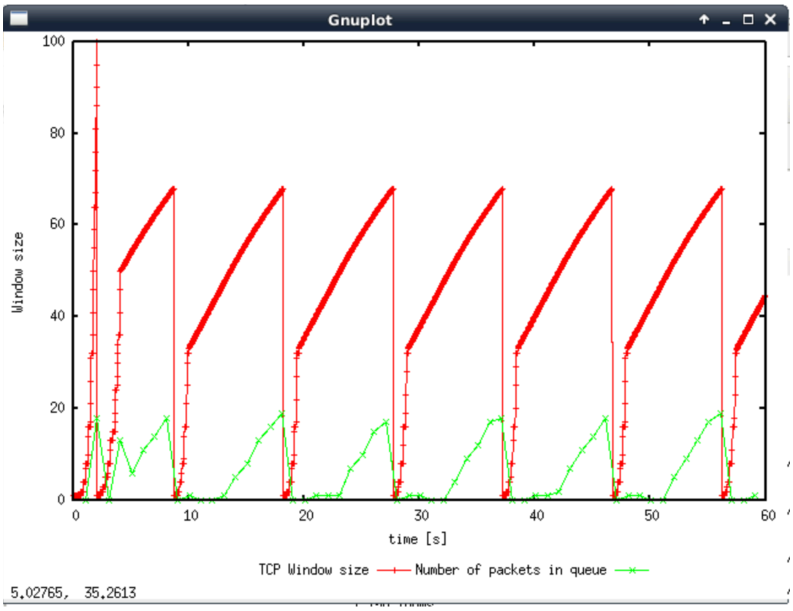
Lab 5: TCP Congestion Control and Fairness/z5146286

Exercise 1: Understanding TCP Congestion Control using ns-2

Q1: 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?



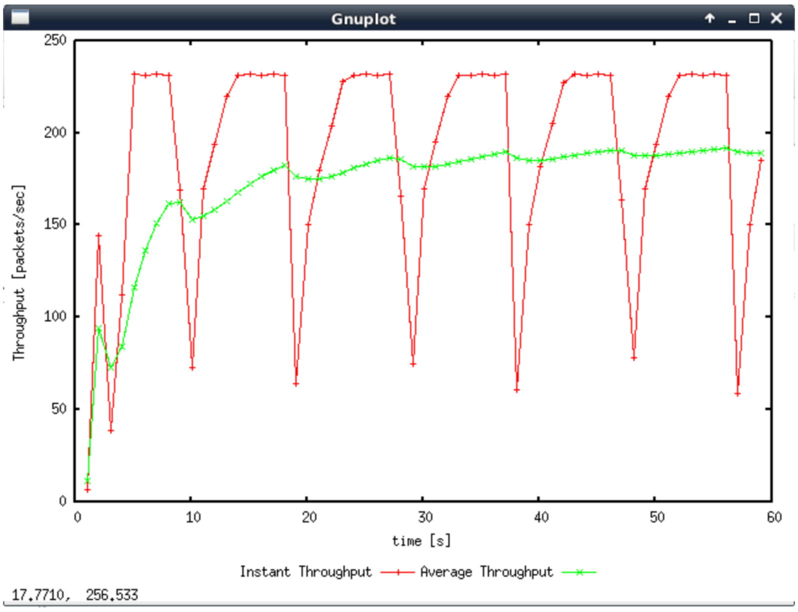
Q: What is the maximum size of the congestion window that the TCP flow reaches in this case?

> **A**: According the graph, the congestion window increases to 100 packets at beginning, as the maximum size of queue is 20 packets, which leads any additional packets to be dropped.

Q: What does the TCP flow do when the congestion window reaches this value? Why? What happens next?

> **A**: When the congestion window reaches this value, the window size will decrease to 1, and the threshold decrease to ½ the size of window. Then, the connection will slowly start until reaching the threshold. Following this the connection transitions to congestion avoidance phase – AIMD. Then the queue become full and congestion happens again. The connection will slowly start again..

Question 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)



A: The green line shows the Average throughput is 190 packets/sec approximately.
The given payload of the packet is 500 Bytes, and the IP and TCP headers is 20 bytes rejectively.
The rate of transmitting **whole data** include payload data and header is :

$$190 * (500 + 20 + 20) * 8 = 820.8 \text{ kbps}$$

The rate of transmitting **useful data** include payload data and header is :

$$190 * 500 * 8 = 760 \text{ kbps}$$

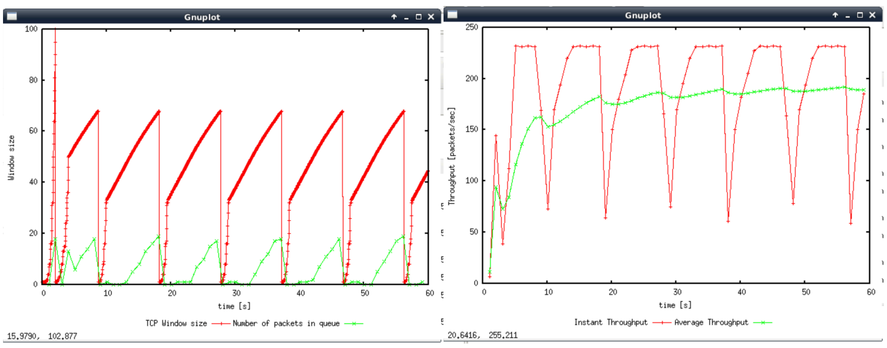
Question 3: Rerun the above script, each time with different values for the max congestion window size but the same RTT (i.e. 100ms). How does TCP respond to the variation of this parameter? Find the value of the maximum congestion window at which TCP stops oscillating (i.e., does not move up and down 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)?

A:

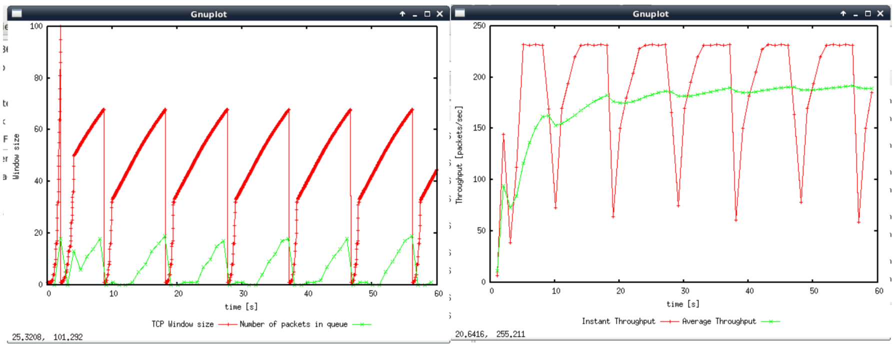
If the maximum congestion window size is set to any size larger than 100, the same graph will be produced. The graph below shows that the initial maximum congestion window <= 50 packets, TCP stabilises immediately. At this point the average packet throughput is approximately 225. The average throughput is 225 * 500 * 8 = 900 kbps, which is almost equal to link capacity.

The initial maximum congestion window <= 66 packets, the oscillations stop after return to slow start. When the congestion window size decreases to ½ size. It is enough to stabilise the number of packets in the sending queue. The queue will never be full. Thus the initial maximum congestion window size should be set to 66.

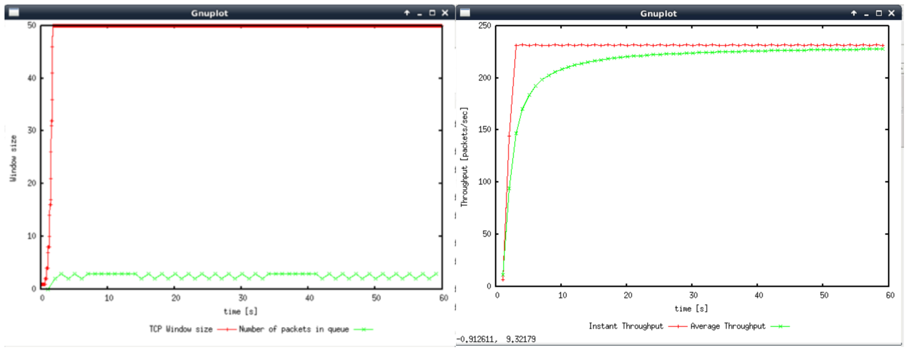
Window size:200



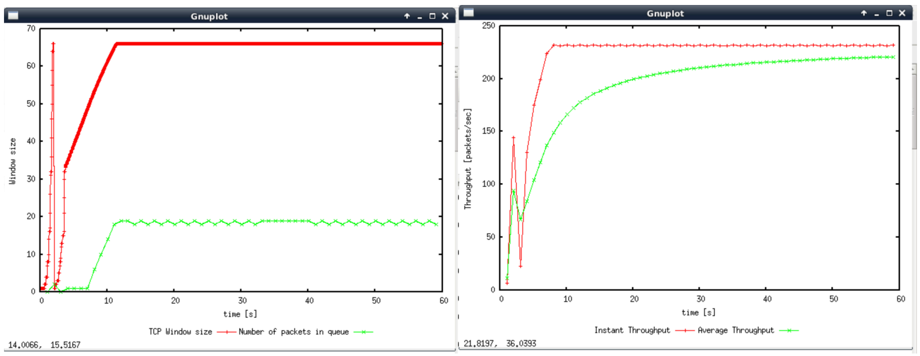
Window size:120



Window size:50



Window size:66



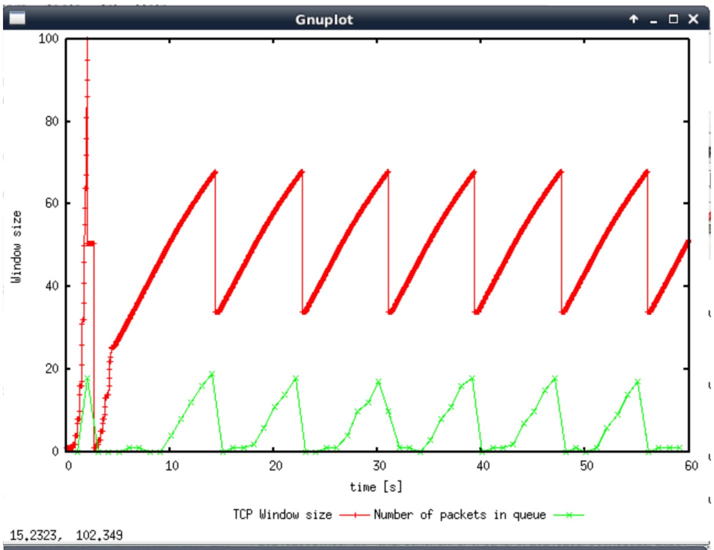
Q4: Repeat the steps outlined in Question 1 and 2 (NOT Question 3) but for TCP Reno. Compare the graphs for the two implementations and explain the differences. (Hint: compare the number of times the congestion window goes back to zero in each case). How does the average throughput differ in both implementations?

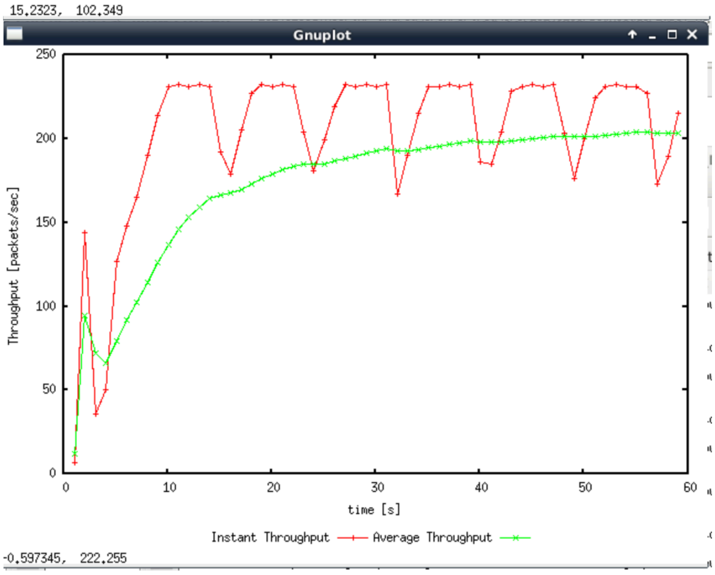
A: The graph below shows, for TCP reno, the TCP connection does not enter slow start and the window size does not be set to 1. Instead, sender halves the congestion windows size, and then increases it linearly.

The average throughout of TCP Reno is almost 200 packets/sec. Thus average throughout of TCP o Reno is:

$$200 * (500 + 20 + 20) * 8 = 864 \text{ kbps}$$

It is higher than TCP Tahoe, because TCP Reno does not have to enter slow start.

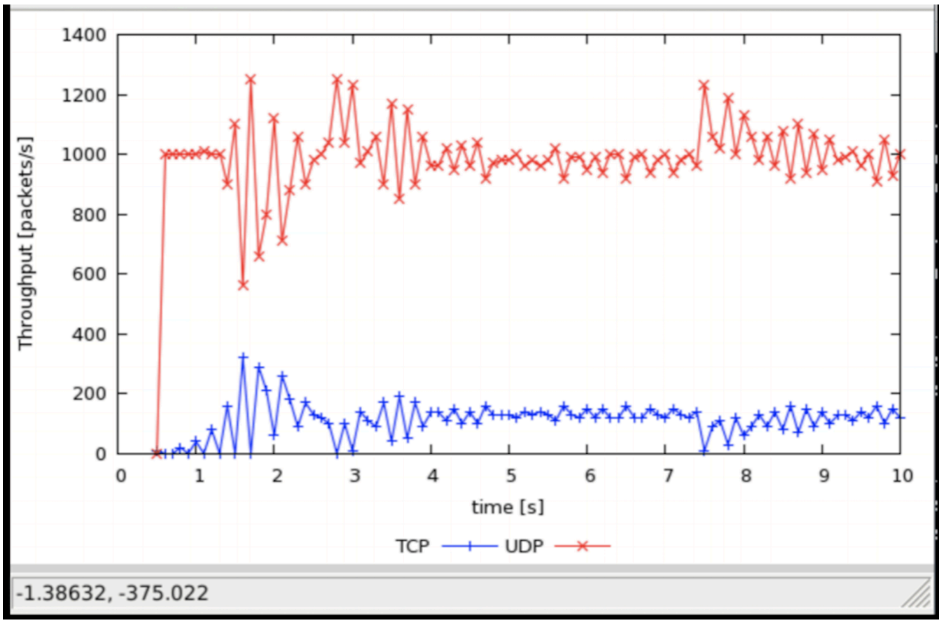




Exercise 3: TCP competing with UDP

Question 1: How do you expect the TCP flow and the UDP flow to behave if the capacity of the link is 5 Mbps ?

A: UDP transmission is faster than TCP transmission, because UDP flow does not decrease the transmission rate to response to congestion. As the graph below shows that the red line is UDP flow and the blue line is TCP flow.



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

A: The UDP throughput is higher than TCP throughput, because UDP flow does not be restrict by congestion control mechanism.

Question 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?

A:

Advantage of UDP used to transfer files: higher average throughout, irrespective of congestion, lower delay for transferring.

Disadvantage of UDP used to transfer files: UDP does not provide reliable data transfer, and does not have congestion control.