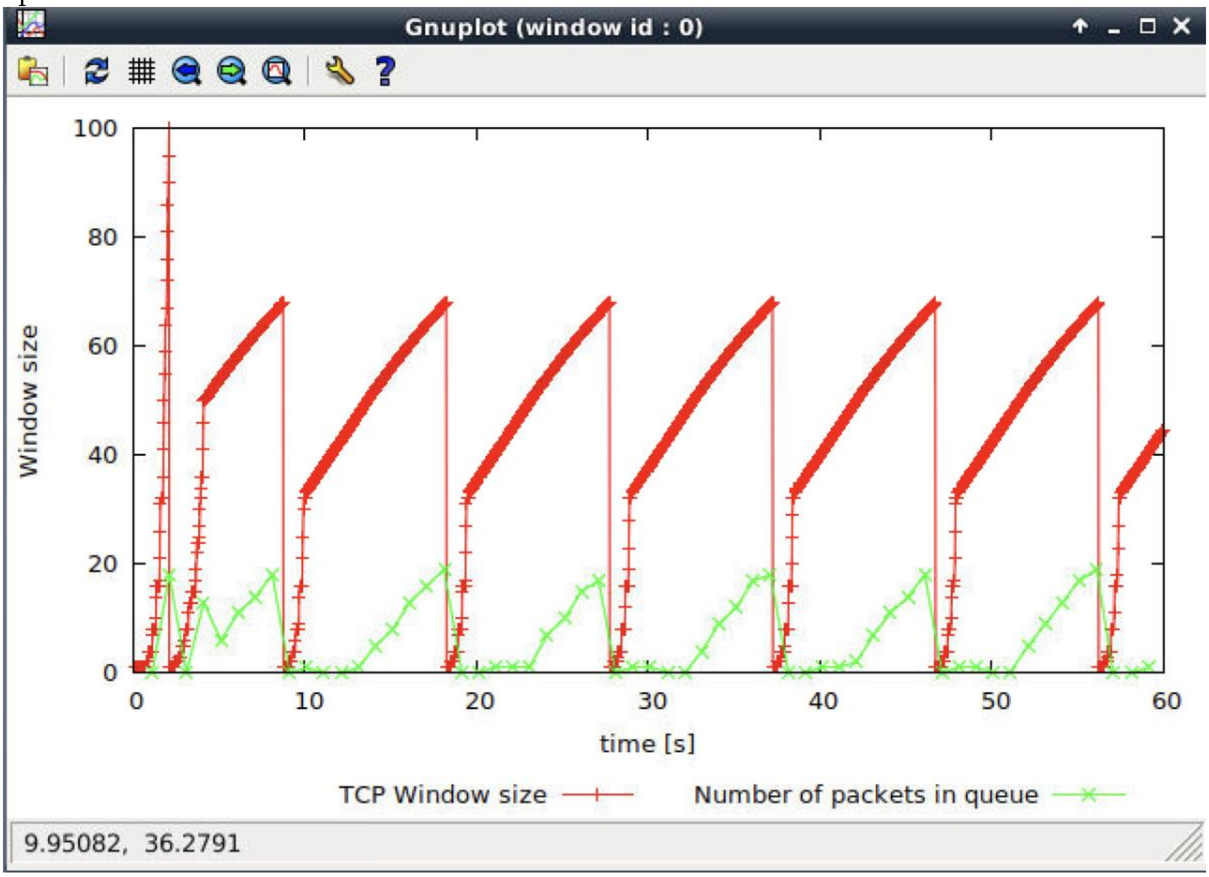
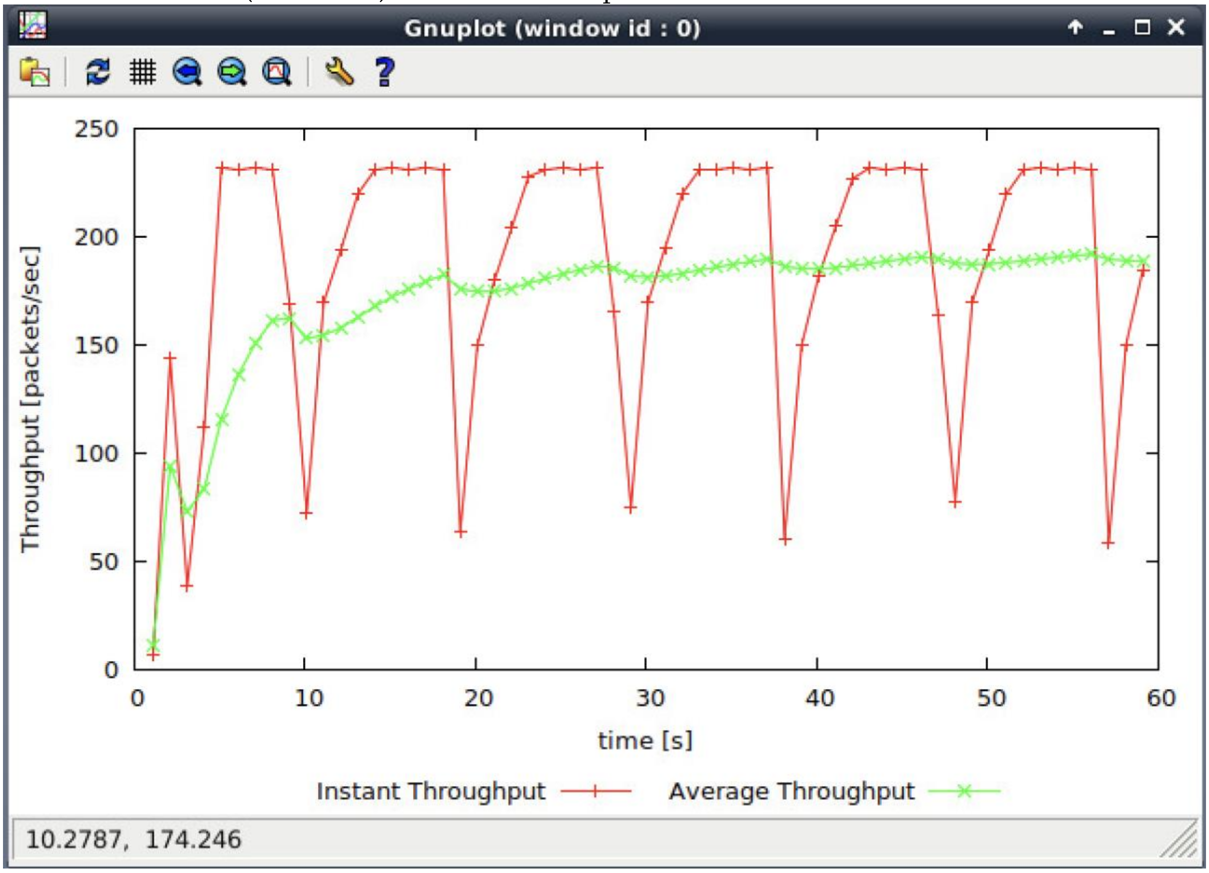
Comp9331 Lab5
StudentID: z5184142
StudentName: JiachenLi

Exercise 1: Understanding TCP Congestion Control using ns-2

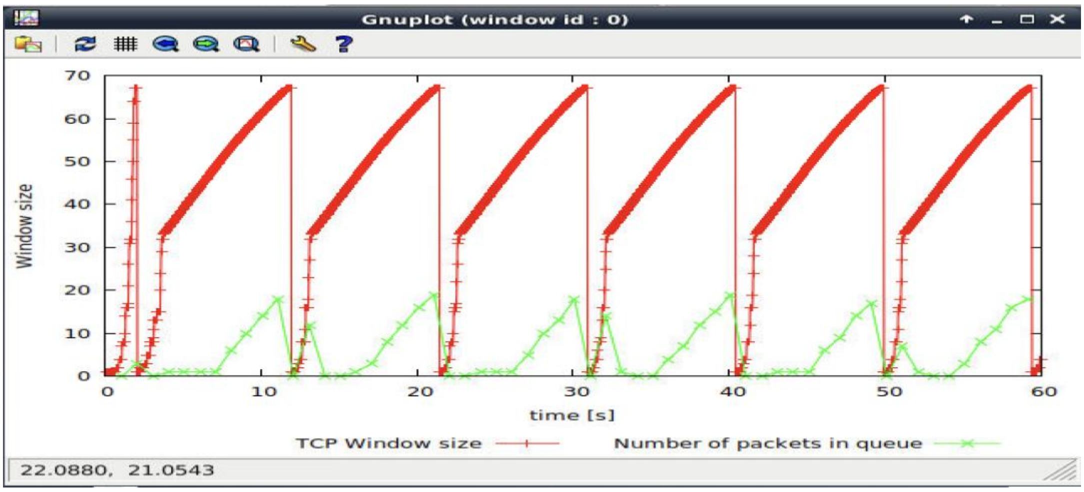
Question 1: What is the maximum size of the congestion window that the TCP flow reaches in this case? What does the TCP flow do when the congestion window reaches this value? Why? What happens next? Include the graph in your submission report.
Answer:
The maximum size of the congestion window that the TCP flow reaches in this case is 100.
When the congestion window reaches this value, the TCP flow reduces the congestion window size to 1 and threshold to ½ the size of the window, because the queue and the link is full and if more packets are going to the window size, it will make packet loss. Next the connection enters slow start and ramps up until it reaches the threshold.



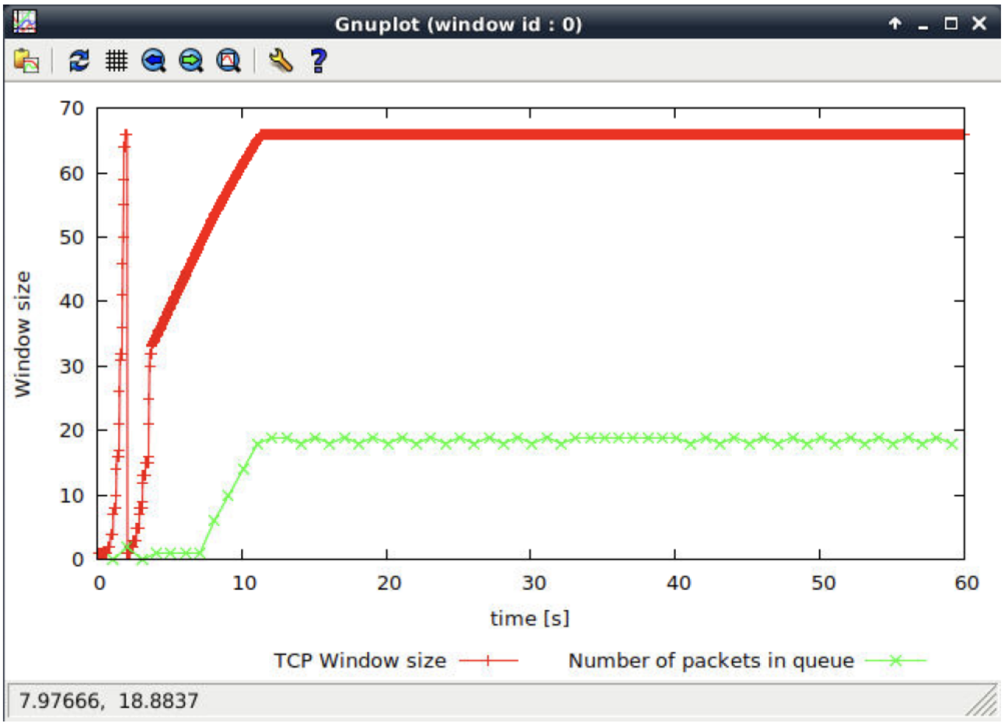
Question 2: What is the average throughput of TCP in this case?
Answer:
The average throughput of TCP in this case is around 190 packets per second.
Calculation formula (500+20+20) * 190 * 8= 802.8kbps



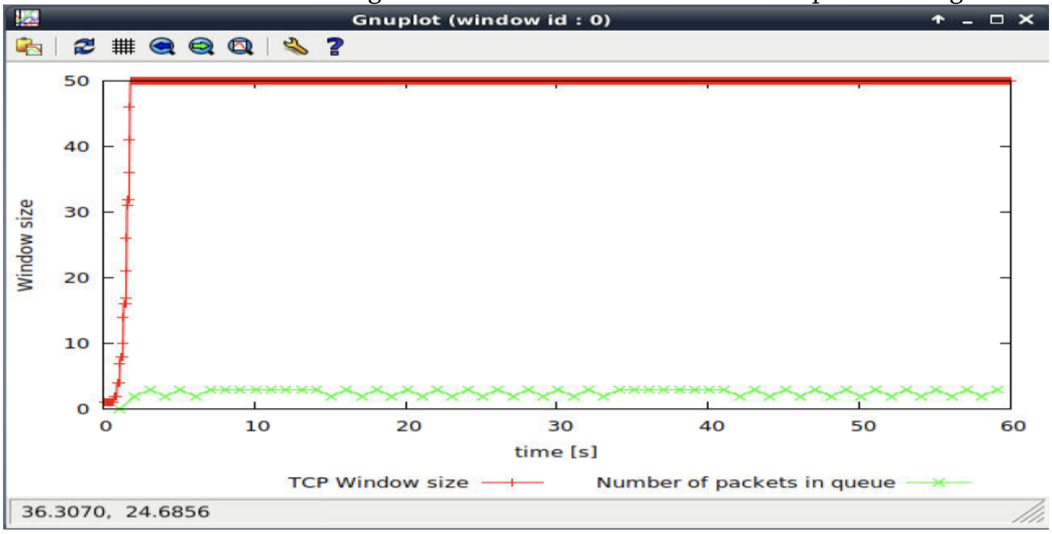
Question 3: Rerun the above script, each time with different values for the max congestion window size but the same RTT (i.e. 100ms). How does TCP respond to the variation of this parameter? Find the value of the maximum congestion window at which TCP stops oscillating (i.e., does not move up and down 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)?
Answer:
When the variation of this parameter is over 66, TCP repeat responses in reducing the window size to 1 and will go back to slow start.



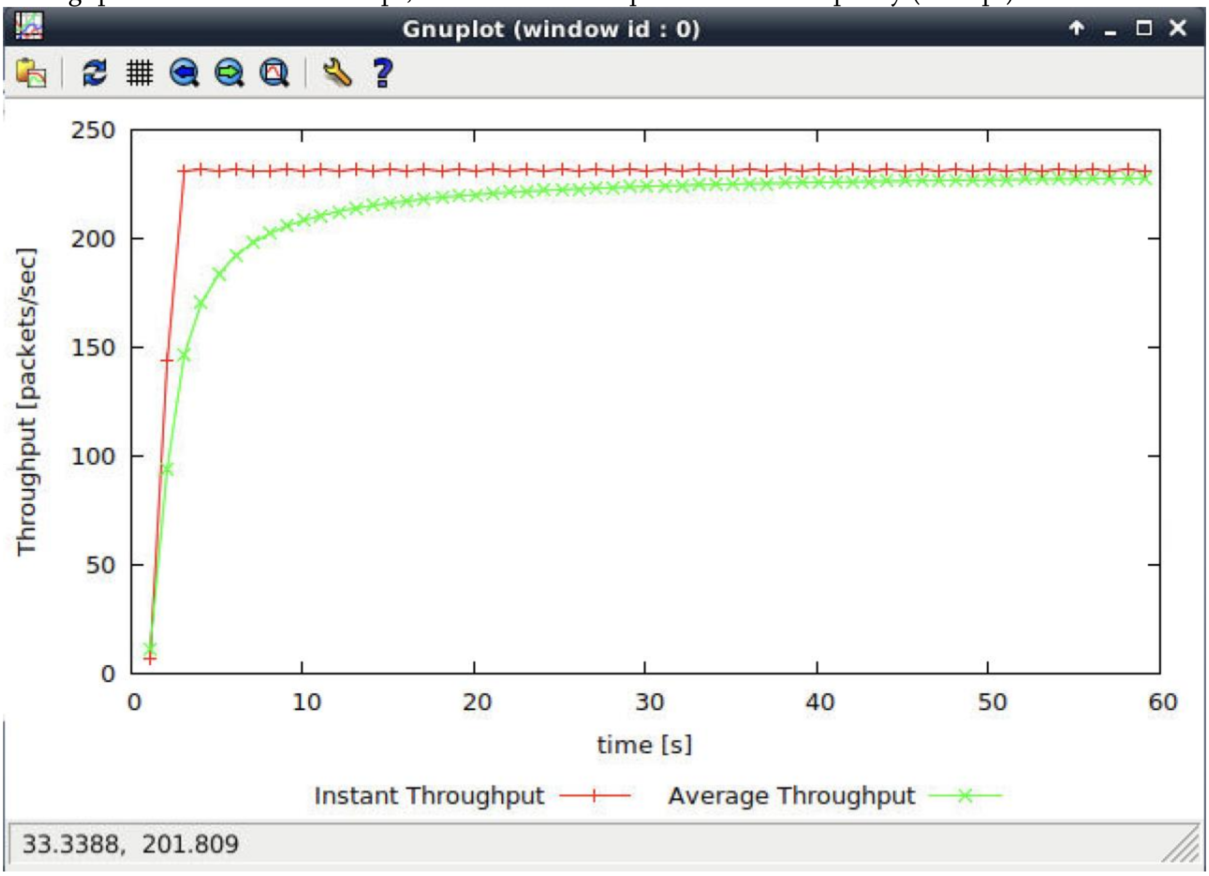
When the variation of this parameter is between 50 to 66, TCP responses in reducing the window size and will go back to slow start, but it will not repeat this progress.



The value of the maximum congestion windows is 50 which TCP stops oscillating.



At this point, the average packet throughput is around 225 packets per second. The average throughput is 225*500*8=900 Kbps, which is almost equal to the link capacity (1 Mbps)

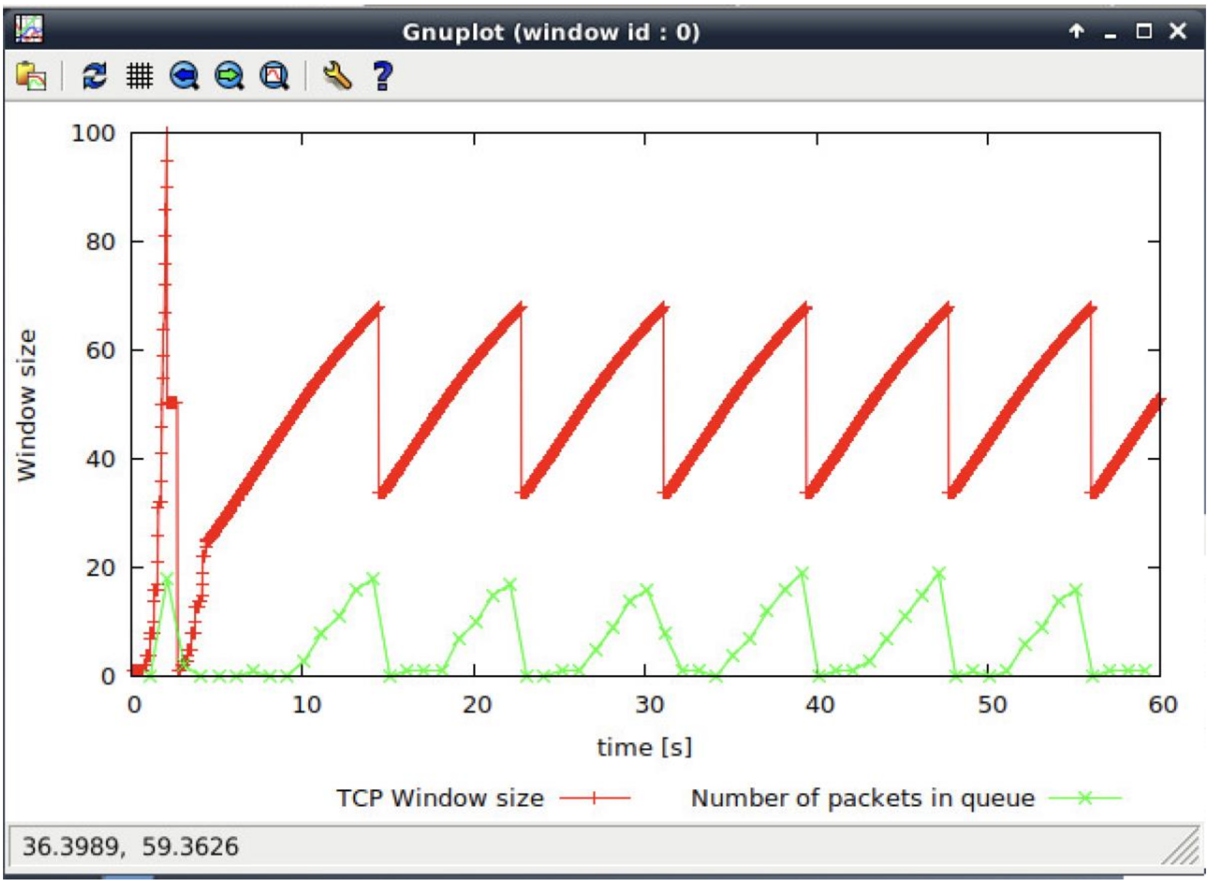


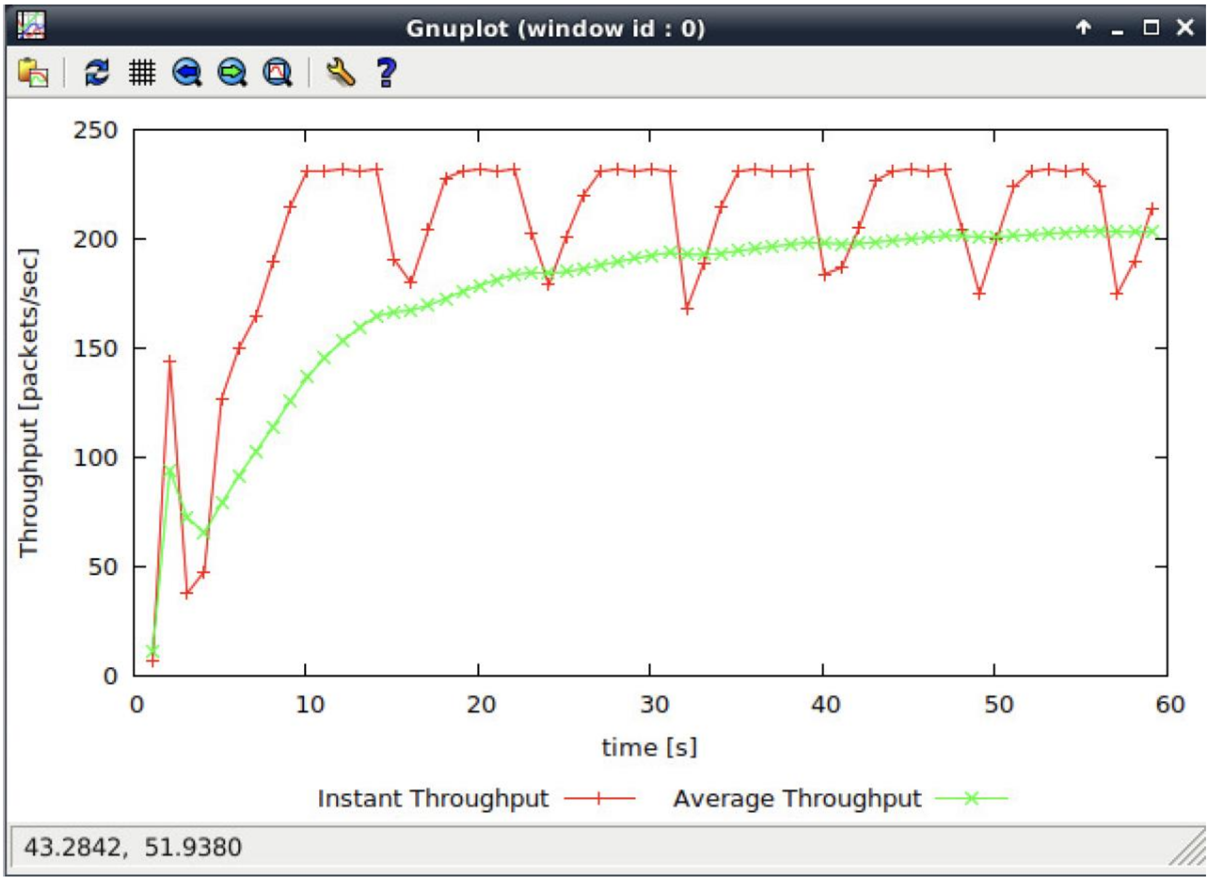
Question 4: Repeat the steps outlined in Question 1 and 2 (NOT Question 3) but for TCP Reno. Compare the graphs for the two implementations and explain the differences. (Hint: compare the number of times the congestion window goes back to zero in each case). How does the average throughput differ in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

Answer:

From the graph below it can be seen that, at the first time TCP flow congestion drops to 1 window size, but after that the sender just cut the half windows size and increase linearly repeatedly until packet loss starts again. However, TCP Tahoe will always make the TCP flow congestion windows size to 1 if there are packet losses.

The throughput of TCP Reno is around 200 packets per second which is a little bit higher than TCP Tahoe. This may because of that TCP Reno does not need to initiate slow start after each congestion event.

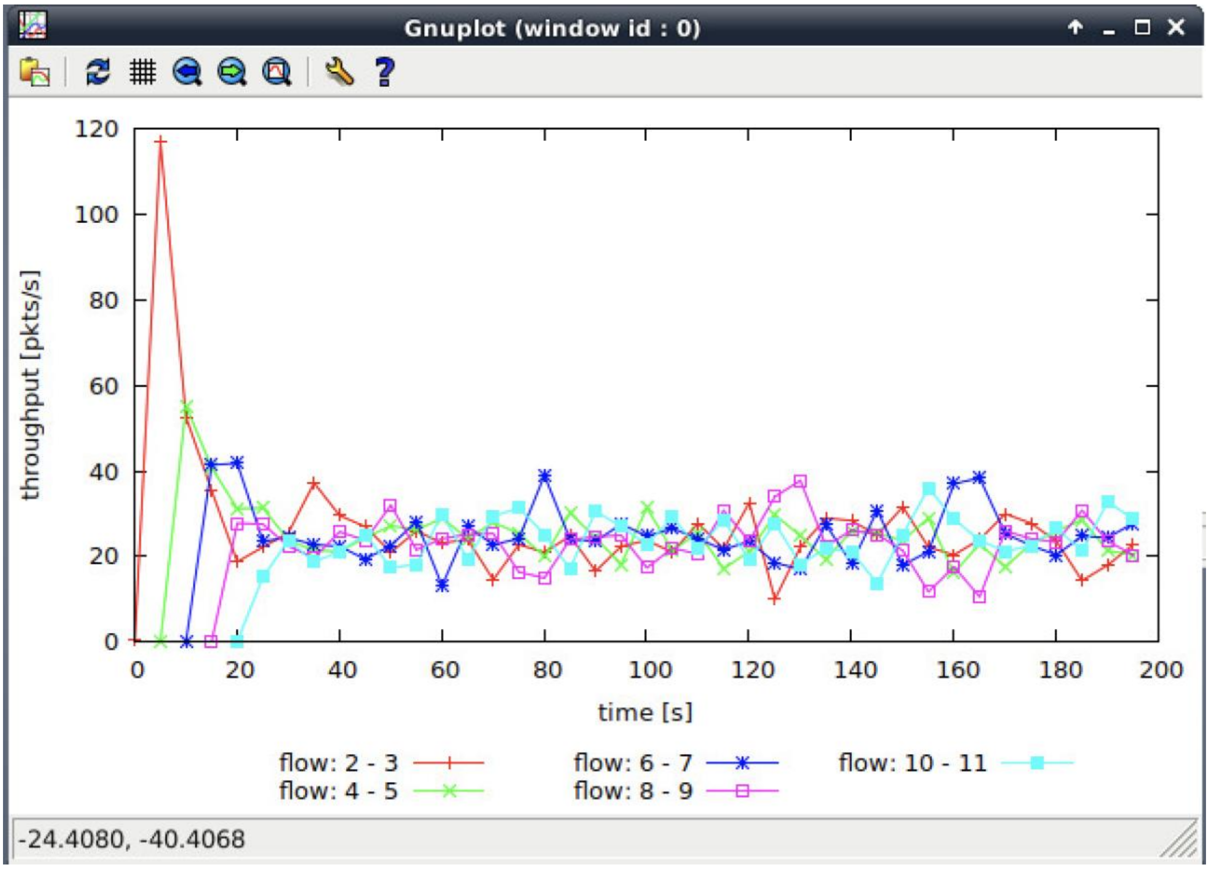




Exercise 2: Flow Fairness with TCP

Question 1: Does each flow get an equal share of the capacity of the common link (i.e., is TCP fair) ? Explain which observations lead you to this conclusion.
Answer:
Yes, when time comes to 20seconds, the throughput for all connection is almost similar. When the flows share one link, they experience the same network condition, so they will react in the same way. It is a result of the tcp AIMD congestion control.

Question 2. What happens to the throughput of the pre-existing TCP flows when a new flow is created? Explain the mechanisms of TCP which contribute to this behavior. Argue about whether you consider this behavior to be fair or unfair.
Answer:
When a new flow is created, the throughput of the pre-existing TCP flows will decrease, and this is because of the fairness mechanisms. When a new flow is created and added to the link, the pre-existing flows will release some bandwidth for the new one. I think this behavior is fair in normal, because when the new flow needs the bandwidth and the pre-existing one should give the resources instead of not sharing it. However, if the new flow is very huge and it is created by parallel TCP connection, then it would take most resources, which is unfair to the pre-existing flow.

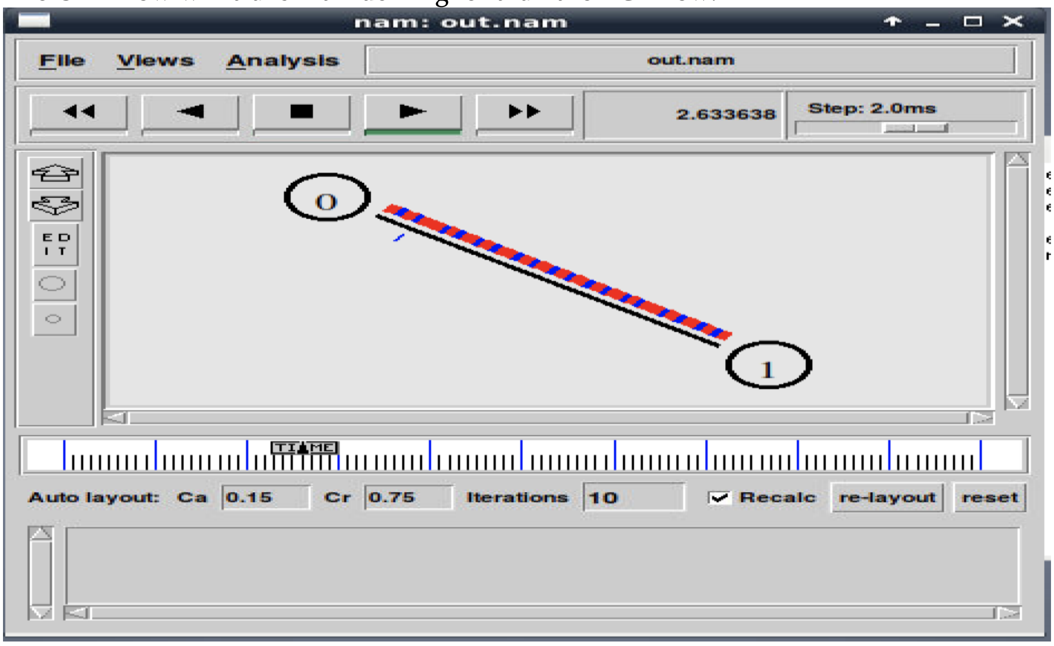


Exercise 3: TCP competing with UDP

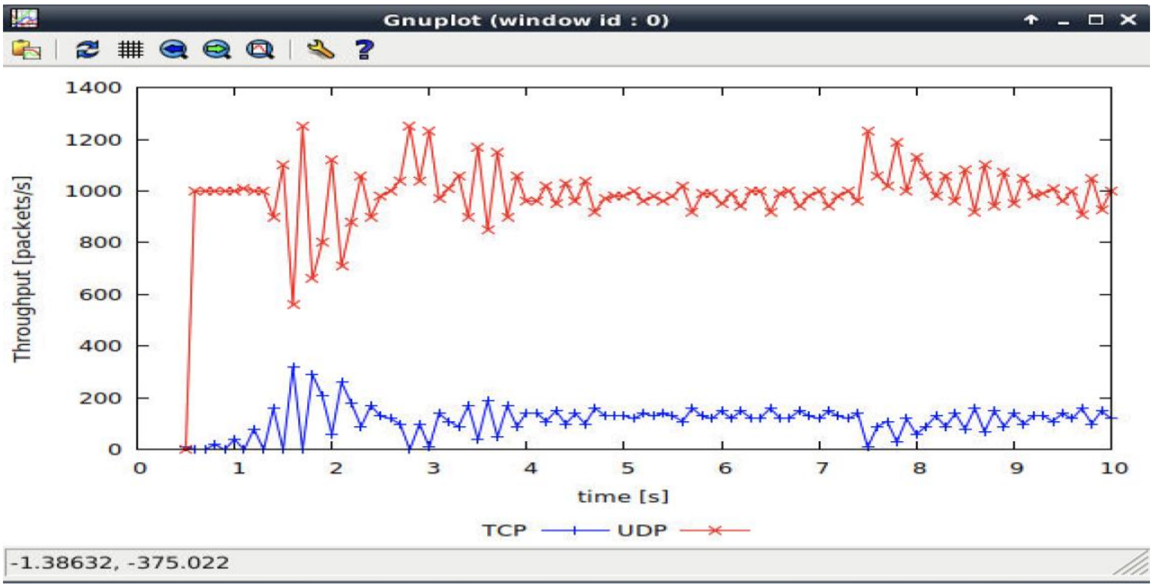
Question 1: How do you expect the TCP flow and the UDP flow to behave if the capacity of the link is 5 Mbps ?

Answer:

The UDP flow will transmit much higher than the 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.
Answer:
Because UDP has no congestion control, so the throughput will most be taken by the UDP and Another reason is that UDP do not has the connection setup process, so the speed of data transmission in UDP is faster than TCP.



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?
Answer:
The advantage of using UDP for a file transfer is that the transmission speed is much faster than using TCP, especially when transmitting the small size of files because UDP do not need to have connection setup. In addition, UDP do not have the congestion control, which means it could take most bandwidth while compete with other flows for the same link. However, UDP could not guarantee that the data could be sent to the receiver successfully, if everybody using UDP instead of TCP, then whether the data could be transmitted without loss will depend on luck.