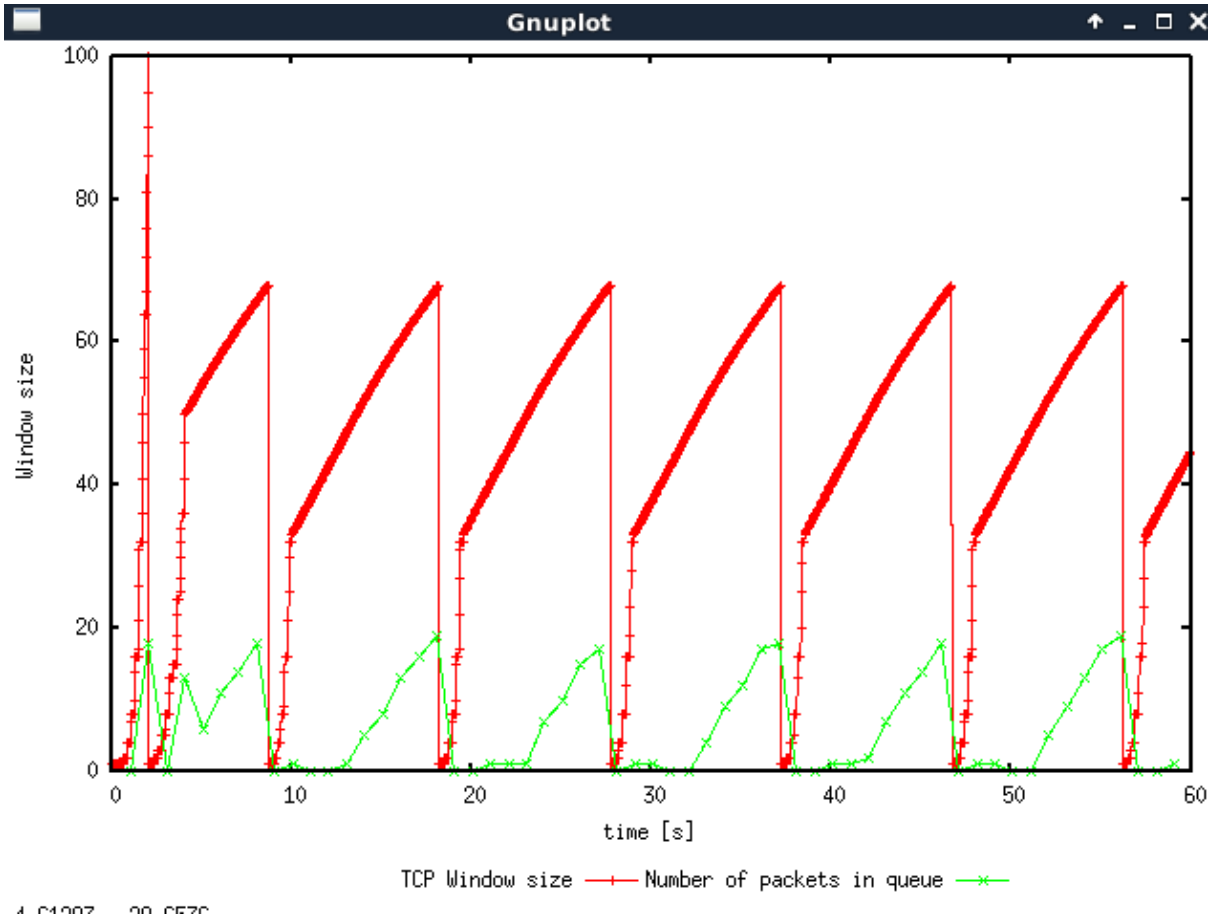
Lab Exercise 5: TCP Congestion Control and Fairness

Exercise 1: Understanding TCP Congestion Control using ns-2

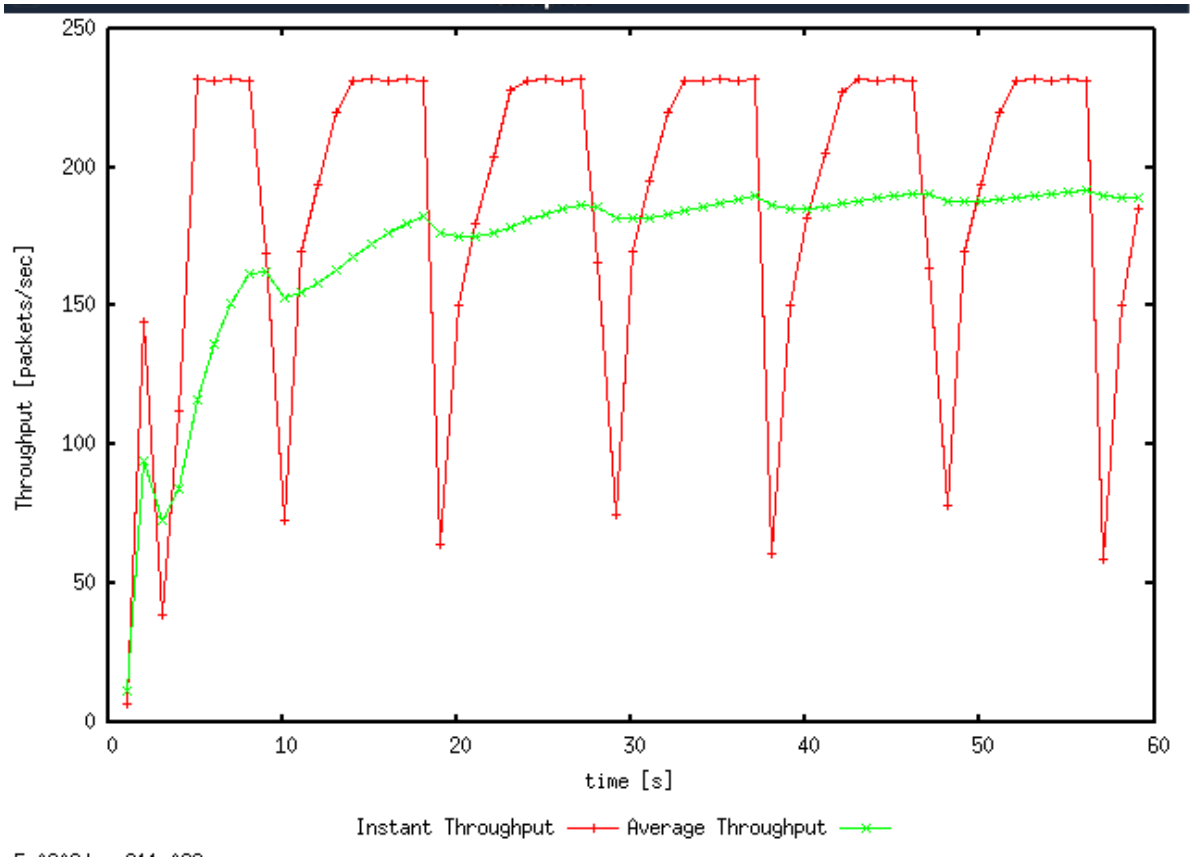
1. What is the maximum size of the congestion window that the TCP flow reaches in this case? What does the TCP flow do when the congestion window reaches this value? Why? What happens next? Include the graph in your submission report.

The max size of the congestion window that the TCP flow reaches is 100. When it reaches this value, the size is set back to zero since a congestion event(timeout) occurs. After that, it slow-starts until it reaches half of the previous value of congestion window - which is 50, then the TCP sender stop slow-start and change to the AIMD algorithm.



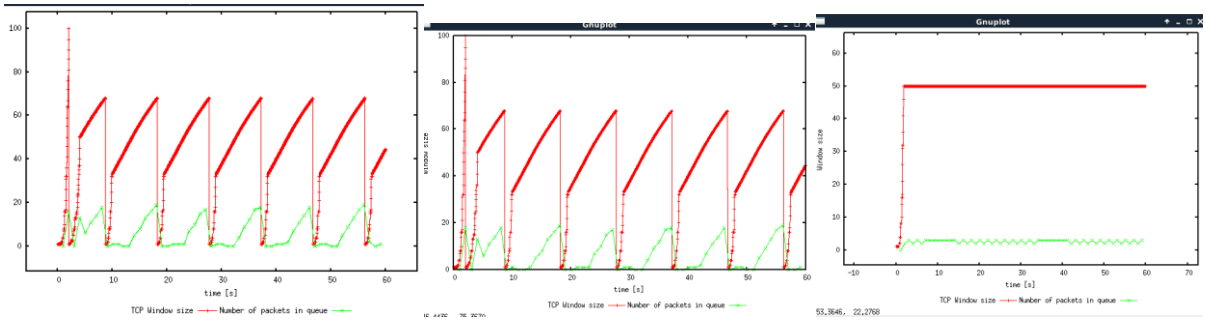
Question 2: From the simulation script we used, we know that the payload of the packet is 500 Bytes. Keep in mind that the size of the IP and TCP headers is 20 Bytes, each. Neglect any other headers. What is the average throughput of TCP in this case? (both in number of packets per second and bps)

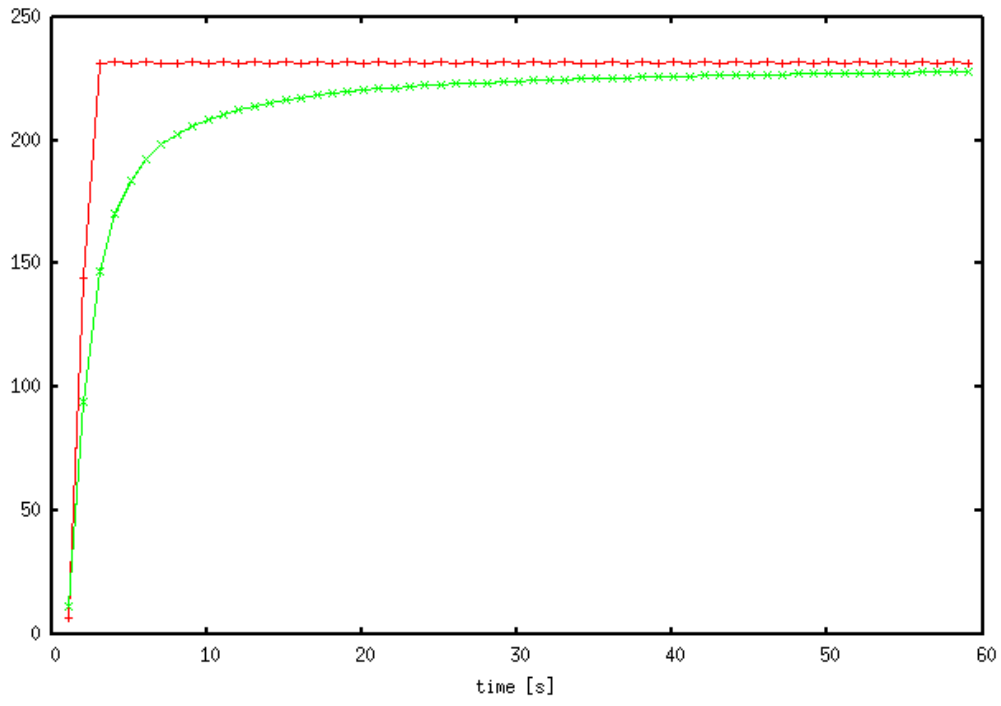
The average throughput of TCP is roughly 180 packets per second, and each packet contain 540 bytes include the payload and the headers. Thus, the average throughput in bps is 540 * 180 * 8 = 777600 bps



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)?

Window size: 100 150 50



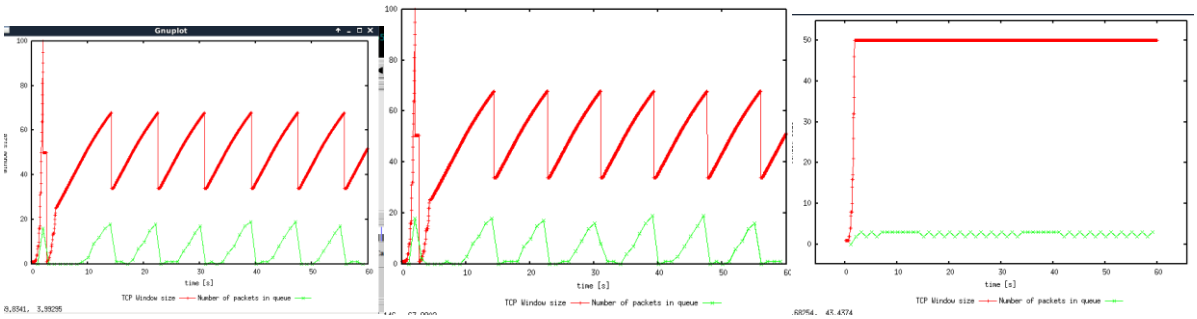


TCP stops oscillating at the window size of 50, the average throughput is around 200 packets per second which is 864000bps. The actual average throughput is thus less than the link capacity 1Mbps.

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?

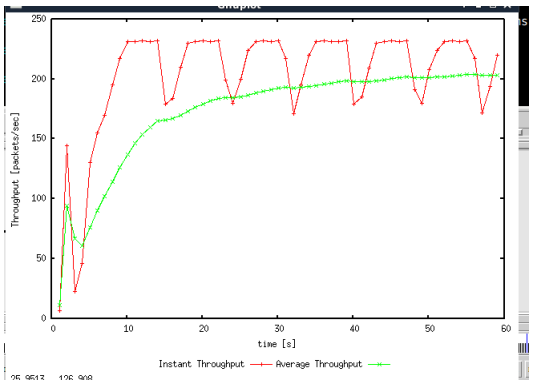
Window size:

100 **150** **50**

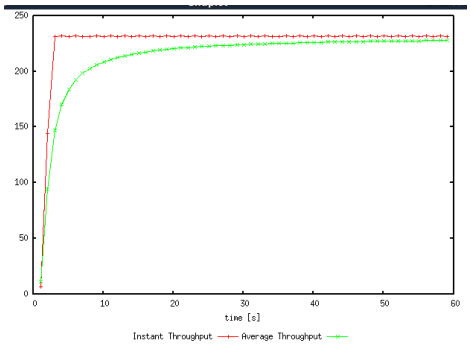


Window size 100: avg throughput

190 packet per second



window size 50: avg throughput



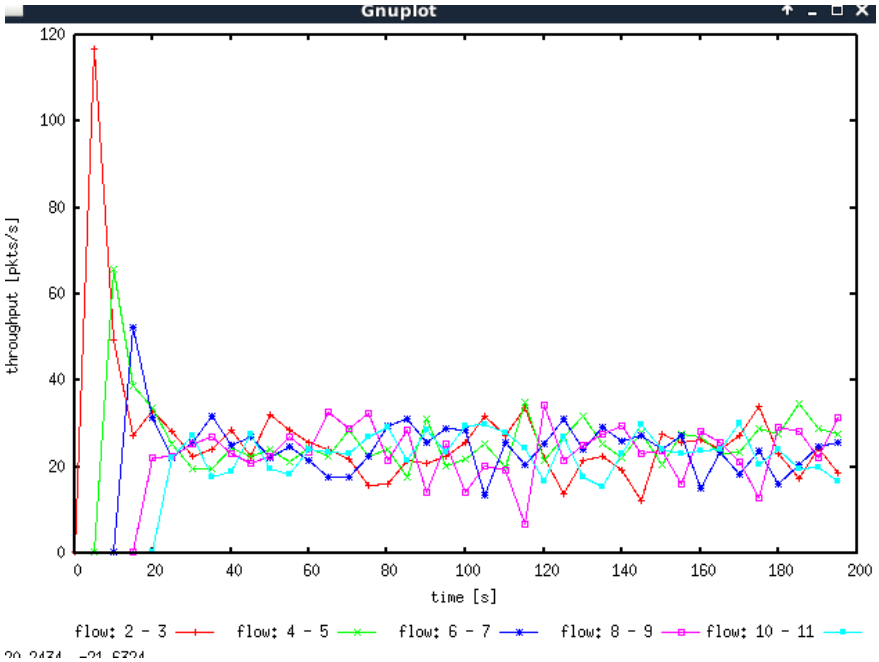
In the first implementation, the congestion window's size goes back to zero every time when lost event occurs, but in the second one, the congestion window just cut the window size to half of the current window size.

Average throughput in first implementation is 180 packets per second(777600 bps) when the window size set as 100, in second implementation it is roughly 190 packets per second(820800 bps) when the window size set as 100. The avg throughput is greater in TCP Reno than in the behaviour of TCP Tahoe.

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.

Yes, it does. We can tell the link capacity is roughly 120 packets per second from the performance of flow 2-3 between t = 0s to t = 10s. And there are 5 different flows in total, each flow has an average throughput of around 24-25 packets per second from the graph. And 24 packets per second * 5 = 120 packets per second which is equal to the link capacity. Thus, this observation tell us each flow get an equal share of the capacity of the common link.



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 behaviour. Argue about whether you consider this behaviour to be fair or unfair.

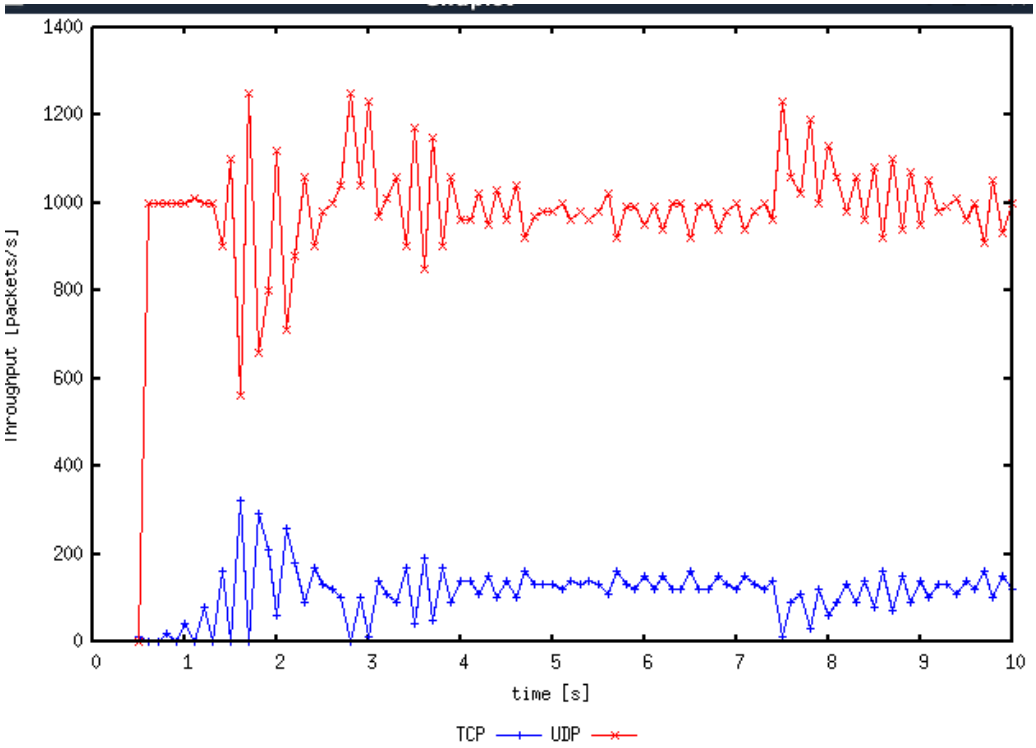
The throughput of the pre-existing TCP flows will all reduced when a new flow is created, and all the spare capacity will be given to the new created flow. This behaviour is contributed by the mechanisms of TCP AIMD algorithms which is a feedback control algorithm that combines linear growth of the congestion window when there is no congestion (new flow get the throughput capacity), and an exponential reduction when congestion is detected (new flow is created). I would consider this behaviour to be fair since all flows include the new created one would get an equal share of the throughputs in the end.

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 ?

I expect the UDP flow will send much more packets per second(higher throughput) than the TCP flow if the capacity of the link is 5 Mbps.

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.



TCP flow has a congestion control mechanism(slow-start, AIMD algorithm) and fair flow control which results in sending less packets per second, but UDP does not have any flow control and it does not care about if congestion occur in the link, thus UDP flow will send much more packets and has a higher throughput than TCP. The mechanisms force the two flows to stabilise to the observe throughput is the length of the packet and the capacity of the link (5 Mbps).

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?

Advantage:

Higher throughput than TCP by using UPD since UDP will send as many packets as it can every seconds. Higher transmission rate and take less time to send the data since there is no congestion control.

Disadvantage:

Packet get loss or transmission failed much easier than using TCP, need to retransmit packets for a file more often than TCP, so not as reliable as TCP.

If everybody started using UDP instead of TCP, all the users will start sending as many packets as possible and it will cause a huge congestion in the link, then many more packets will get dropped and data lost for each user. So it's won't be more efficient for everybody to use UDP though UDP might have higher throughput than TCP, on the contrary, each user will have less effective throughput capacity.