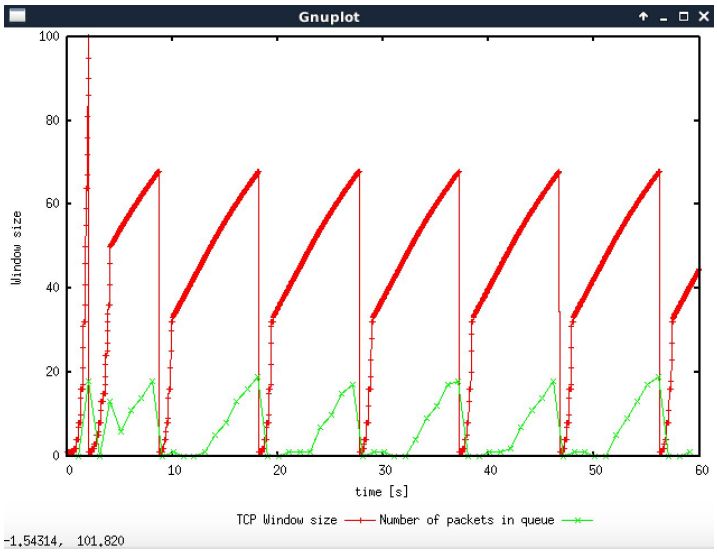
Exercise 1: Understanding TCP Congestion Control using ns-2

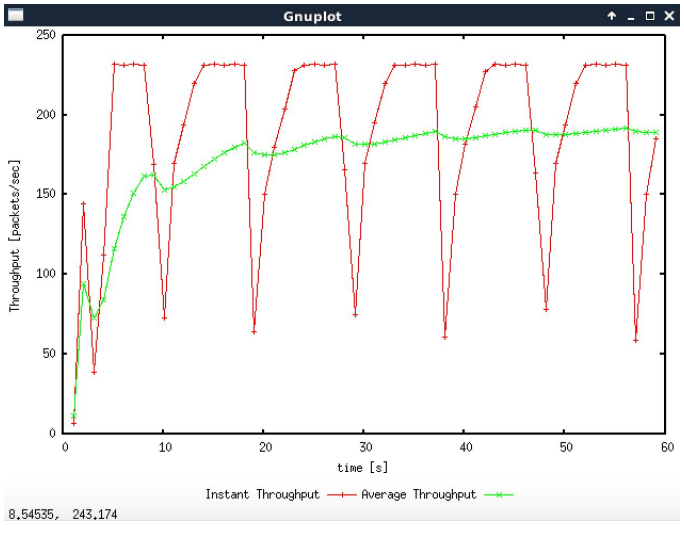
Question 1. What is the maximum size of the congestion window that the TCP flow reaches in this case? What does the TCP flow do when the congestion window reaches this value? Why? What happens next? Include the graph in your submission report.

The maximum size of the congestion window that the TCP flow reaches is 100 in this case. When it reaches the highest value, which is 100, and if it experiences a packet loss event in the case of TCP Tahoe, the window size becomes 1 again and the slow start phase occurs next.



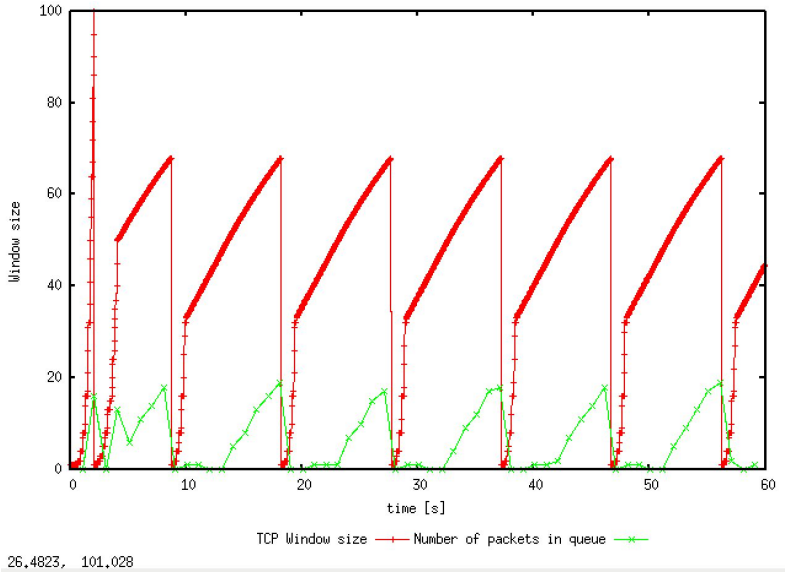
Question 2: From the simulation script we used, we know that the payload of the packet is 500 Bytes. Keep in mind that the size of the IP and TCP headers is 20 Bytes, each. Neglect any other headers. What is the average throughput of TCP in this case? (both in number of packets per second and bps)

The average throughput which can be examined by hovering over the green line, is approximately 190 packets/second. By adding the payload of the packet and the size of the IP and TCP headers, each packet is 540 bytes (500 bytes + 20 bytes + 20 bytes). To calculate the average throughput in bps, 540 * 8 (8 bits in a byte) * 190 = 820,800 bps. Both answers are close approximations.

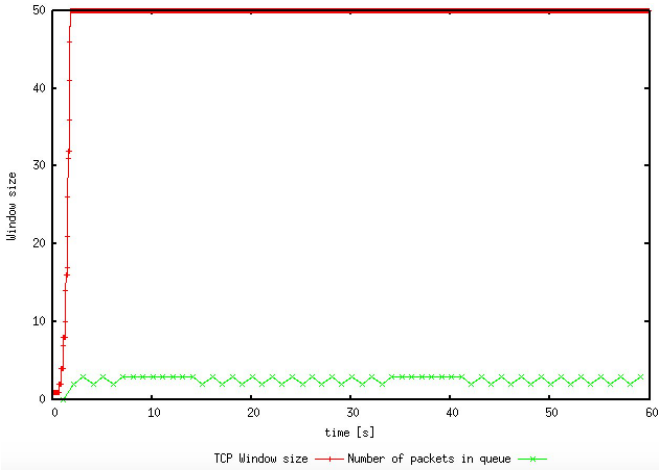


Question 3: Rerun the above script, each time with different values for the max congestion window size but the same RTT (i.e. 100ms). How does TCP respond to the variation of this parameter? Find the value of the maximum congestion window at which TCP stops oscillating (i.e., does not move up and down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

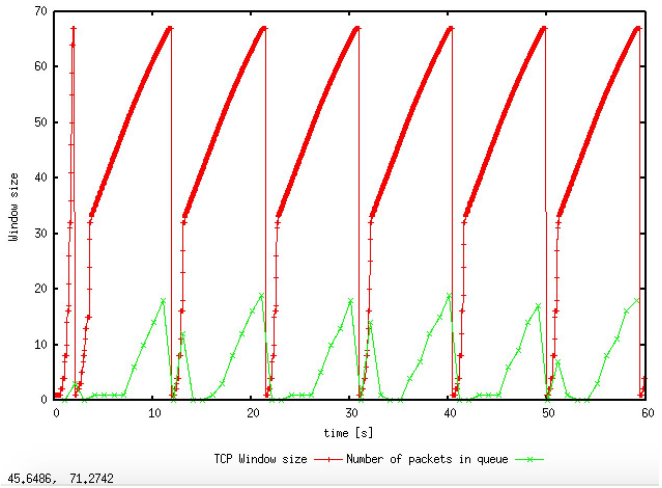
TCP responds to the variation of this parameter by adjusting the max window size to the parameter that we give it. The smaller the window size, the less oscillations that are present.



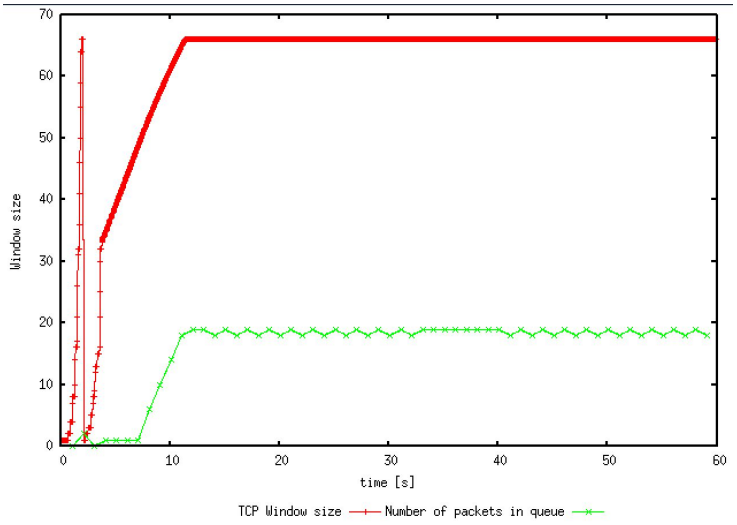
This image above is window size of 100, of which it is still oscillating.



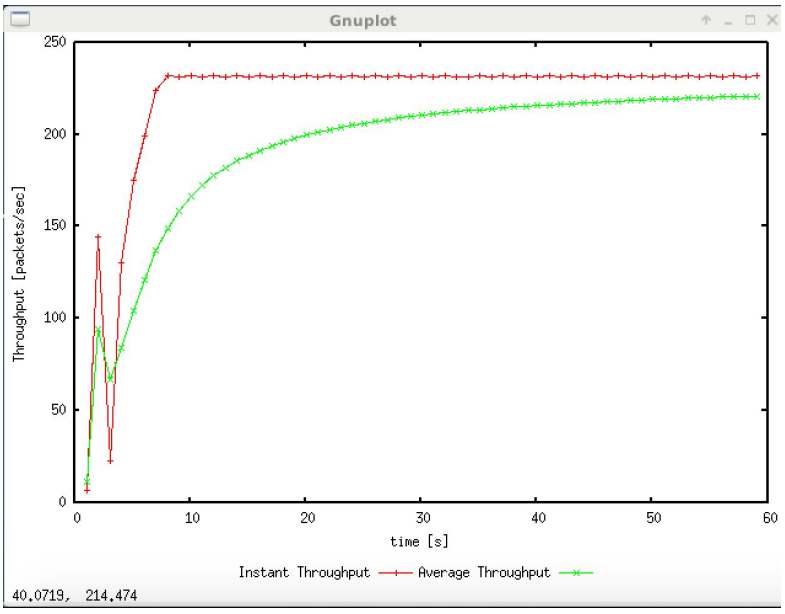
This one is a window size of 50, of which can be observed that the oscillations have stopped occurring.



This image above is 67, and oscillations can still be observed.



The image above shows that the maximum congestion window at which TCP stops oscillating is 66, of which stable behaviour can be observed.



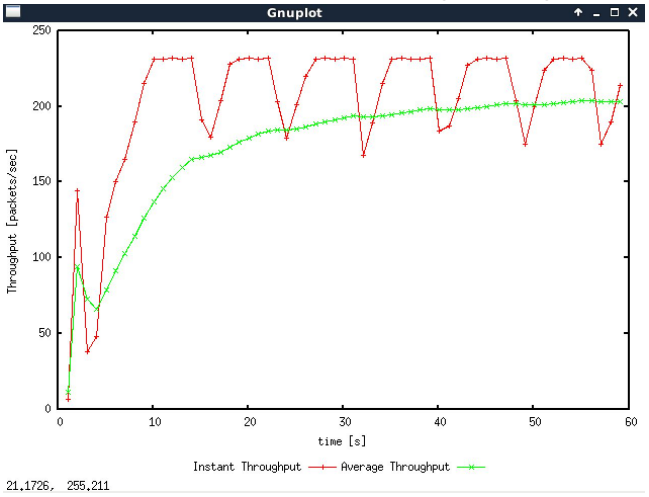
The average throughput in packets is 214 packets/sec.
The average throughput in bps is 214 *8 * 540 = 924,480 bps.

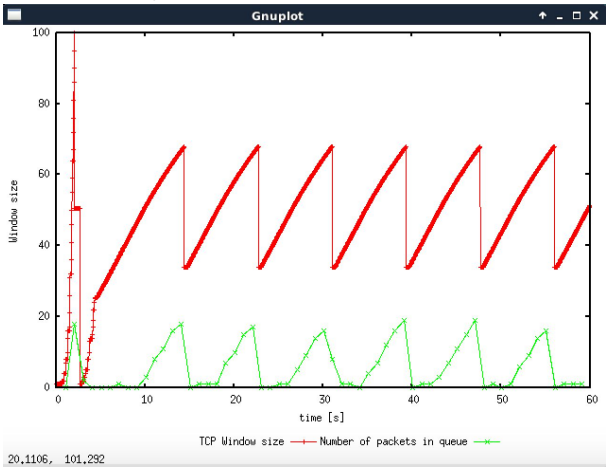
The actual average throughput takes up 1Mbps is equivalent to 1,000,000 bps.
Link capacity:
(924,480 bps/ 1,000,000 bps) * 100 = 92.448%.

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?

In contrast to the previous implementation, the number of times the congestion window goes back to zero in the Reno implementation is only once, whereas in the Tahoe implementation, it occurred multiple times (7 is shown). Thus, the Reno average throughput is a lot more stable in contrast with the Tahoe implementation. This difference between the two comes from the fact that they deal with packet loss differently. Reno resets it to half the previous window size whilst Tahoe will reset the window size to 1.

The average throughput in the Reno implementation is approximately 200 packets/sec whereas the Tahoe implementation average throughput is approximately 190 packets/sec.

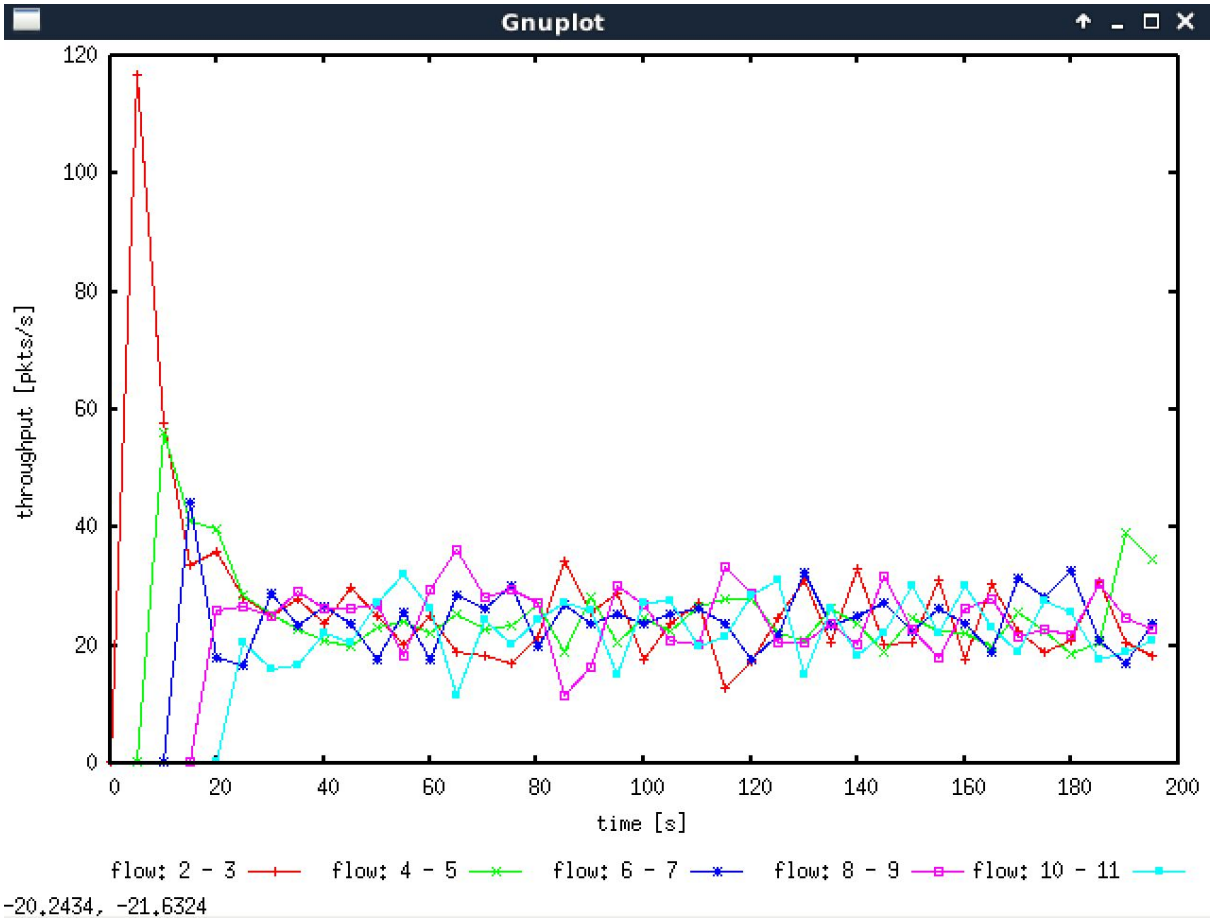




Exercise 2: Flow Fairness with TCP

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.

As observed from the graph, each flow does get an equal share of the capacity of the common link. Although each flow is fluctuating on its own, they average out to share an equal 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 flow will decrease significantly everytime a new flow is created in order to achieve sharing bandwidth equally.

The mechanisms of TCP that contribute to this behaviour is congestion control. Congestion control moderates capacity taken up by each flow to avoid congestion collapse.

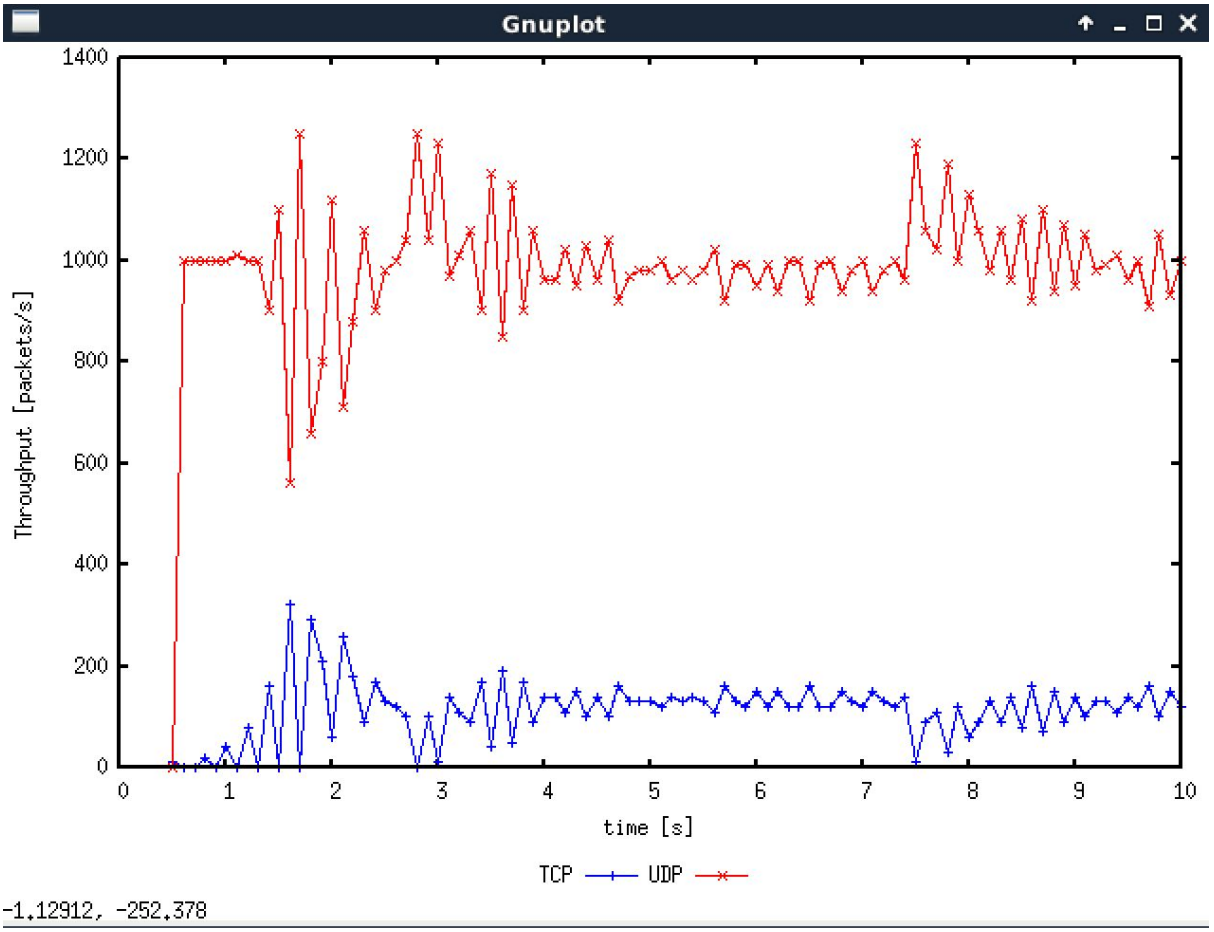
I think that this behaviour is fair because each connection will automatically be adjusted to share equal amounts of bandwidth.

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 UDP flow to take up a large amount of the capacity because UDP has no regulations thus it can send packets continuously whereas TCP has multiple regulations to prevent congestion.

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.

UDP achieves a higher throughput than TCP because it lacks regulations, thus allowing it to continuously send packets. TCP has regulations such as acknowledgements which results in the lower number throughput of packets. TCP's throughput remains lower in comparison to UDP because of its inability to send continuous packets without acknowledgements and congestion control. UDP remains stabilised high due to its ability to send continuous packets with no regulations.



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?

Advantages of UDP over TCP	**Disadvantages of UDP over TCP**
High throughput rate, fast transmission	Unreliable, risk of losing packets
Faster because it is a “best effort” protocol	No regulations and lack of congestion control can lead to congestion collapse, no error checking, no flow control

If everyone started using UDP instead of TCP, various issues would arise due to the lack of regulations and protocols that TCP has. Issues would include loss of packets, congestion collapse, corrupted packets etc.