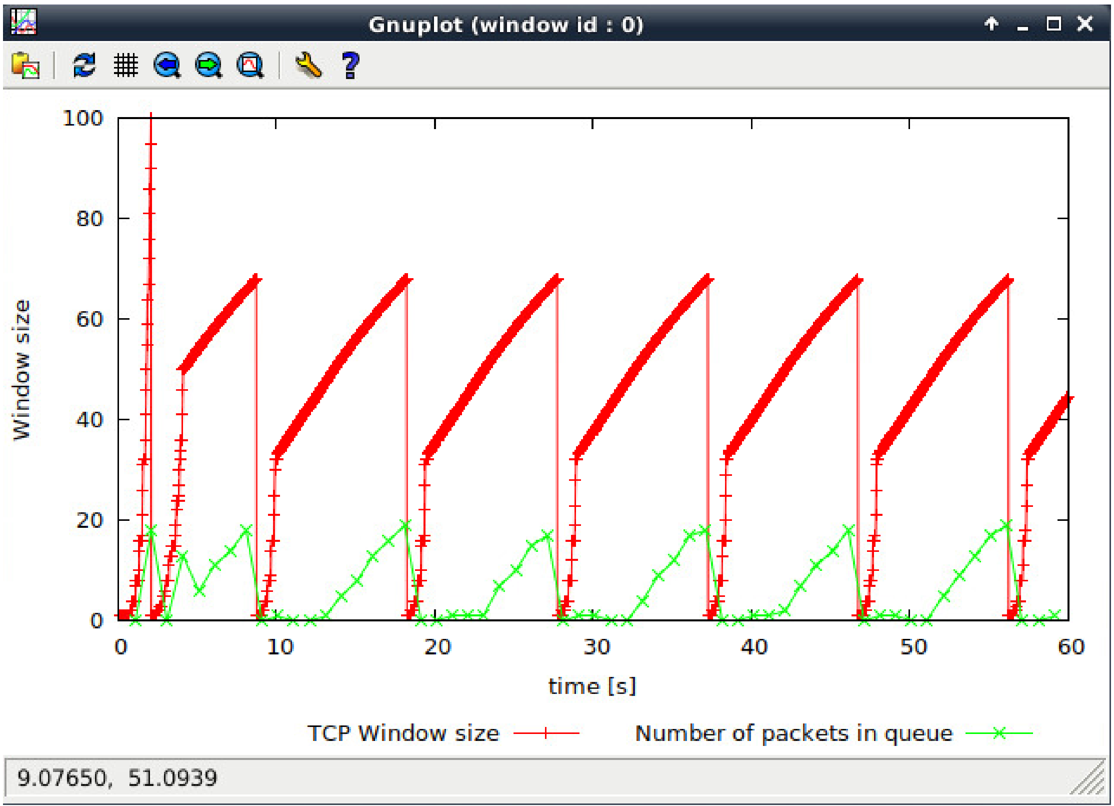
Written by Maowen Zhou for COMP9331 lab5.

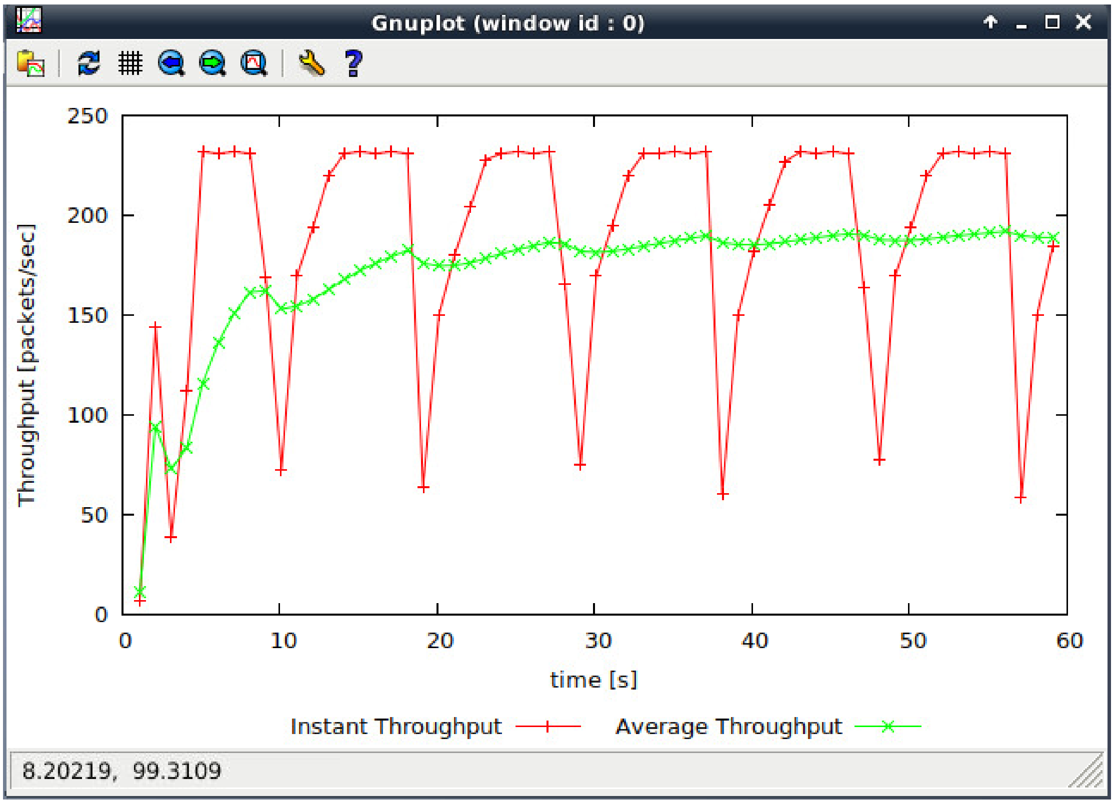
Exercise 1:

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? Include the graph in your submission report.



The maximum size of the congestion window that TCP flow reaches in this case is 100 MSS, although we have set the maximm congestion window 150 MSS. Because the maximum size of queue is only 20, any additional packets are dropped. When it reaches 100 MSS it means loss occurs, so the sender stops increasing the congestion window size, changes the value of ssthresh to the half of congestion window and at the same time sets congestion window equal to 1 MSS , because TCP-Tahoe will set cwnd equal to 1 on triple dup ACK and timeout, then begin slow start phase, when congestion window reaches ssthresh, additive increase phase(congestion avoidance) begins until loss occurs again.

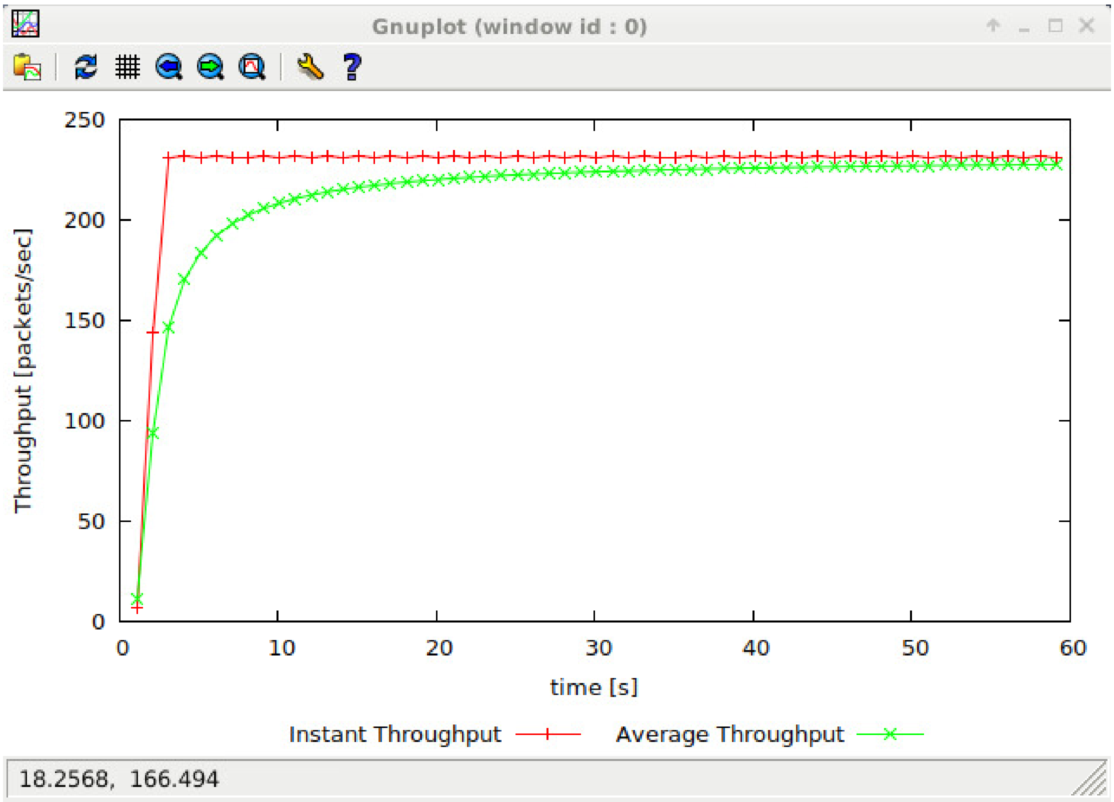
Q2: From the simulation script we used, we know that the payload of the packet is 500 Bytes. Keep in mind that the size of the IP and TCP headers is 20 Bytes, each. Neglect any other headers. What is the average throughput of TCP in this case? (both in number of packets per second and bps)



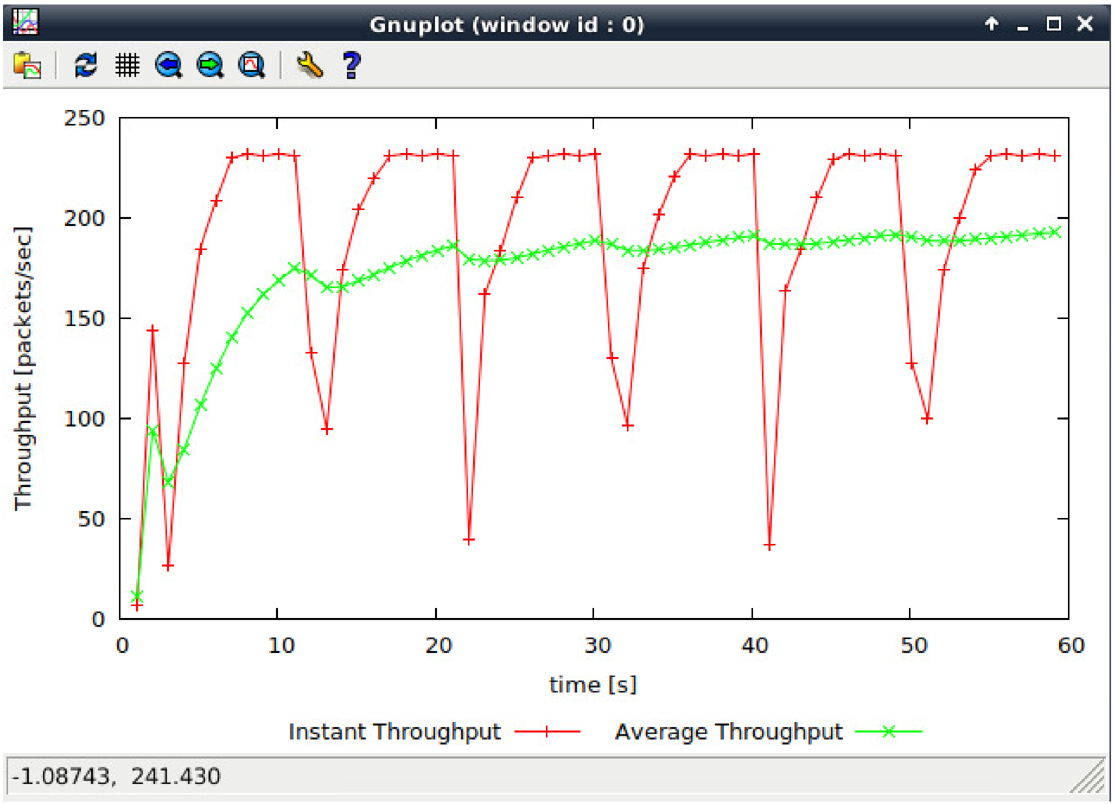
As is shown in the picture above, the average throughput of TCP is around 190 packets per second, because the average throughput is relatively stable after 20 second.

The size of packets is (500 + 20 + 20) bytes, so the throughput in bps is 540 * 190 * 8 = 820.8kbps.

Q3: 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)?



The max congestion window size is 50



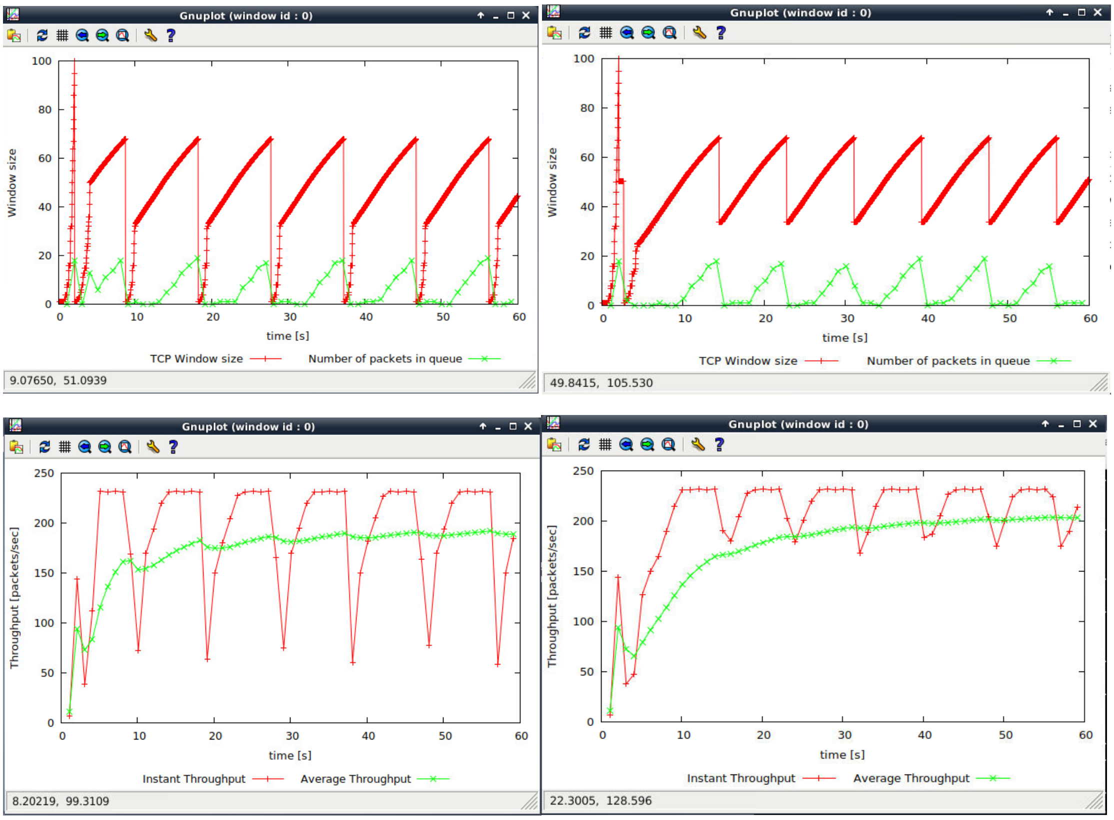
The max congestion window size is 60

So according to the two pictures above, there will be less congestion and the instant throughput and average throughput tend to increase and become stable. When the maximum congestion window size is set to 50, TCP stops oscillating.

The average packet throughput is around 220 packets per second, the average throughput in bps is 220 * 540 * 8 = 950.4 kbps. I would say it is almost close to the link capacity.

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?

TCP Tahoe TCP Reno



There are some differences, one is that TCP Tahoe will set cwnd to 1 for both triple duplicate ACK and timeout, but TCP Reno will set cwnd to 1 on timeout and cwnd = cwnd / 2 on triple duplicate ACK. And TCP Reno has Fast Recovery.

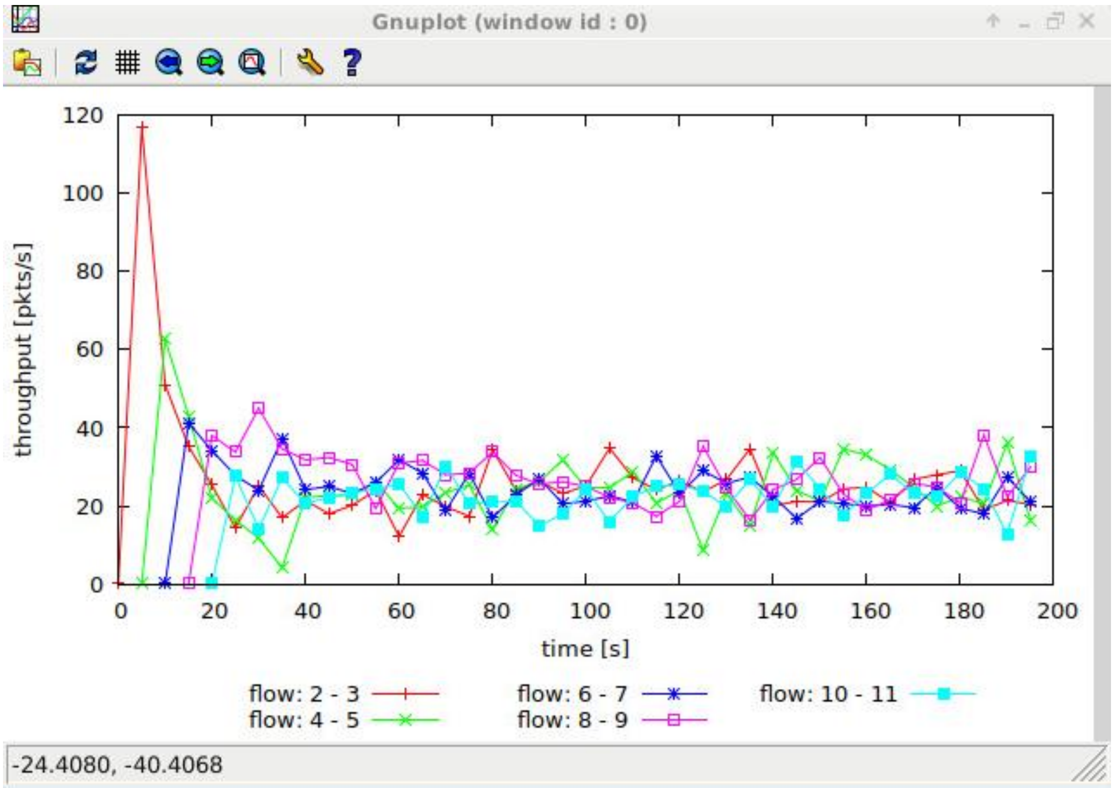
So in TCP Reno case, most of the losses are detected due to triple duplicate ACKs rather timeouts.

TCP Tahoe throughput: around 190 packets per second, 190 * 540 * 8 = 820.8 kbps.

TCP Reno throughput: around 200 packets per second, 200 * 540 * 8 = 864 kbps.

Exercise 2:

Q1: 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.



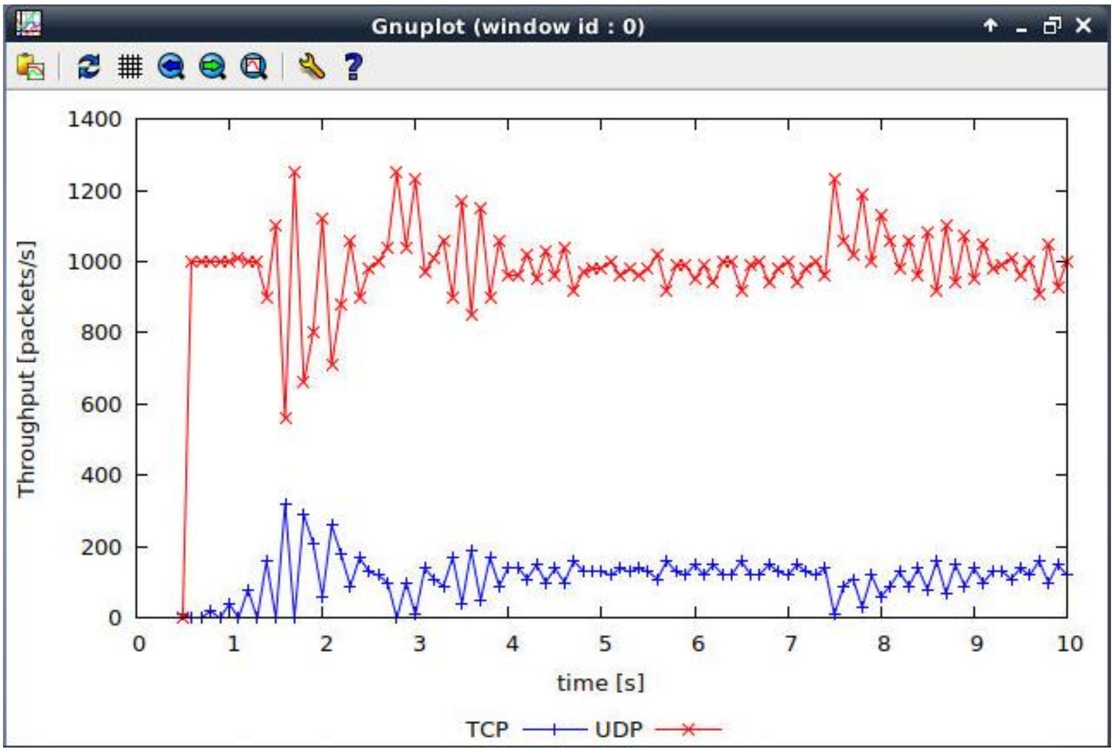
Every time a new flow connects to the common link, the throughput of each flow will drastically decrease. This indicates that each flow get an equal share of the capacity of the shared link. This approximate fair behavior is a direct result of the AIMD congestion algorithm used by TCP.

Q2: 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.

When a new flow joined the link, the throughput of the pre-existing TCP flows will decrease drastically and share the link with the new flow. TCP is trying to make sure each flow take the same percentage of the whole throughput. Thus, this is a fair behavior, all existing TCP connections detect losses through duplicate ACKs and timeout and adapt the size of their congestion window in order to avoid overwhelming the network, every connection needs to reduce accordingly.

Exercise 3:

Q1: How do you expect the TCP flow and the UDP flow to behave if the capacity of the link is 5 Mbps ? Now, you can use the simulation to test your hypothesis. Run the above script as follows.



I expect that UDP flow has higher throughput than TCP, because UDP does not implement any congestion control, which means that UDP flow will not reduce its transmission rate if there is congestion.

As it is shown above, my expectation is right. UDP achieves higher throughput than TCP, because TCP has congestion control but UDP does not.

Q2: 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.

Because UDP does not have congestion control, so it will transmit packets at a constant rate regardless of the drop of packets. But this will affect TCP flows, it will decrease transmission rate when it detects congestion in the network. So TCP flows will be forced to have a lower throughput due to more aggressive UDP flows.

Q3: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: UDP is not affected by the congestion in the network, it will keep transmitting data at an unrestrained speed, this may reduce the delay for transferring files, but this is not always the case.

Disadvantages: UDP will lay much burden on the network. And reliable data transfer need to be applied if UDP is uses as the file transfer protocol.

If everybody starts using UDP instead of TCP, the network may not support such high transmission rate and eventually give up working.