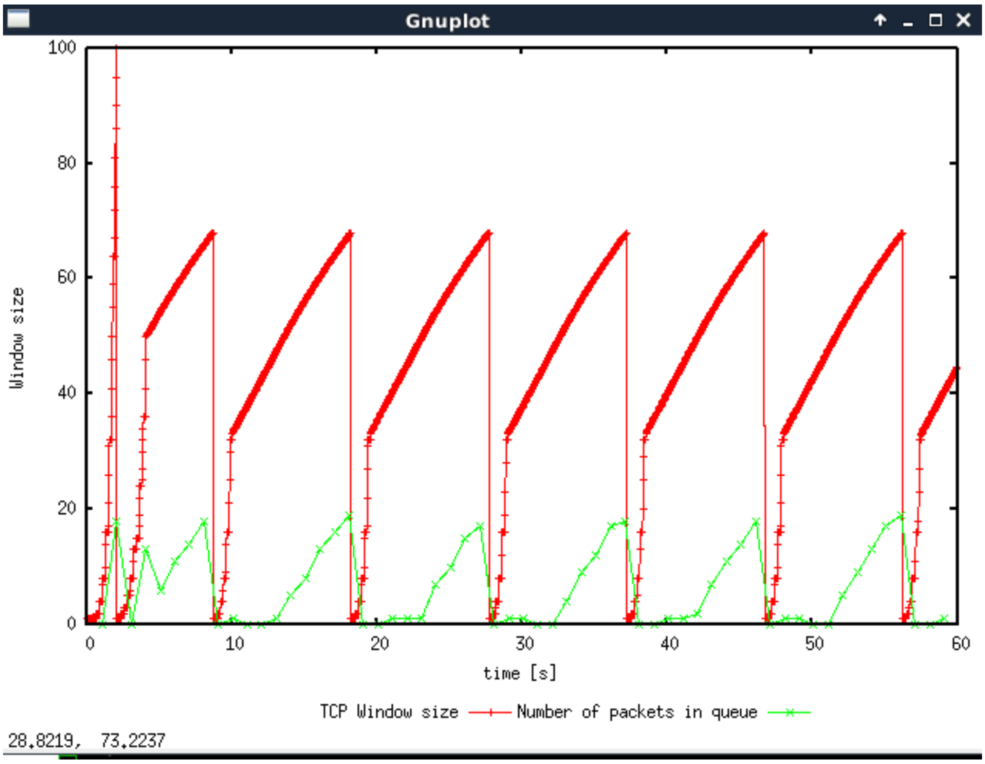
Lab5 COMP9331-21-T2
Z5216214
Haojin Guo

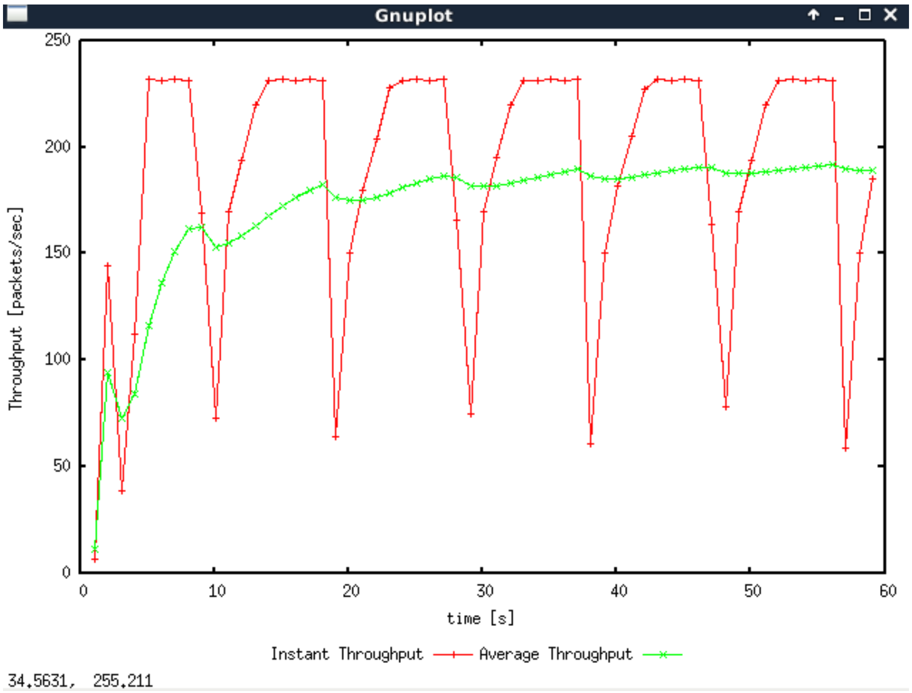
Exercise 1: Understanding TCP Congestion Control using ns-2

Question 1:



1) The maximum size of the congestion window that the TCP flow reaches in this case is 100 packets during the slow start phase. But the maximum congestion window we have set is 150, this is because the maximum queue is only 20, any additional packets will be dropped.
2) From the graph above, as the window size is reached to 100, the queue will become full which leads to a congestion event. Then, the sender stops increasing and reduces the congestion window size to 1 and the threshold to 1/2 the size of window —-50 packets.
3) Next, the congestion enters in the phase of slow start. When the window size reaches the threshold, the congestion avoidance (AIMD) phase will be activated. Eventually, the queue will become full again (which may result in packet loss). Afterwards, repeat the process.

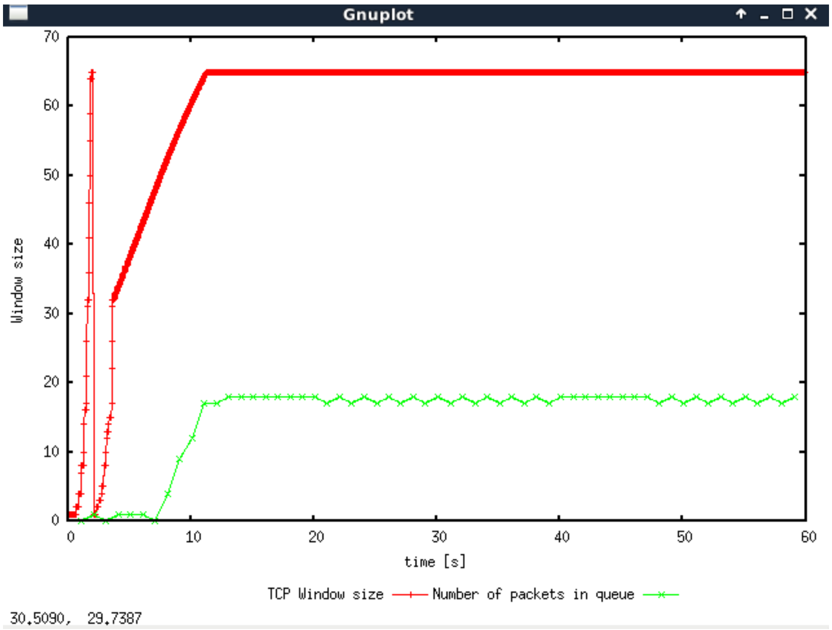
Question 2:



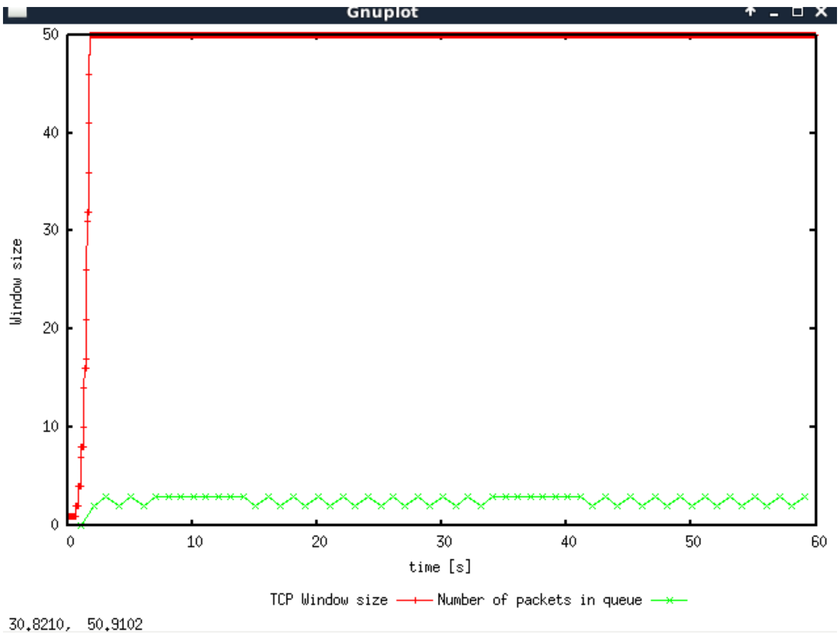
1) The graph above shows the average and instantaneous throughput (in packets/sec) for the TCP connection.
   After 20s, the average throughout is becoming relatively stable. And hence the average throughput of TCP is around $190\ pps$.
2)
   a) If the throughout computation includes header and payload data,
   The throughout = (500 TCP payload + 20 TCP header + 20 IP header) *8 *190 packet/sec = 820.8bps
   b) If the throughout computation only includes the payload data.
      The throughout = 500*8*190 packet/sec = 760bps

Question 3:

1) The graph below shows the window size when the initial max congestion window is set to 66. When the congestion window size <= 66, after the window size returns to 1 at about 2 seconds, the threshold will become 1/2 size (33 packets) of the maximum window size we set. And then the slow start phase will begin, then after the second phase of congestion avoidance (AIMD), the final congestion window size will reach the maximum window size we set (66 packets) and remain stable state.

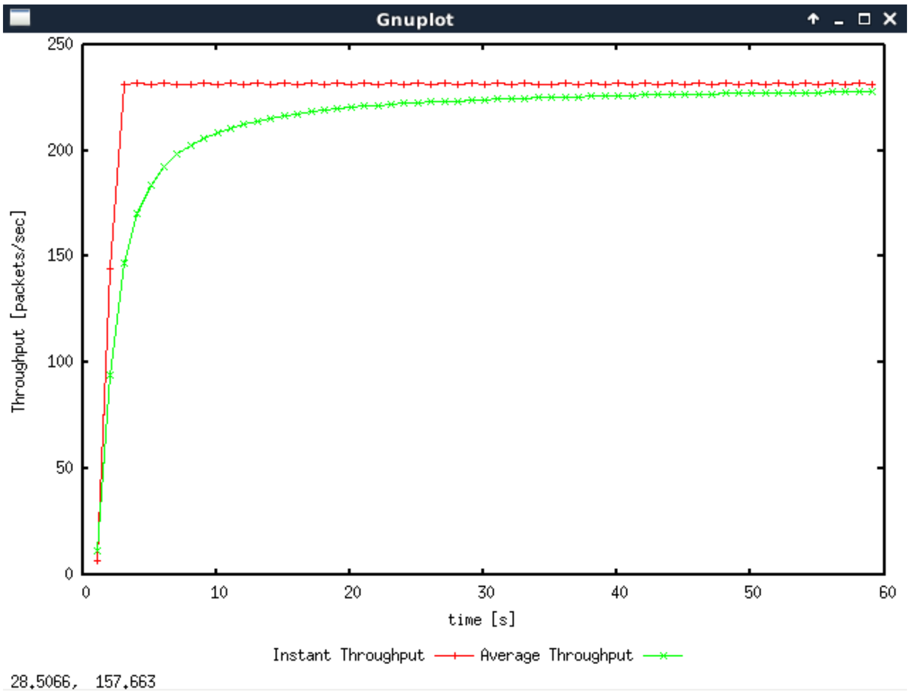


2) The graph below shows the window size when the initial max congestion window is set to 50. If we set the maximum congestion window size to 50 packets, TCP starts fast with slow start, and then the window size increases to the maximum congestion window size and then keeps stable.

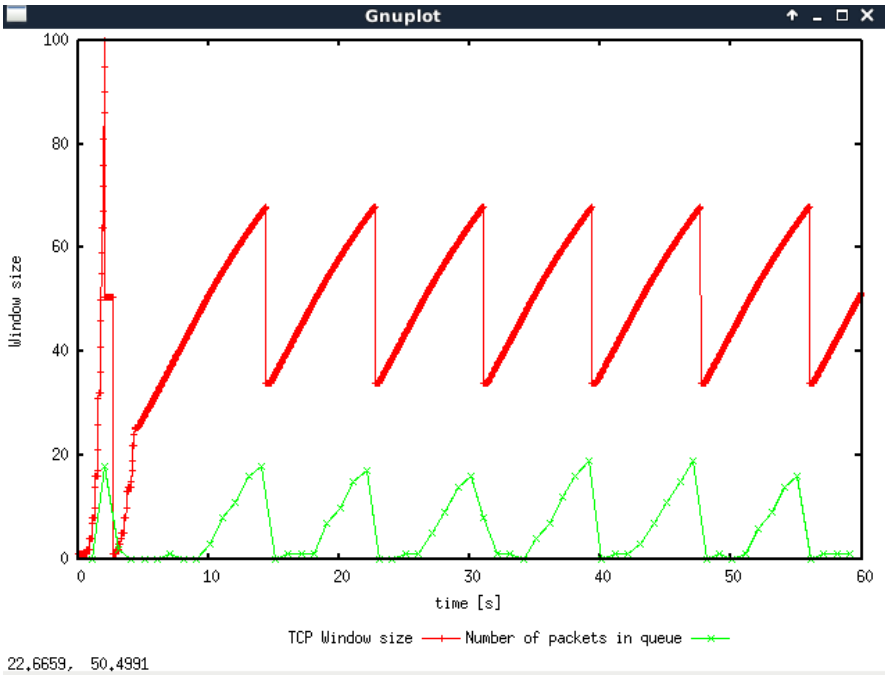


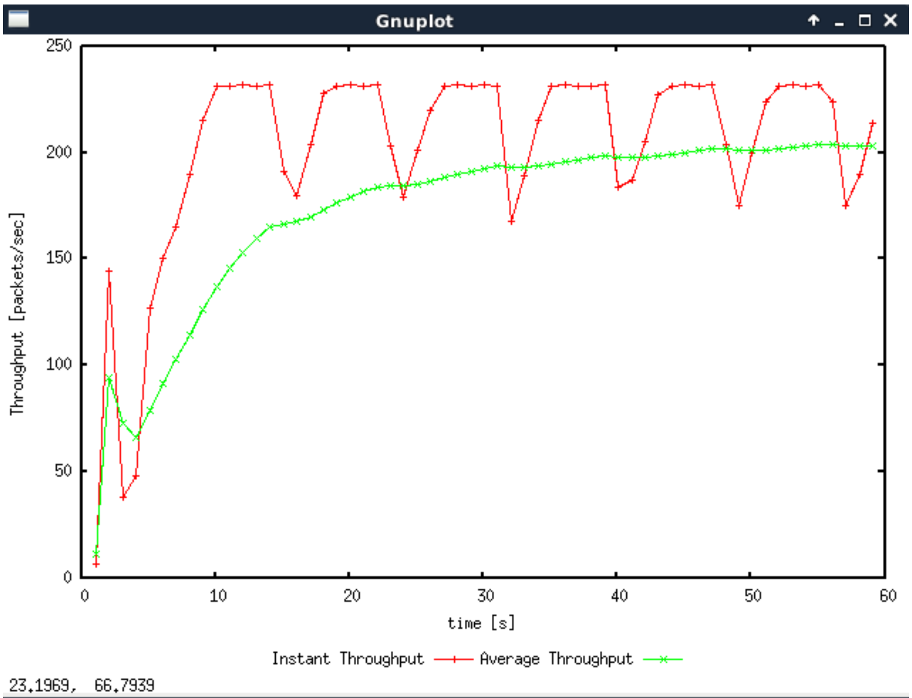
3) The graph below shows the throughput when the initial max congestion window is set to 50.

The average packets throughput of TCP is around $220\ pps$. And The average throughput is $220x500x8 = 880\ Kbps$, which is almost equal to the link capacity (1 Mbps)



Question 4: TCP Tahoe vs TCP Reno





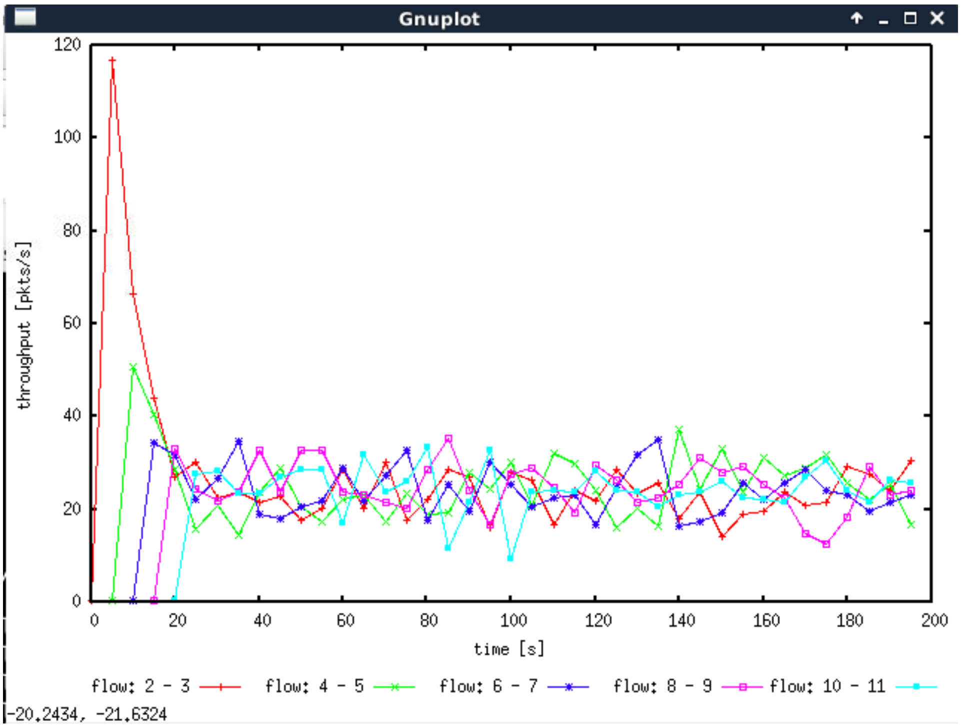
1) As for TCP Reno, the TCP connection does not enter slow start in the most part (except one timeout event at the start). When there are 3 duplicate ACKs (Fast retransmission mechanism), the congestion window size of render will be reduced to 1/2 the size of the current congestion window and increases it linearly by AIMD util losses occurs taking place again. This is different to TCP Tahoe where the congestion window is reduced to 1 after each congestion event.

2) The throughout is round 200 packet/sec for TCP Reno, which is greater than TCP Tahoe 190 packet/sec slightly. This is may because that the TCP Reno does not have the repeat slow start when congestion event occurs.

Exercise 2: Flow Fairness with TCP

Question 1:

Yes, TCP is fair, the throughout for all 5 connections can get a roughly equal share of the capacity of the common link from the graph below. After the initial slow start phase, 5 connections achieve a long-term fairness based on the AIMD strategy of adapting the congestion window size. Since all flows experience the same network conditions, they will react in the same manner.

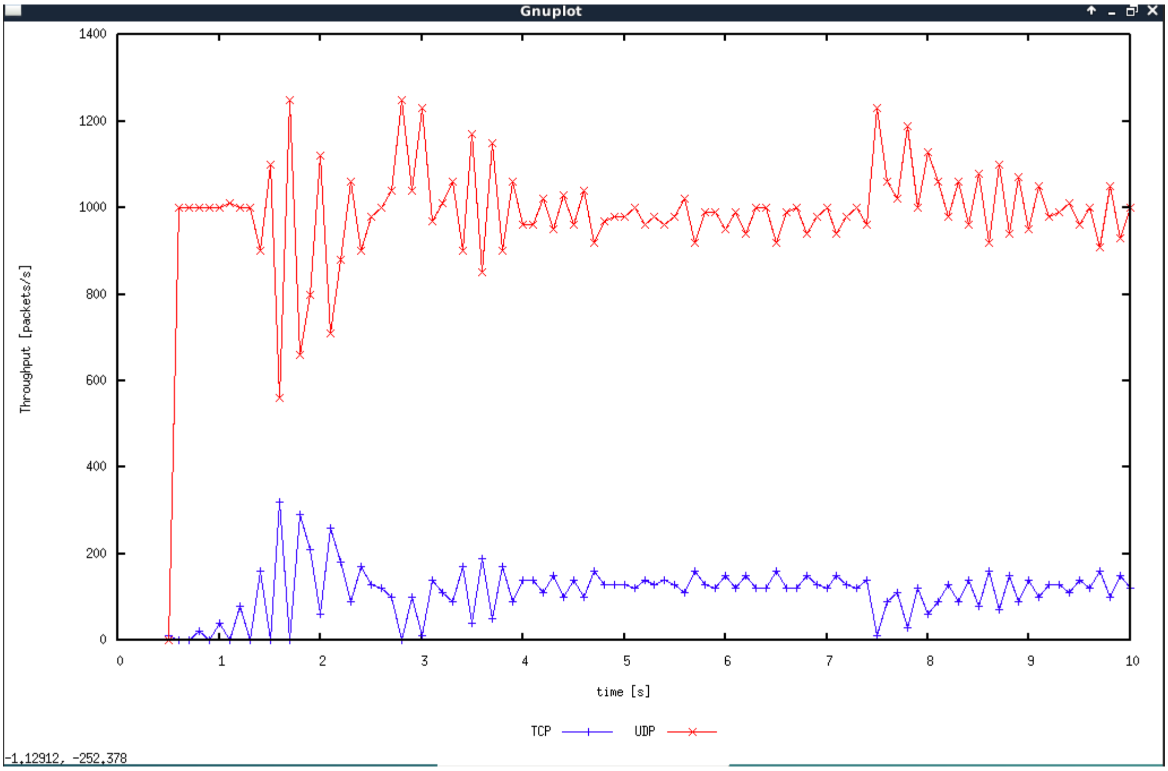


Question 2.

According to the graph above, when a new flow is created, the throughput of all pre-existing flows is reduced. This is because the new flow added will increase quickly result in congestion during the slow start phase. And then every TCP connection will adapt the size of their congestion window size by detect losses through ACKs and timeouts. This is a fairness process, since once new traffic is joined in, all existing traffic is reduced accordingly.

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?



UDP connection cannot reduce its transmission rate when congestion happens. In contrast, TCP provide congestion control, leading to TCP connections will experience the process of transmission rate adaption. Also, according to the graph above, when the capacity of the link is 5 Mbps, the throughput of UDP is larger than TCP.

Question 2:

The throughput of UDP is higher than TCP. Because UDP does not restrained by congestion control and it transmit will a constant rate, regardless of if any of them get dropped. However, as for TCP, its congestion control will reduce its transmission rate when congestion is detected. This is unfair to TCP. Therefore, UDP could occupy more network resources and TCP shares little.

Question 3:

1)

	Advantage	Disadvantage

UDP	1. Unrestricted transmission regardless of congestion 2. Keeping transmitting with a higher speed. 3. The delay would be low	1. Unreliable transmission, because the file transfer protocol running on top of UDP would need to implement reliable data transfer (application layer). 2. UPD will increase the burden of the network.

2) If everybody started using UDP instead of TCP, there is a possibility that the network would encounter congestion collapse to a large extent, especially when it comes to the condition of flow congestion.