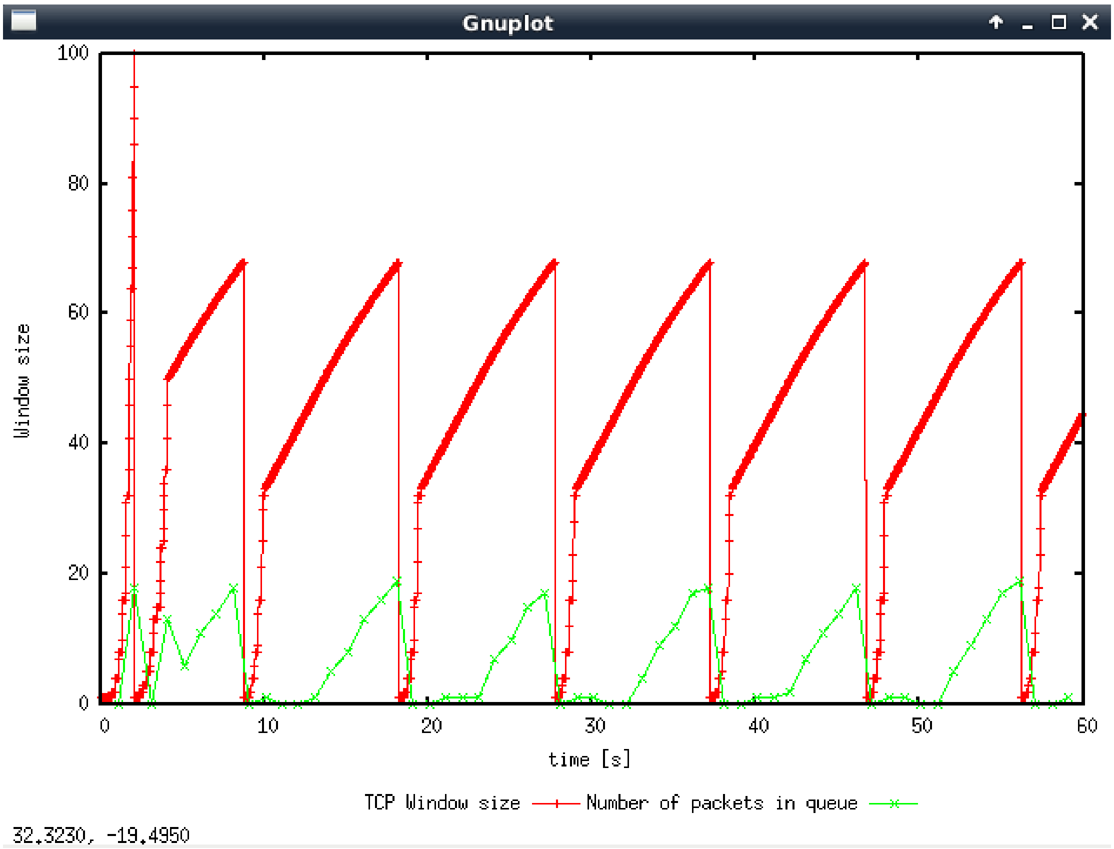
Lab 5

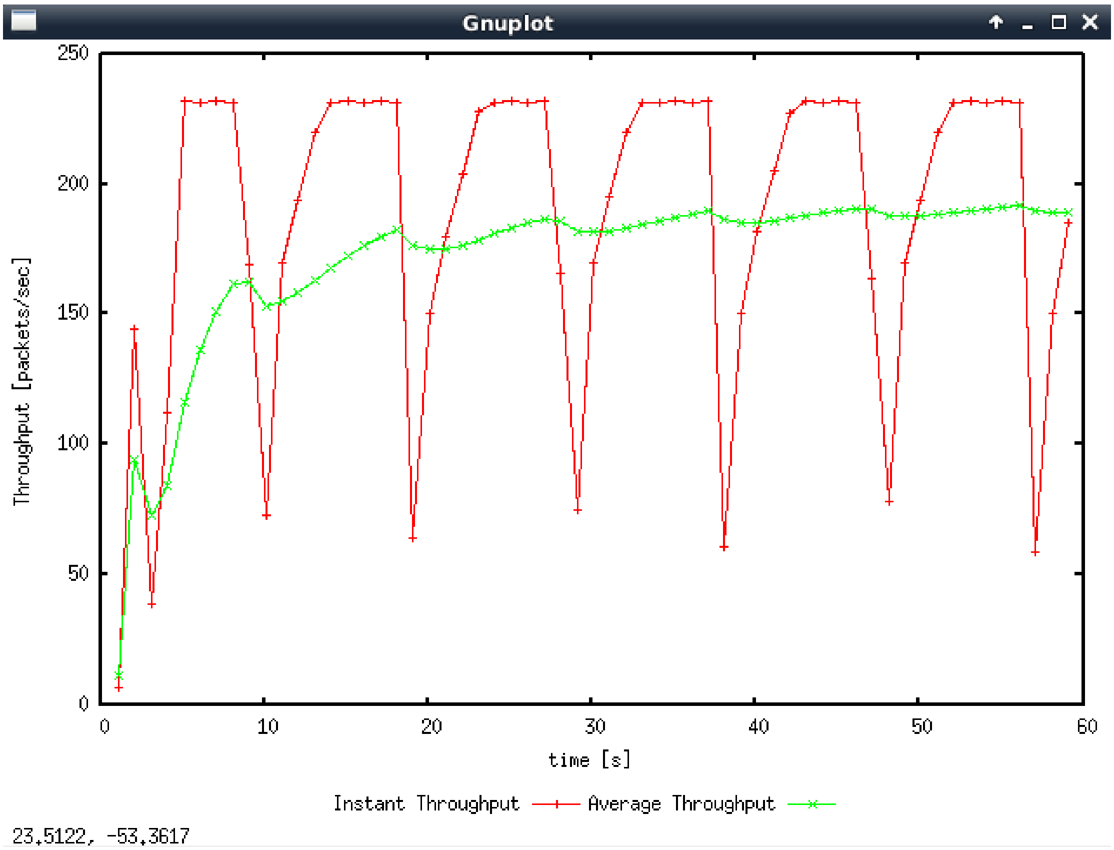
Exercise 1: Understanding TCP Congestion Control using ns-2

Question 1:



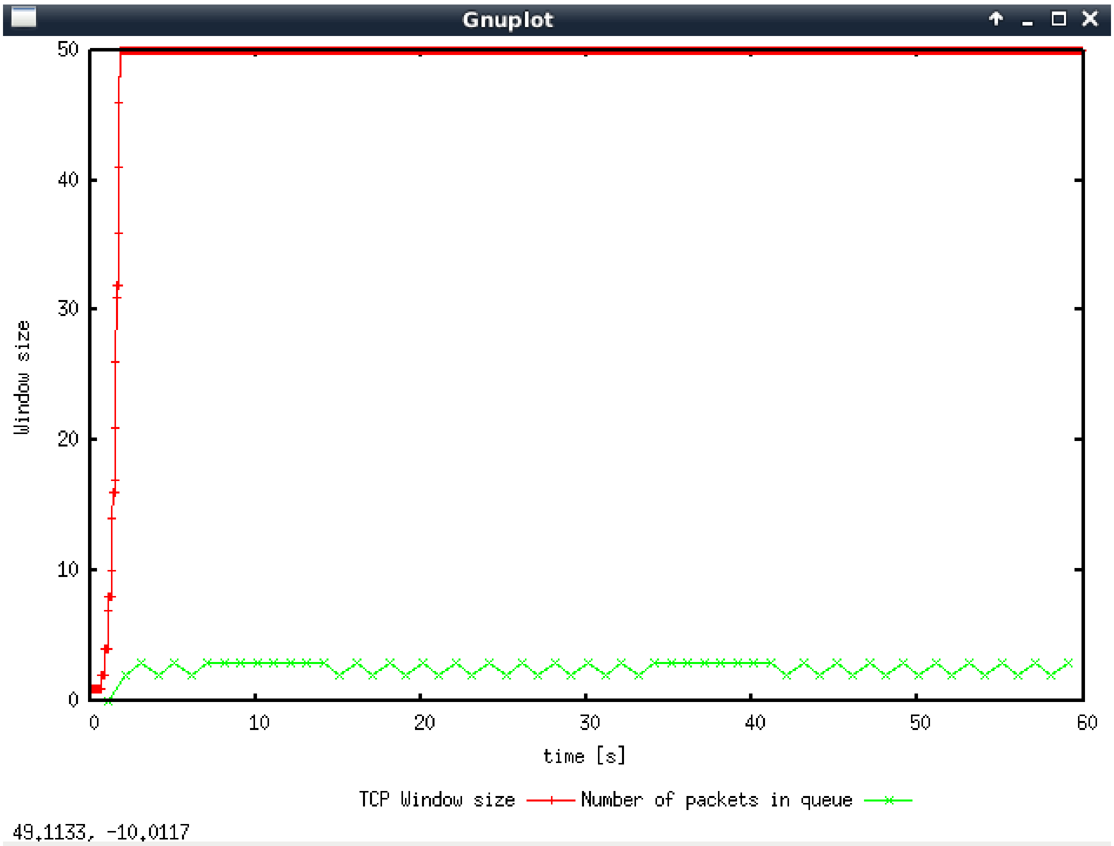
Although we have set the maximum congestion window 150 MSS, the maximum size of the congestion window that TCP flow reaches in this case is 100 MSS. Because the maximum size of queue is 20, any additional packets are dropped. When it reaches 100 MSS loss occurs, so the sender stops increasing the congestion 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 , TCP Tahoe will set cwnd equal to 1 on triple duplicate ACKs and timeout, then it begins the slow start phase, when congestion window reaches ssthresh, additive increase phase(congestion avoidance) begins until loss occurs again.

Question 2:

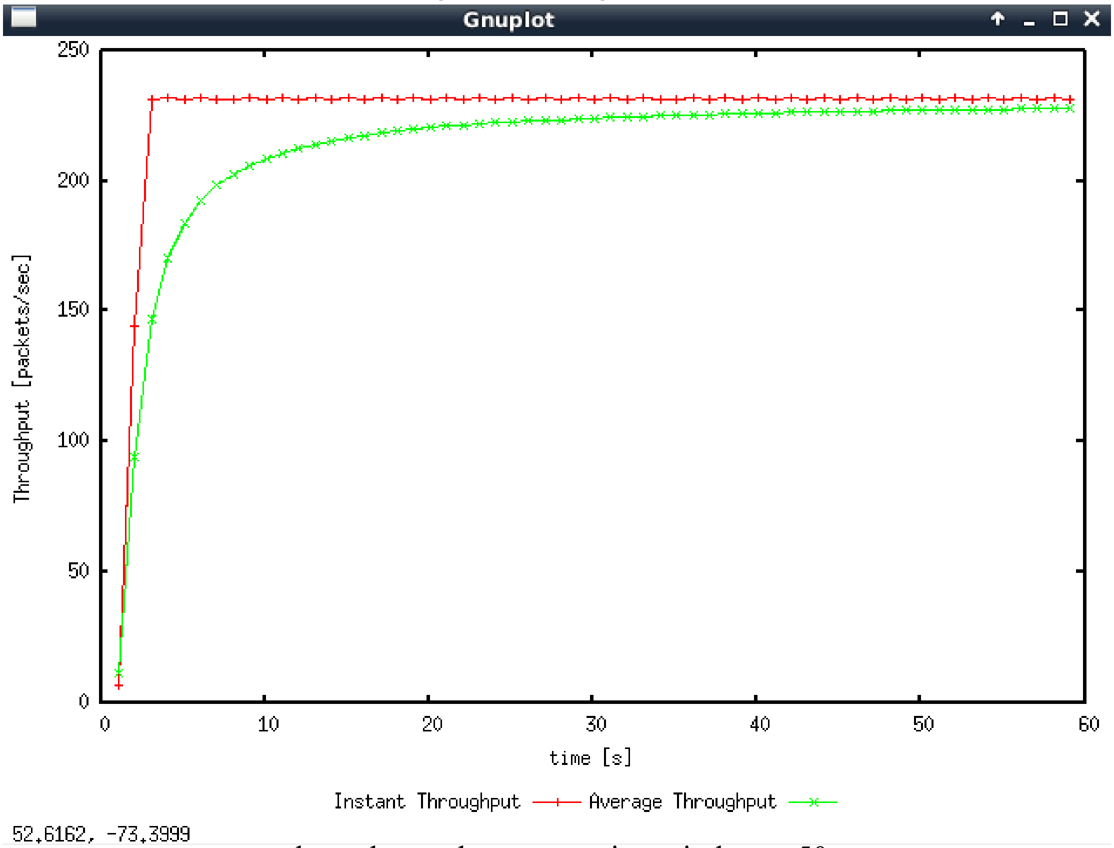


As is shown in the graph above, the average throughput of TCP is around 190 packets per second and the average throughput is relatively stable after 20 second. The size of packets is (500 + 20 + 20) bytes, so the throughput in bps is 540 * 190 * 8 = 820.8kbps.

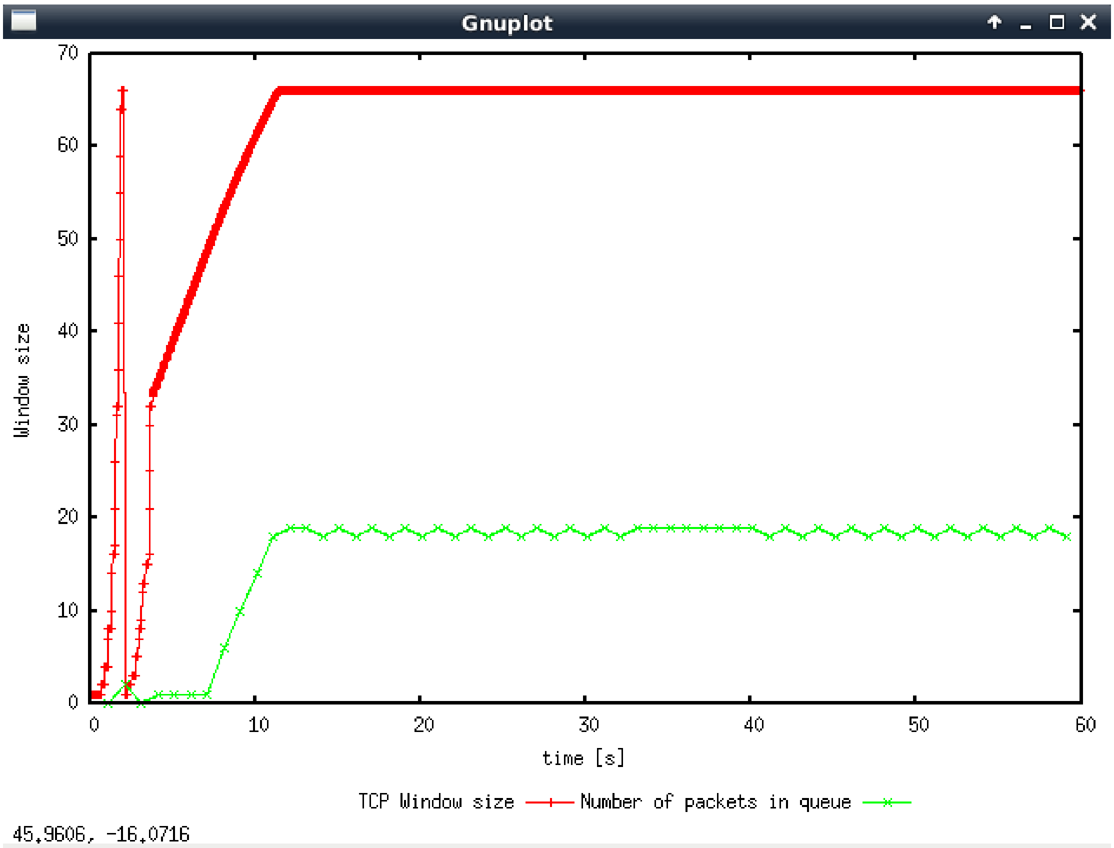
Question 3:



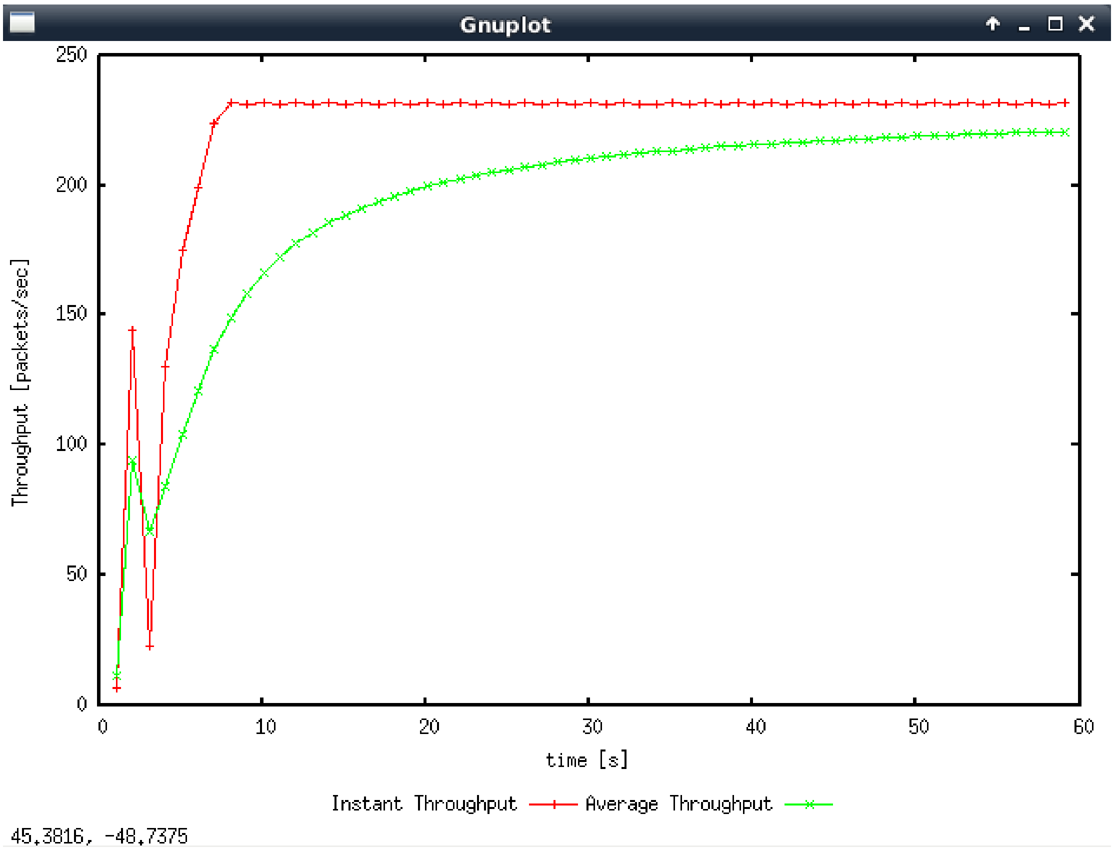
windows change when congestion window = 50



throughput when congestion window = 50



windows change when congestion window = 66



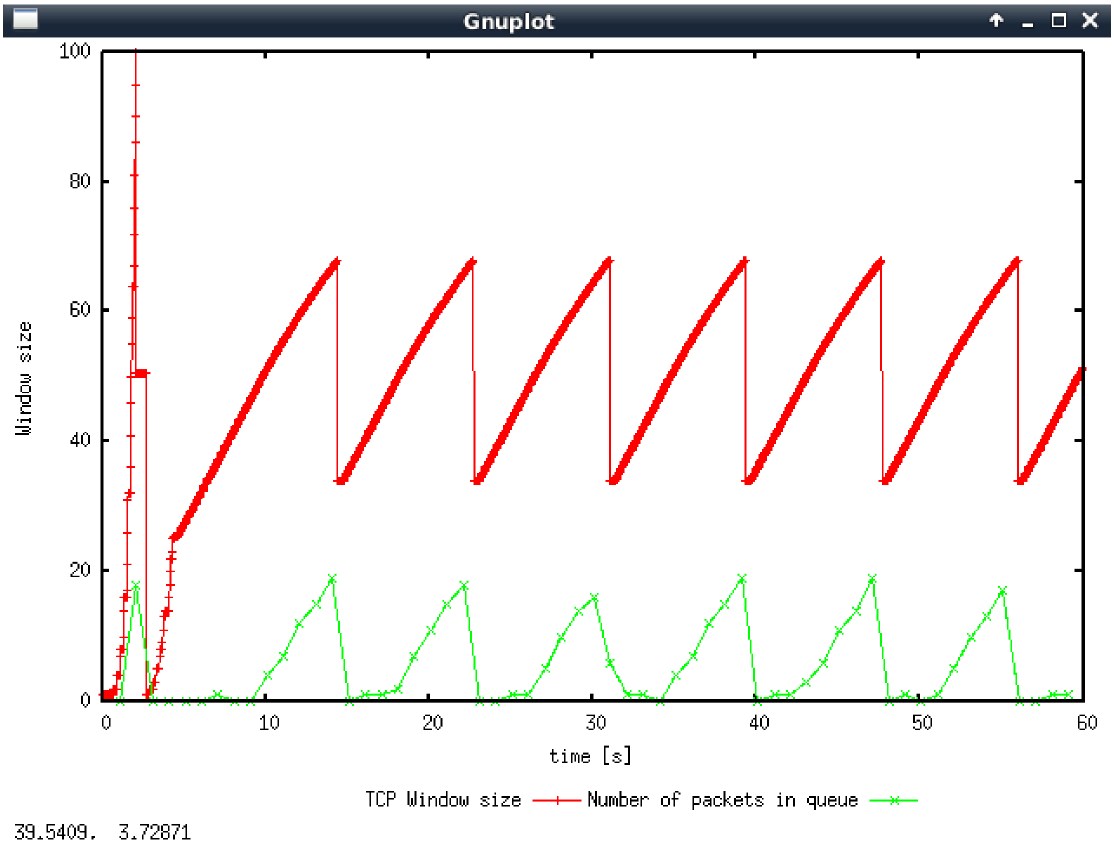
throughput when congestion window = 66

From graphs above, we can find that:

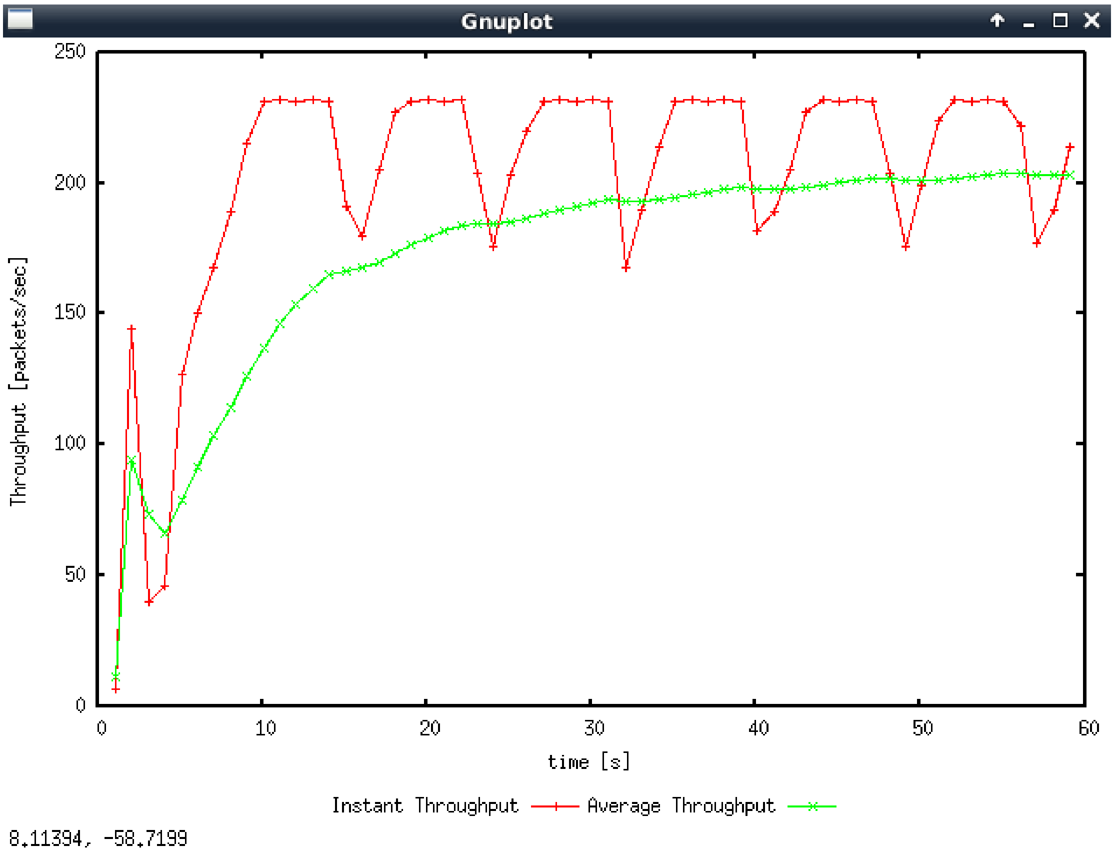
When 50 < The initial maximum congestion window size <=66, the oscillating situation stop after the first slow phase. And it get a balance situation when the congestion window size reduces to half of its size. This means that the queue gets full and packets are not dropped anymore.

When the initial maximum congestion window size <=50, TCP gets balance situation after slow start phase. The average packet throughput is around 225 packets per second. If we neglect the TCP and IP header, the average throughput is 225*500*8 = 900 kbps. If we take TCP and IP header into consideration, the average throughput is 225*540*8 = 972 kbps.

Question 4:



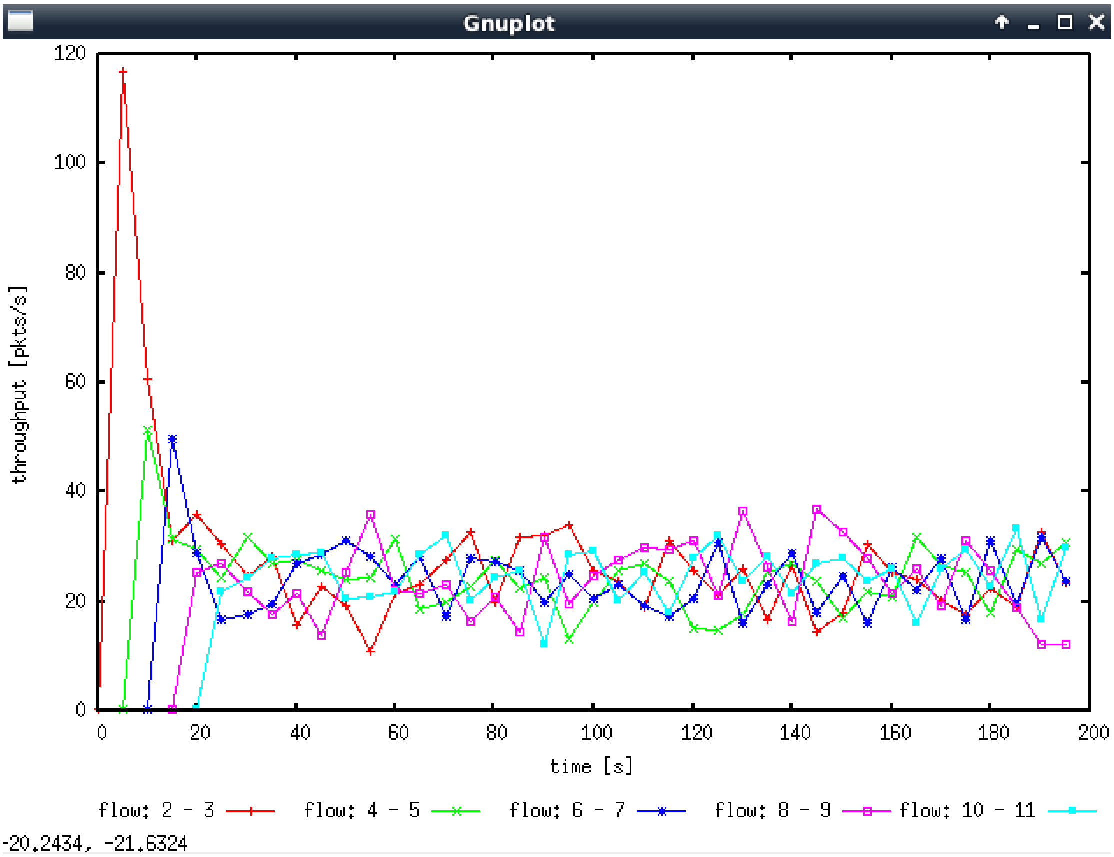
From the graph above, we can find that the sender halves it's current congestion window and increases it linearly until losses starts happen again and this repeats, which means most of the losses are due to triple duplicate ACKs. This is because when timeout appears, the window size will reduce to 1 and then begin slow start phase.



From the graph above, we can find that the throughput of TCP Reno is around 200 packets per second, the throughput of TCP Tahoe is around 190 packets per second. The throughput of TCP Reno is a little bit higher since it does not need to enter slow start phase after each congestion event.

Exercise 2: Flow Fairness with TCP

Question 1:



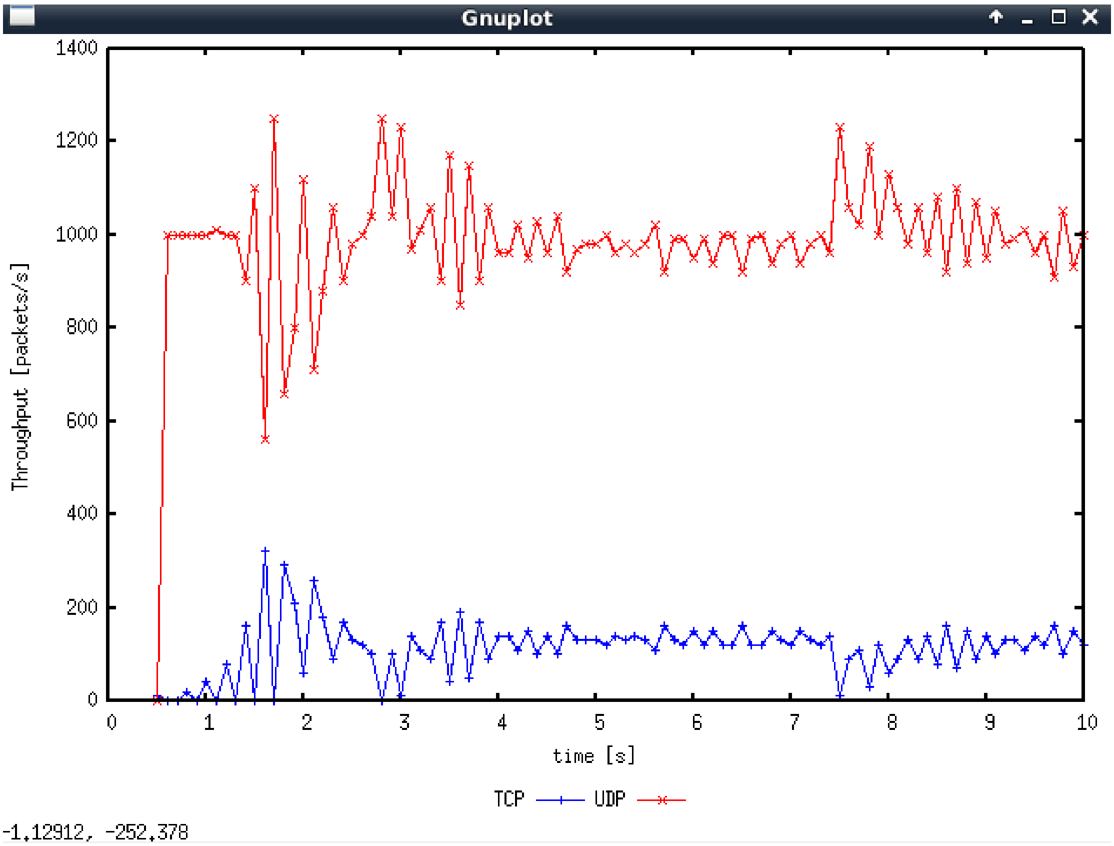
The throughput for all 5 connections is quite similar, this indicates that each flow gets an equal share of the capacity of the shared link. This approximate fair behavior is a direct result of the AIMD congestion algorithm used by TCP.

Question 2:

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 takes 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: TCP competing with UDP

Question 1:



UDP flow should has higher throughput than TCP since it does not implement any congestion control, which means that UDP flow will not reduce its transmission rate if there is congestion. As shown above, UDP achieves higher throughput than TCP, because TCP has congestion control but UDP does not, which indicates my expectation is correct.

Question 2:
UDP flow achieve higher throughput than TCP because the network congestion control will not influent UDP. In other words, UDP can have a stable transmission rate although packets loss could happen. In contrast, TCP congestion mechanism could decrease transmission rate due to network congestion.

Question 3:
Advantage: use UDP can sometimes increase transmission rate, since UDP does not affected by network congestion control mechanism, sender can send packets by a maximum speed.
Disadvantage: UDP is an unreliable data transfer protocol, so file transfer protocol running on UDP have to implement reliable data transfer, otherwise the transfer process won't work properly.
If everyone started using UDP instead of TCP, there will be a high possibility of

network paralysis because everyone can send packets by a maximum rate. This can cause congestion in a termless duration.