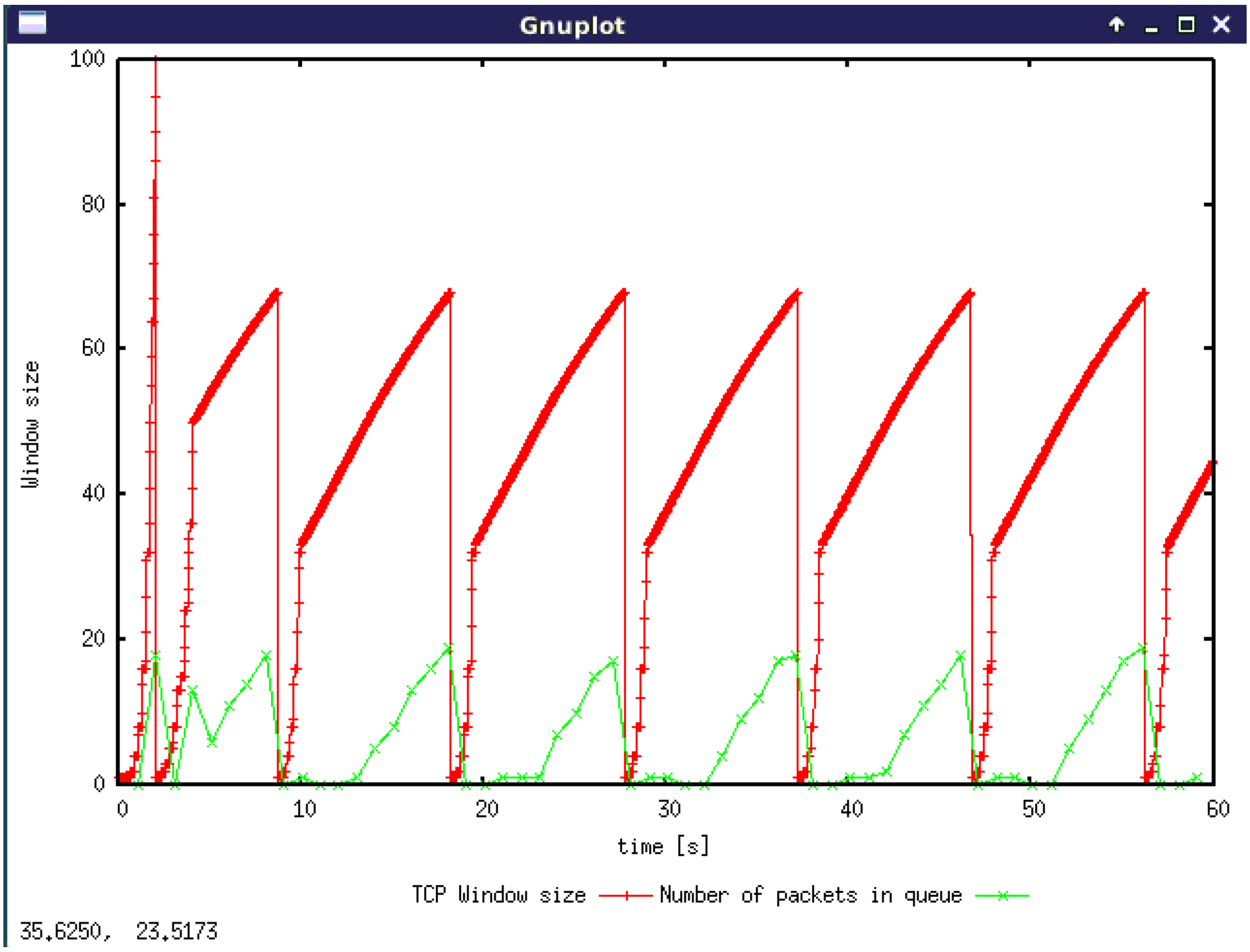
Exercise 1

Question 1



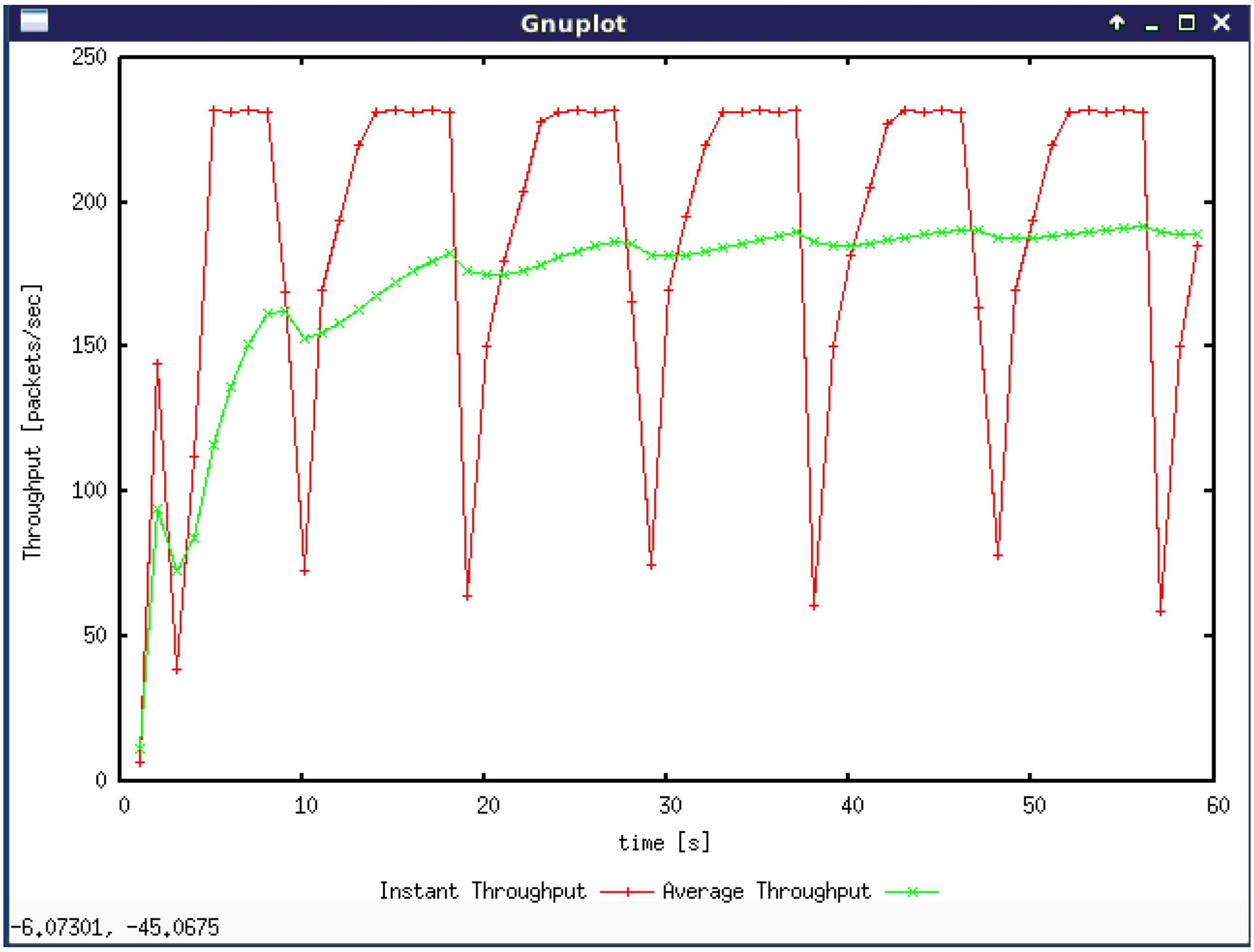
The maximum size of the congestion window that the TCP flow reaches in this case is 100. It could be concluded that the maximum size of the queue is 20, and all additional packets will be dropped.

When the congestion window reaches this value(100), the queue will be full, resulting in timeout or triple duplicate ACKs. The sender thus will reduce the congestion window to 1 and set ssthresh to 50 (100 / 2). The slow-start process will begin again. When the congestion window size increase to ssthresh, slow-start phase ends and the connection transitions to congestion avoidance phase. When the queue becomes full, which leads to packet loss, the connection will enter to slow-start phase again.

Question 2

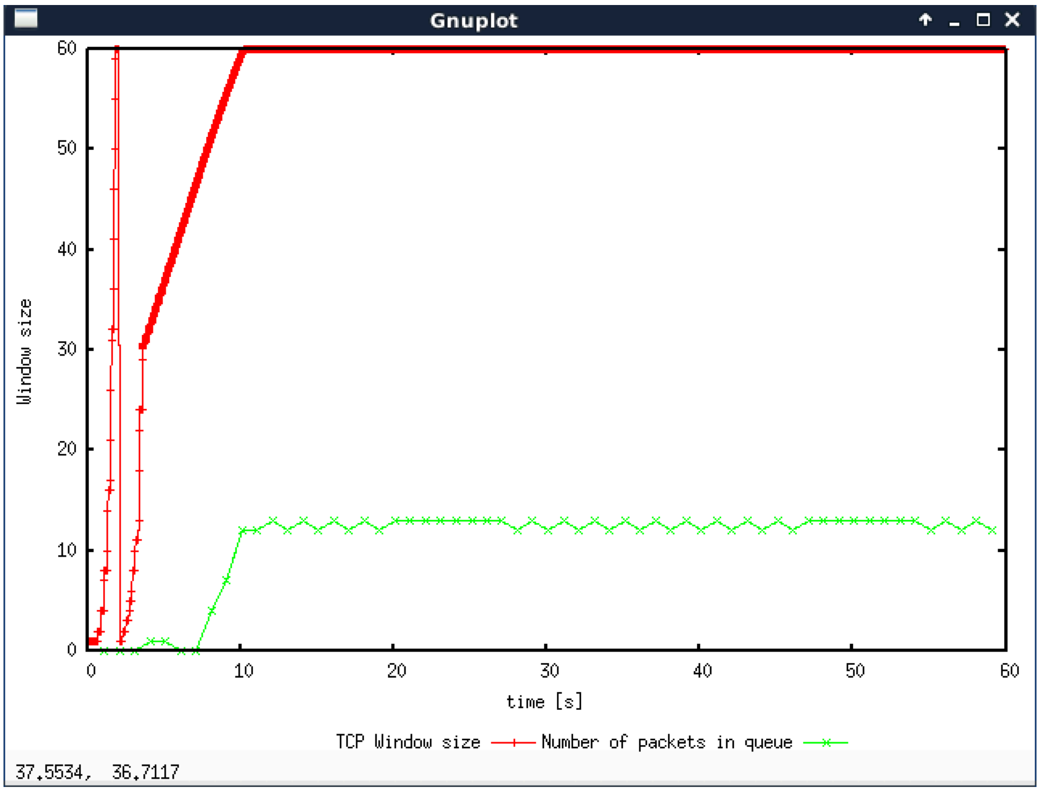
According to the picture below, the average throughput of TCP is 190 packets/sec.

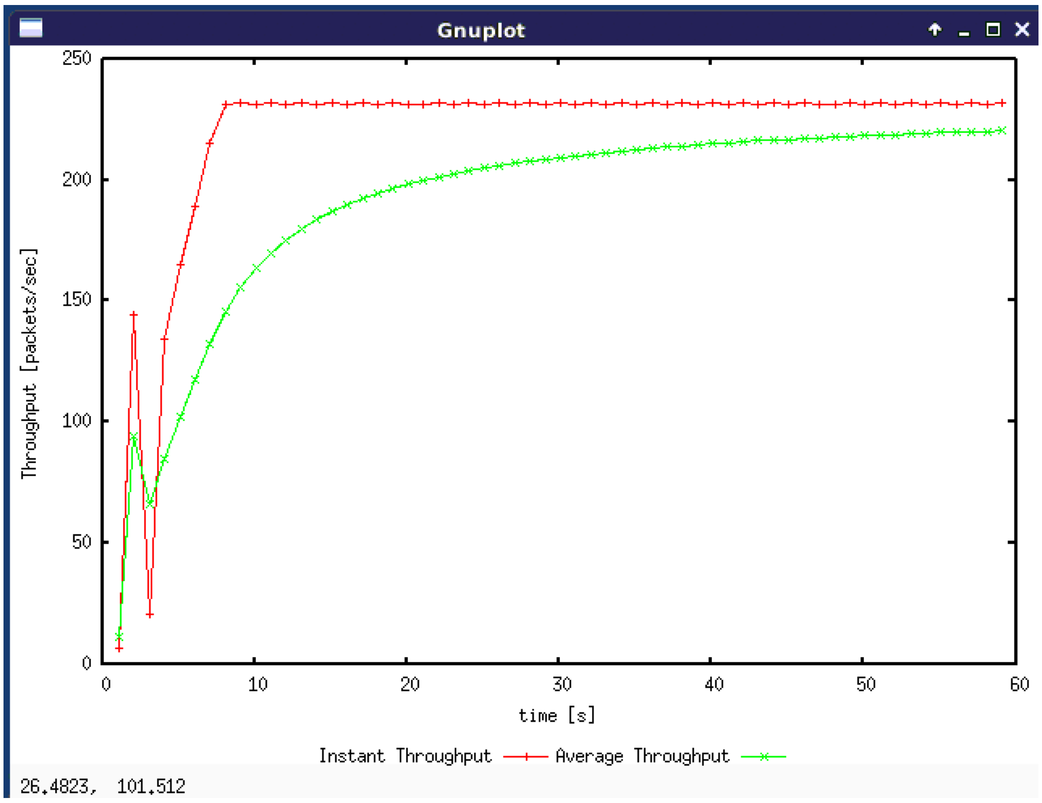
Which means the average throughput is 190 * (500+40) * 8 = 820.8 Kbps



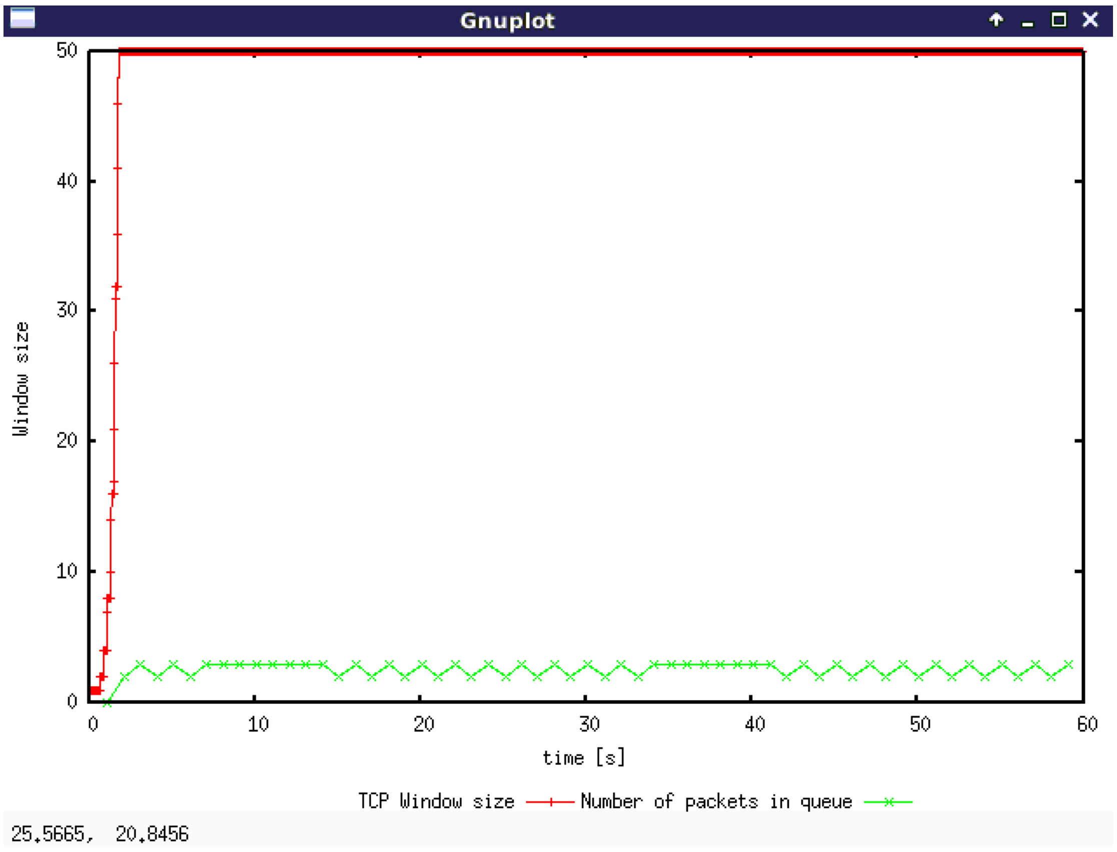
Question 3

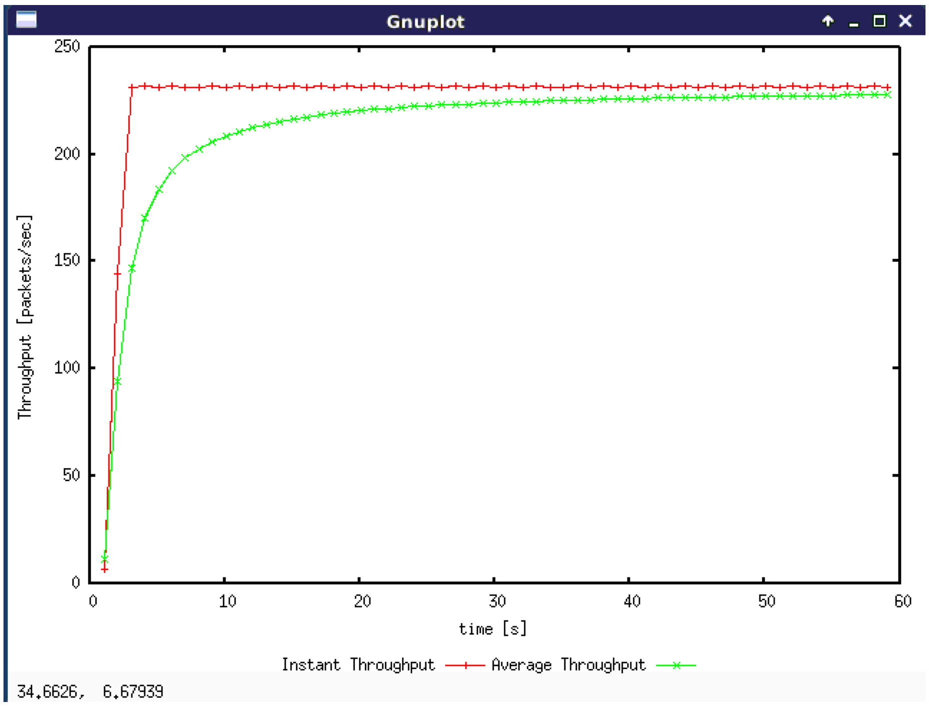
Max congestion window size = 60:





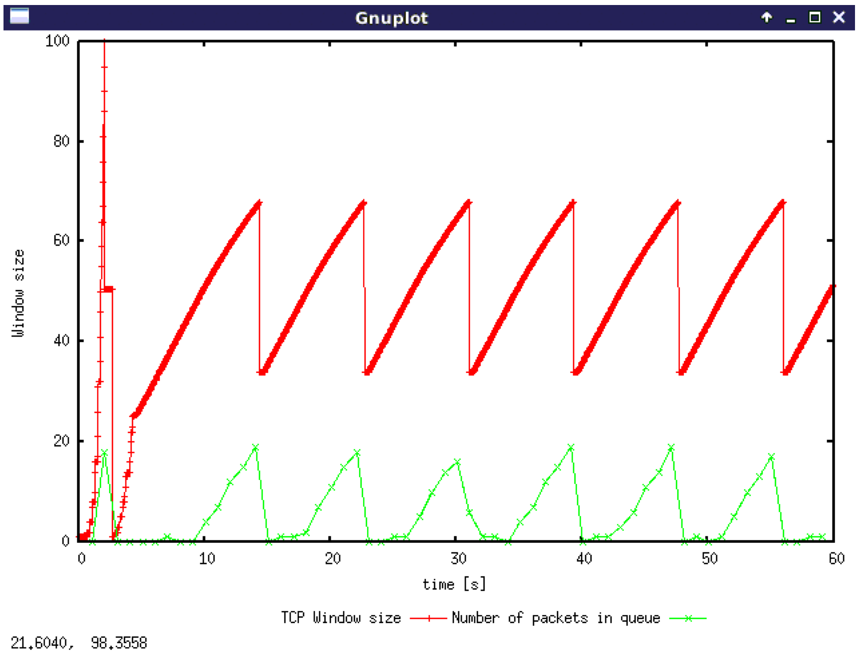
Max congestion window size = 50:

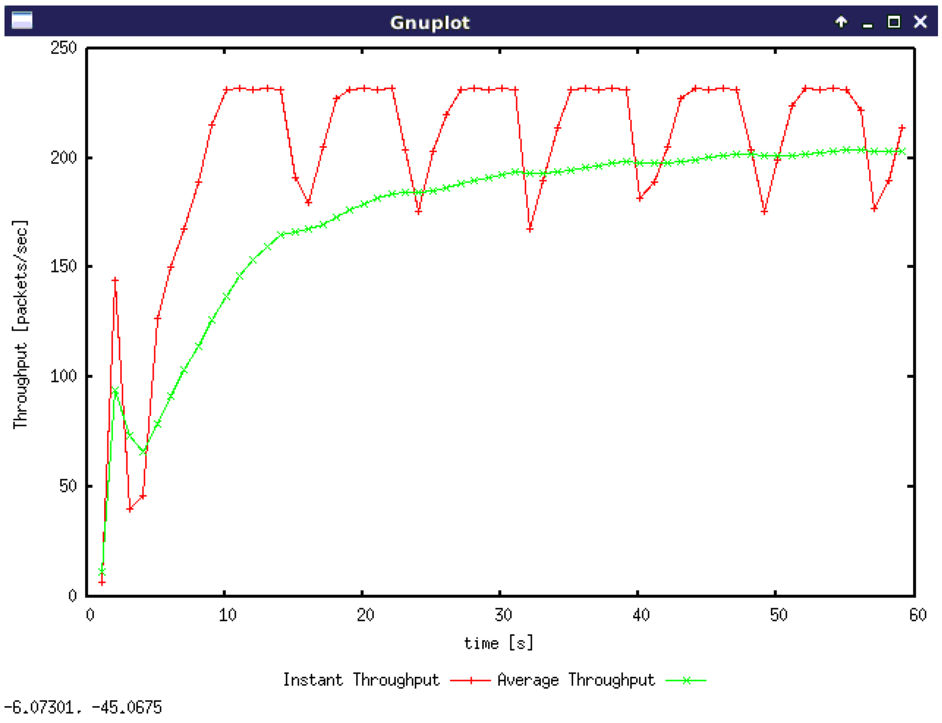




When the cwnd is greater than the maximum queue size, timeout or triple duplicate ACKs would happen, and TCP will set up the slow-start phase.
The pictures above shows that when the congestion window size was set to 60(<66), TCP window size will decrease only once, which means the connection enters to slow-start phase only once(except the slow-start phase at the beginning).
When the cwnd was set to 50, TCP window size will increase and finally keep stable at 50. The average packet throughput is close to 225 packets/sec. The average throughput is 225*500 = 900Kbps, which is close to 1Mbps.

Question 4

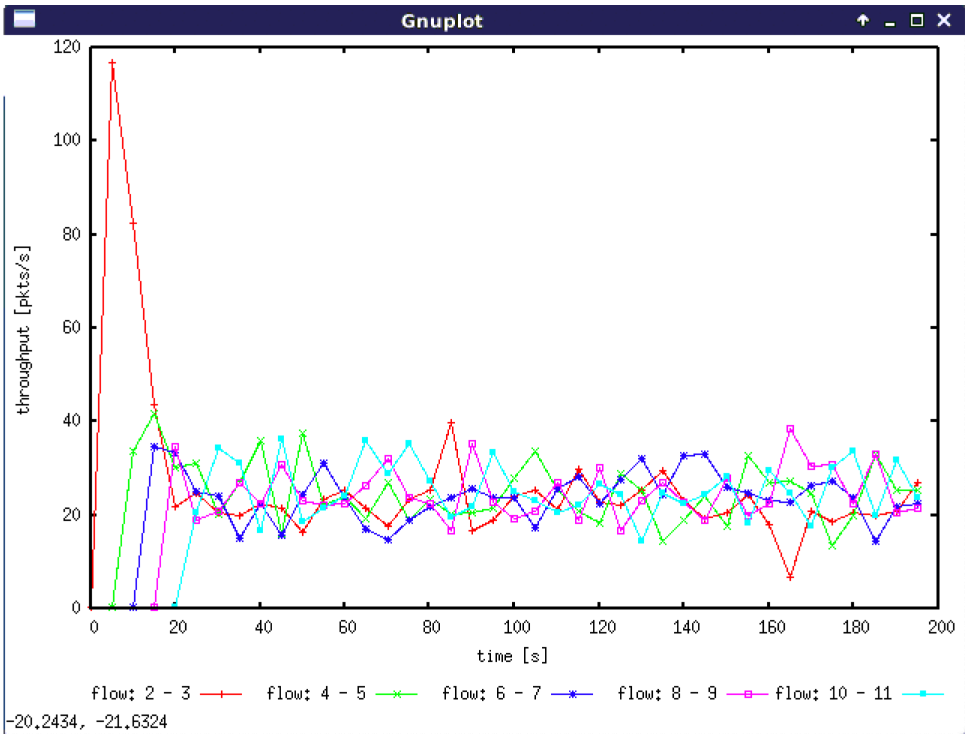




It could be found that except the first timeout event at the beginning, the TCP connection does not enter to slow-start phase. Instead, TCP window size was set to half the congestion window size and then increases linearly until packet loss happens again. But TCP Tahoe always gets back to 1.
The throughput of TCP Reno is 200 packets/sec, which is obviously greater than that of TCP Tahoe.

Exercise 2

Question 1



Despite some fluctuations in the graph, the five connections have similar throughput after all of them commence. The fluctuations may result from AIMD and slow start, but overall all flows share a bottleneck link.

Question 2

According to the graph above, when a new flow joins in, the throughput of all pre-existing flows will decrease. And when a new flow joins in, its window size will increase rapidly, which leads to many timeout and packet loss. In order to share the network, all existing TCP connections will change their window size to avoid overwhelming the network. This behavior should be fair, because when a new flow joins in, the fair share of all existing flows should reduce together.

Exercise 3

Question 1

TCP provides congestion control, while UDP does not. So a UDP flow cannot reduce its transmission rate when congestion happens. Thus it could be concluded that UDP would keep the same transmission rate, but TCP will adapt its transmission rate.

Question 2

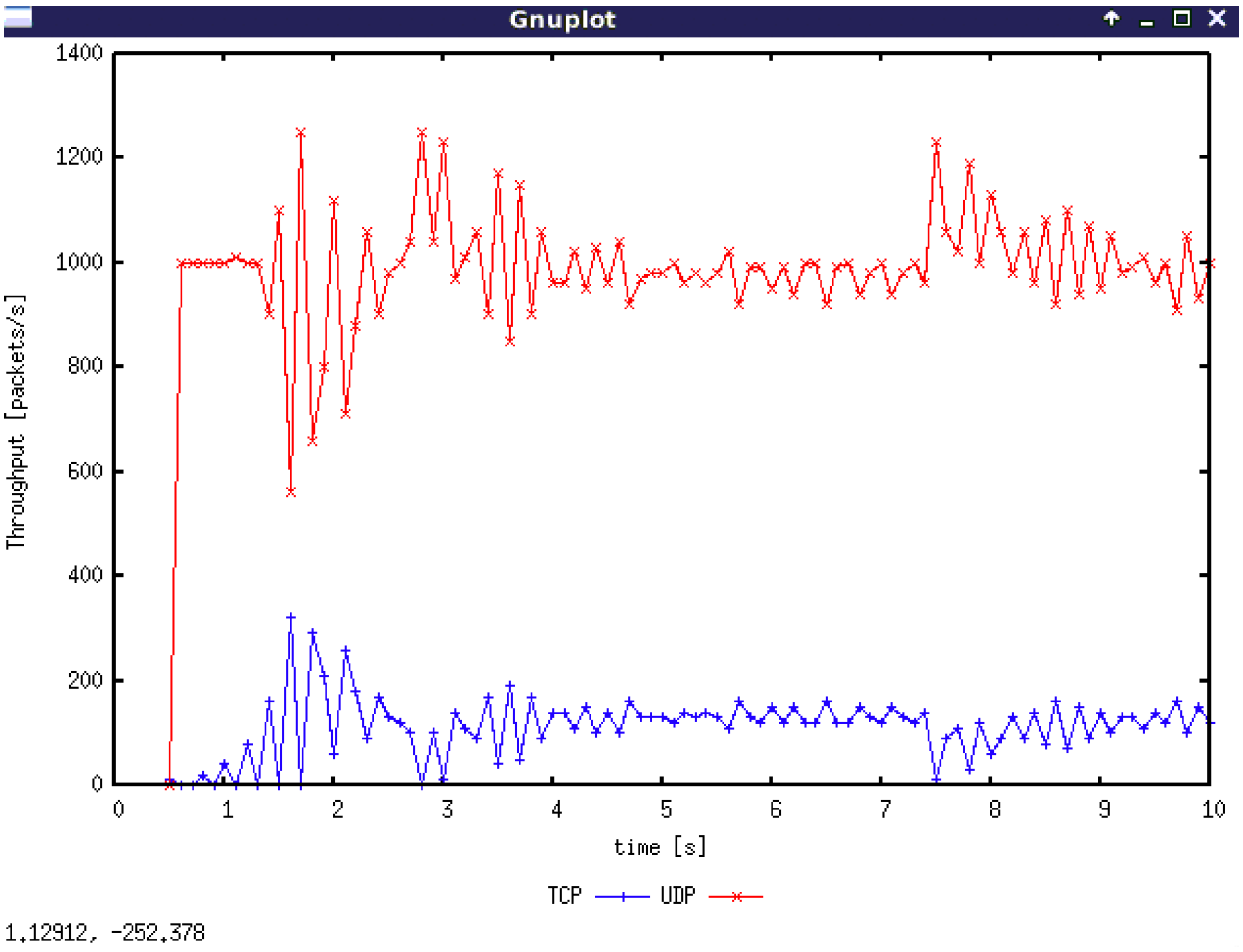
Because UDP does not provide congestion control, it has higher throughput than TCP. For UDP, the transmission rate is stable, regardless the occurrence of congestion in the network. As for TCP, its congestion control will reduce its transmission rate when congestion is detected. Thus, UDP could occupy more network resources and TCP shares little. This is unfair for TCP.

Question 3

Advantage: the sender can ignore congestion and keep transmitting unrestrained, which may reduce the delay for transferring files.

Disadvantage: UDP does not provide reliable data transfer, thus this function should be implement in other layers.

If everybody started using UDP instead of TCP for this, when congestion happens, the flows will not adapt their rate. This means that some network may face terrible collapse.


Gnuplot
Throughput [packets/s]
time [s]
TCP
UDP