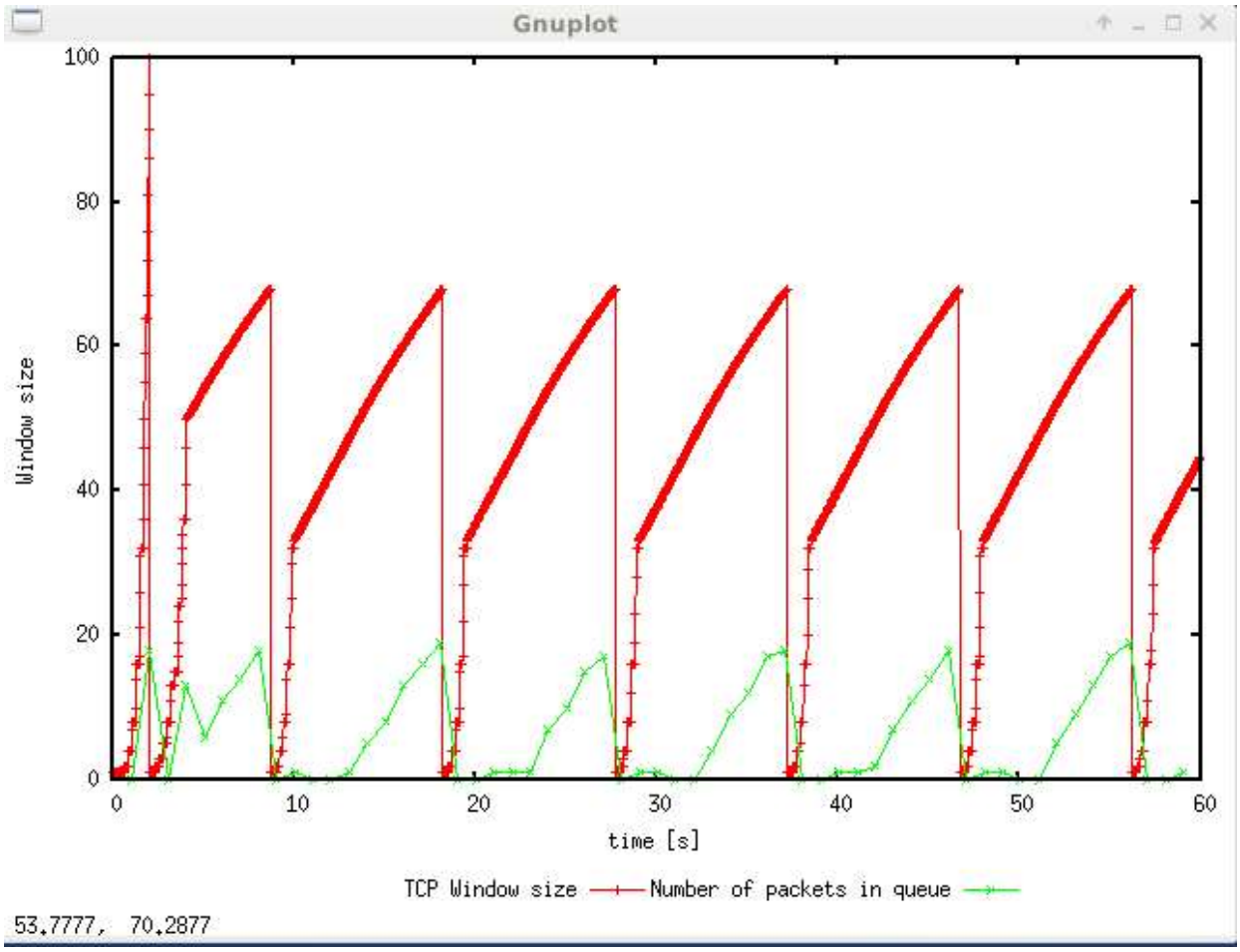
Exercise 1: Understanding TCP Congestion Control using ns-2

Question 1:

Answer:

In the beginning, threshold is 150 packets. But from the graph given below, the maximum size of the congestion window that the TCP flow reaches in this case is 100. When the congestion window reaches this value, the threshold set to half of the value of congestion window(around 50 packets) and congestion window set to 1 before new slow-start phase. Because congestion cause packet loss which result in timeout event happen.

The TCP flow enters slow start and increases rapidly (2^n exponential growth) until it reach the threshold. And then this connection enters congestion avoidance phase. Finally, the queue becomes full again, which creates packet loss. The connection will be back to slow start and it will repeat the process.



Question 2:

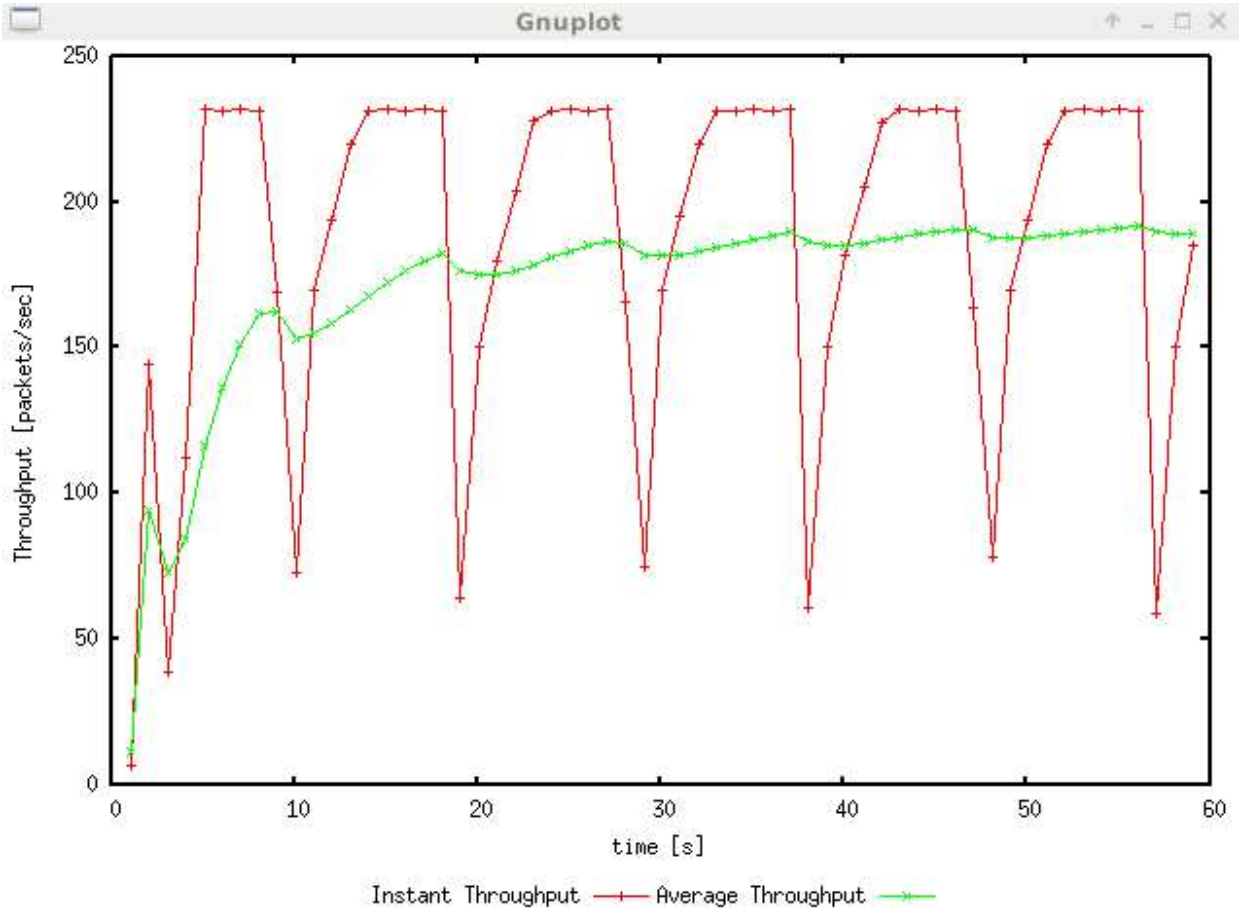
Answer:

The average throughput of TCP in this case is around 190 packets / s because after 20s, the throughput is becoming stable.

As for the average throughput of TCP in bps, it can be divided to two situations:

1.The rate at any data of TCP connection, which means that it includes header and payload data. Because both the header of TCP and IP is 20 bytes, so the throughput is 190 * (500 + 20 + 20) * 8 = 820.8 kbps

2.1. The rate at payload data, which means that it excludes the header, so the throughput is 190 * 500 * 8 = 760 kbps



-6.07301, -45.0675

Question 3:

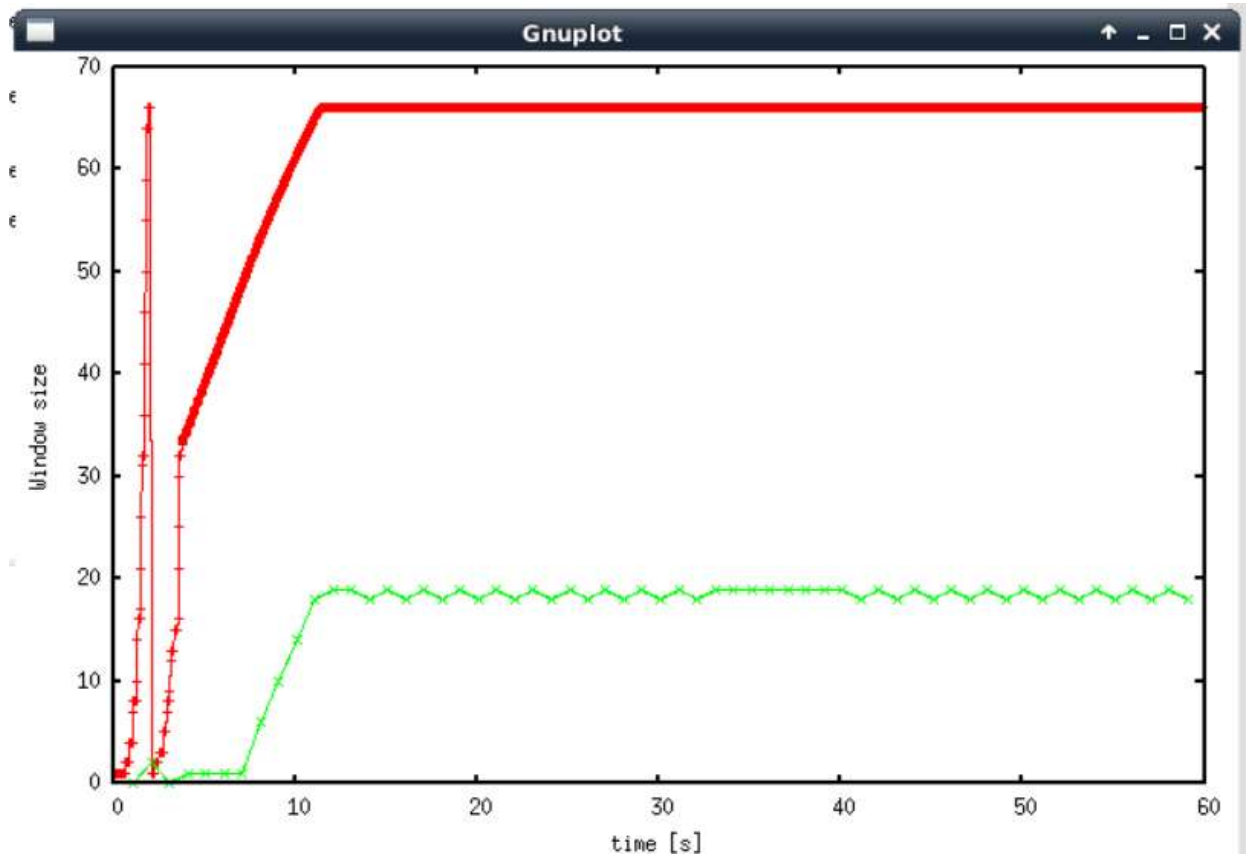
Answer:

After many attempts, we can find that:

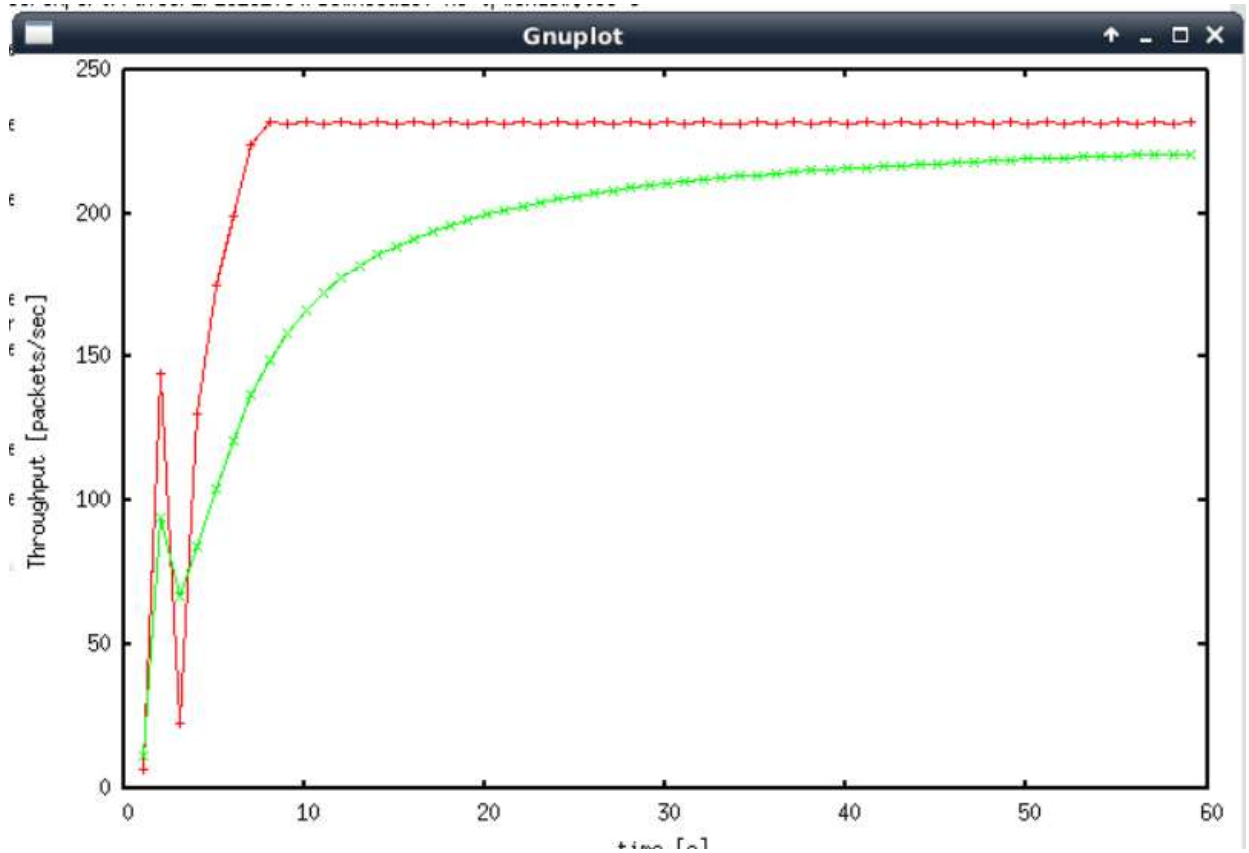
(1) when 50 < The initial maximum congestion window size <= 66, the oscillating situation stop after the first slow start phase. And it get a balance situation when the congestion window size reduces to half of its size. It means that the queue never gets full and packets are not dropped anymore.

(2) when the initial maximum congestion window size < 50, TCP gets balance situation after slow start phase.

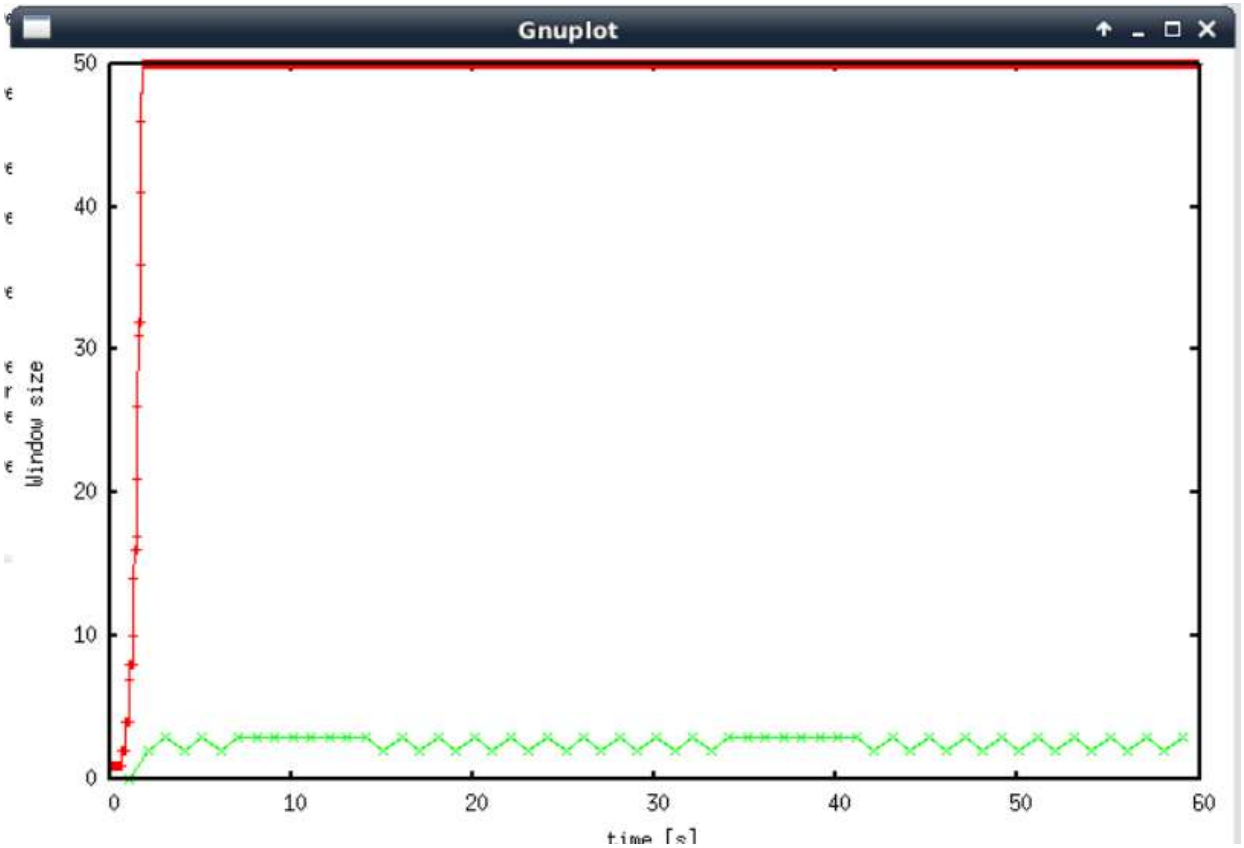
from the graph below, the average packet throughput is around 225 packets per sec , and we just neglect the TCP and IP header. So we can get that the The average throughput = 225 * 500 * 8 = 900 Kbps(don’ t consider TCP and IP header), the average throughput = 225 * 540 * 8 = 972 kbps(consider about TCP and IP header)



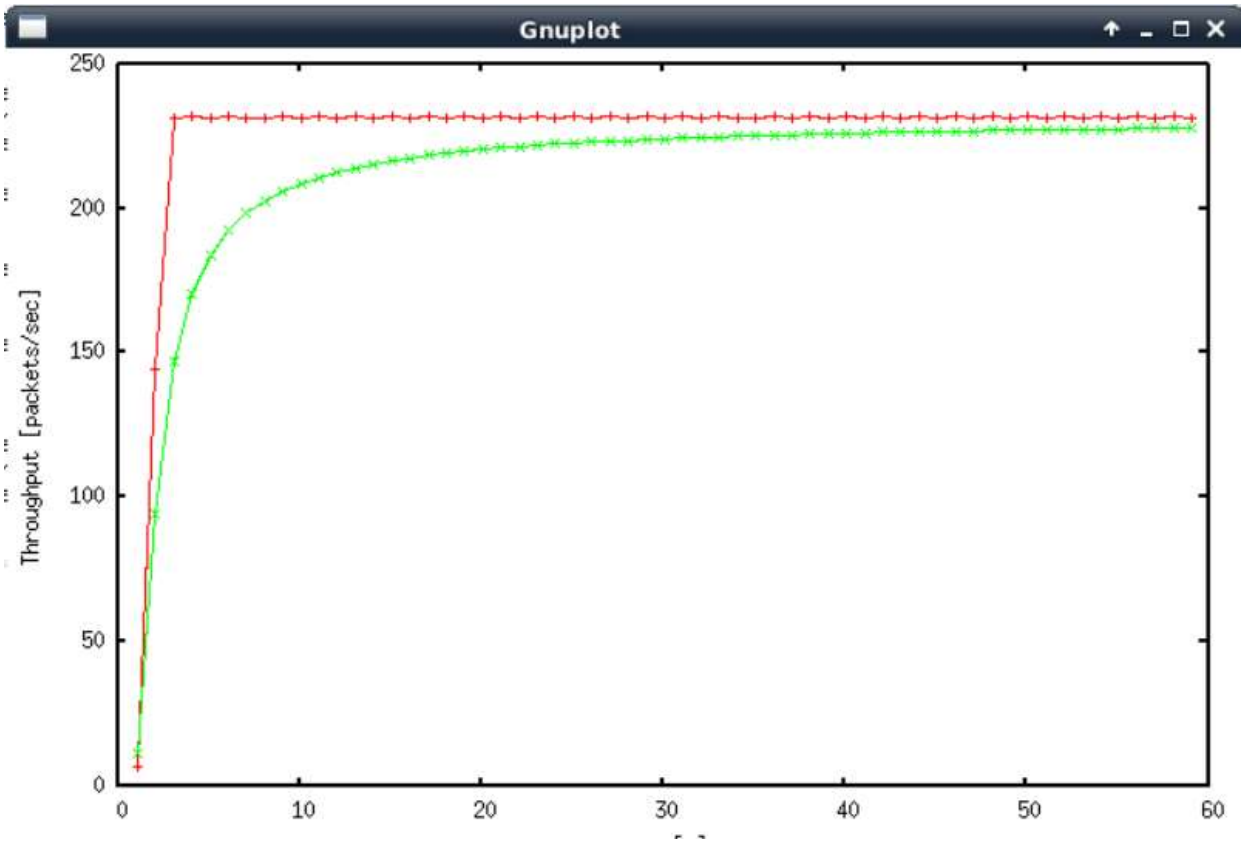
the graph above is windows change when congestion window = 66



the graph above is throughput when congestion window = 66



the graph above is windows change when congestion window = 50



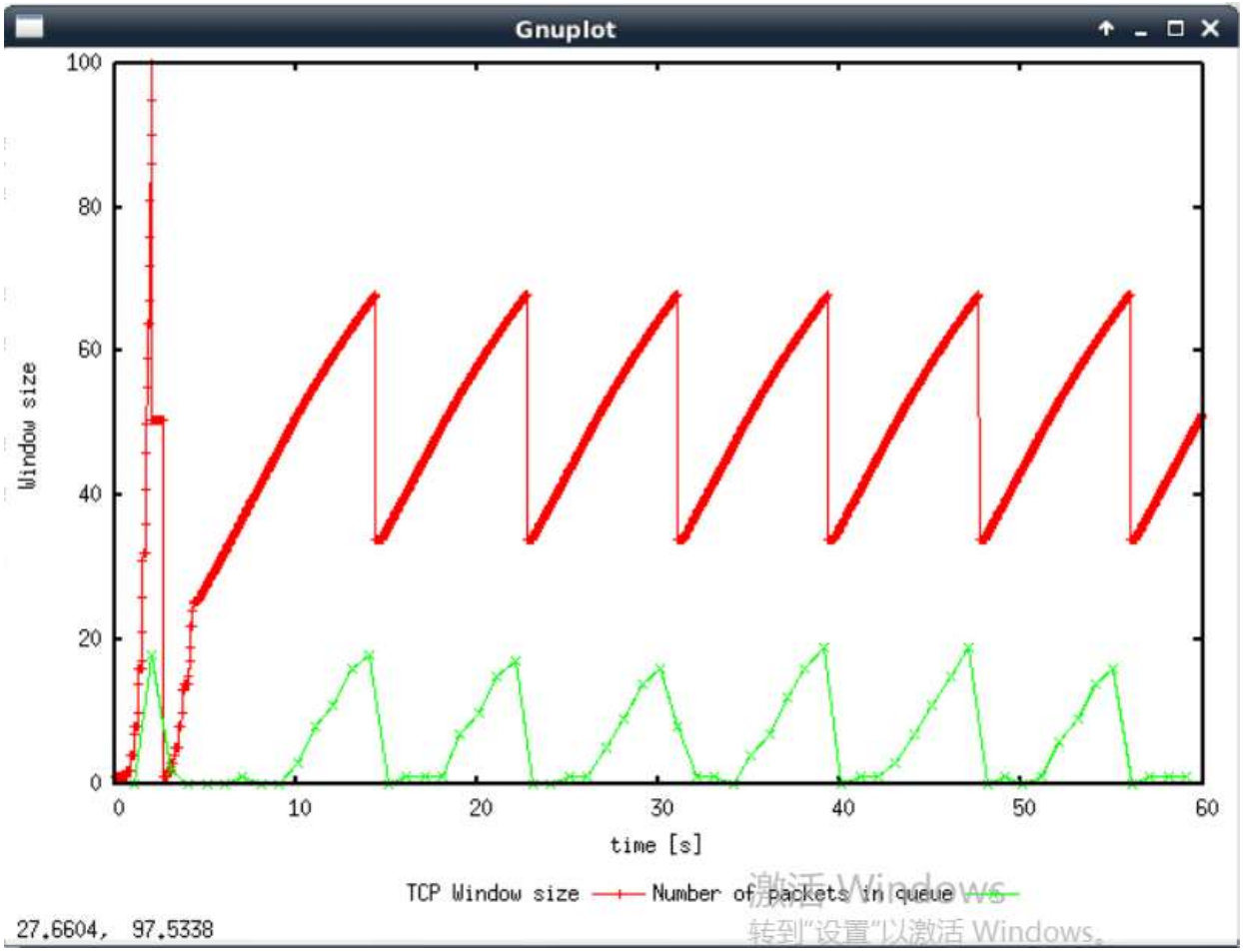
the graph above is throughput when congestion window = 50

Question 4:

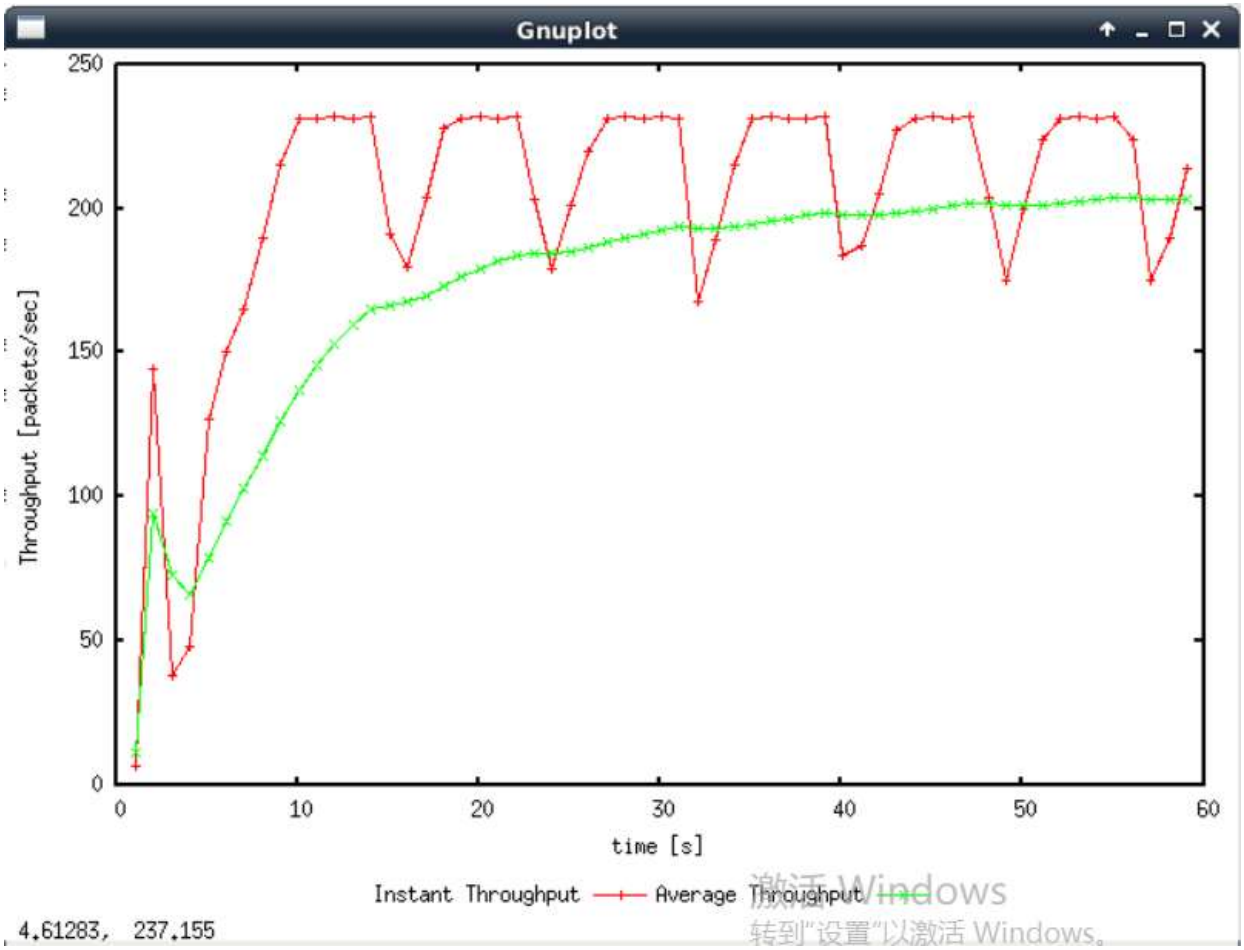
Answer:

From the graph below, we can find that the sender halves its current congestion window and increases its linearly, until losses starts happen again. And it repeats. This means most of the losses are detected due to triple duplicate ACKs (not timeout). When timeout appears, the window

size will reduce to 1 and then enter slow start phase. This behavior is different from TCP Tahoe whose window size is reduced to 1 after each congestion event.



The throughput of TCP Reno is around 200 packets per sec, which is higher than TCP Tahoe (190 packets per sec). This is because TCP Reno doesn't need to enter slow start phase after each congestion event .

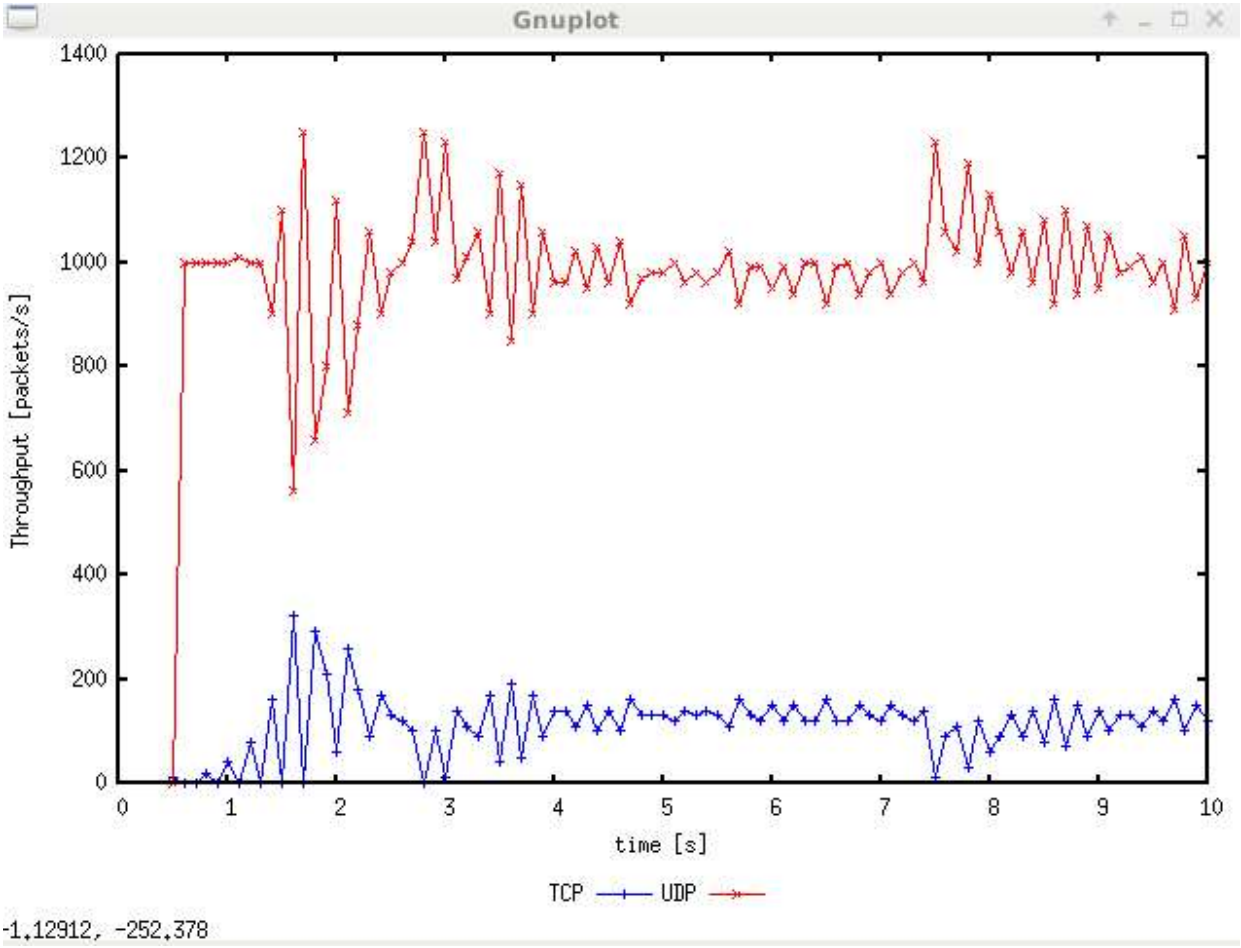


Exercise 3: TCP competing with UDP

Question 1:

Answer:

From the graph below, we can find that the throughput of UDP is larger than throughput of TCP if the capacity of link is 5Mbps.



Question 2:

Answer:

Because the network congestion control will not influent UDP protocol. So UDP could have a stable transmission rate although packets loss could happen. In constrast, TCP congestion mechanism could decrease transmission rate due to network congestion.

Question 3:

Answer:

Advantage: In some condition, It could increase the transmission rate. Because UDP don't have network congestion control, the sender can send packets by a maximum rate.
Disadvantage: it will increase the network burden. And since UDP is an unreliable data transfer protocol, the files transfer protocol running over the UDP have to use reliable data transfer.

If everybody started using UDP instead of TCP, it may cause network paralysis because everybody can send packets by unlimited rate.