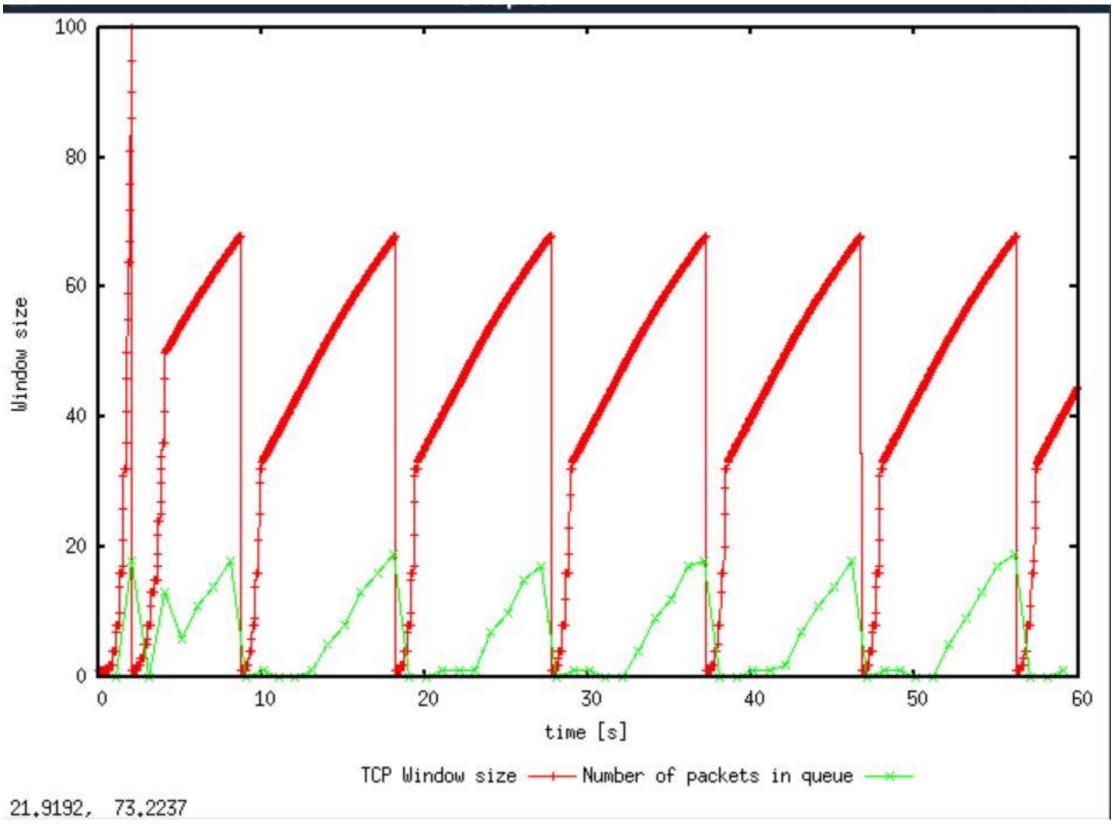
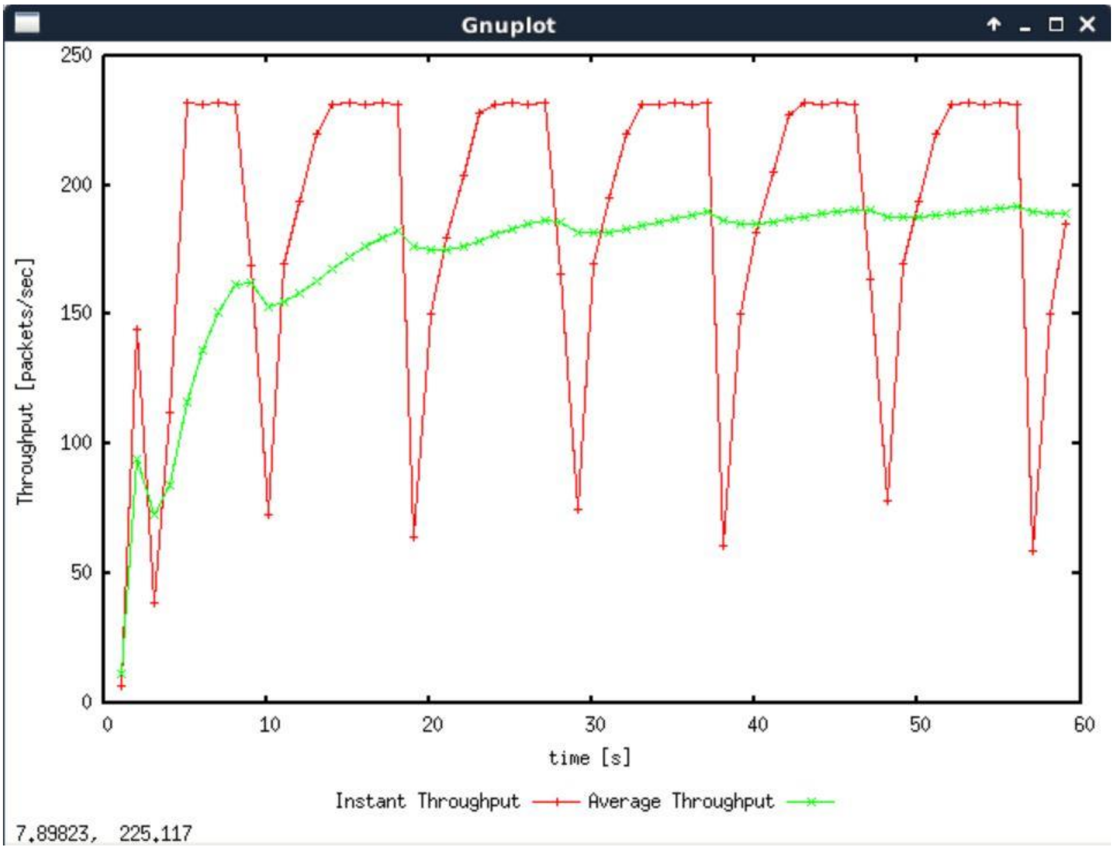
## Exercise 1

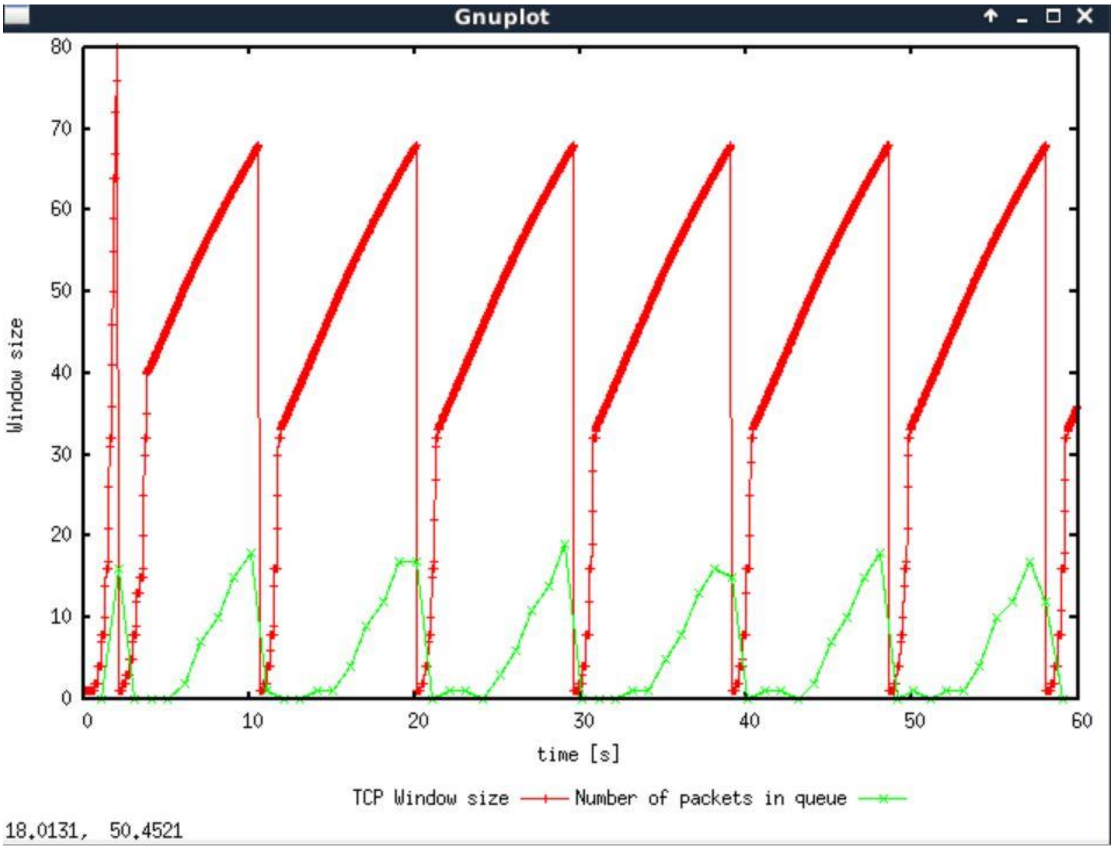
1. The maximum size of the congestion window that the TCP flow reaches 100. Then the window size decreased to 1. Because some packets were dropped in previous transmission. Then the packet window increased in slow-start way until window size was 50. Next the congestion window increased in linear way. This is because the mechanism Tahoe use. The threshold value was determined in first slow-start round. Since the maximum size of the congestion window was 100, the threshold value is 50. The window size will increase in linear way when it reached 50.



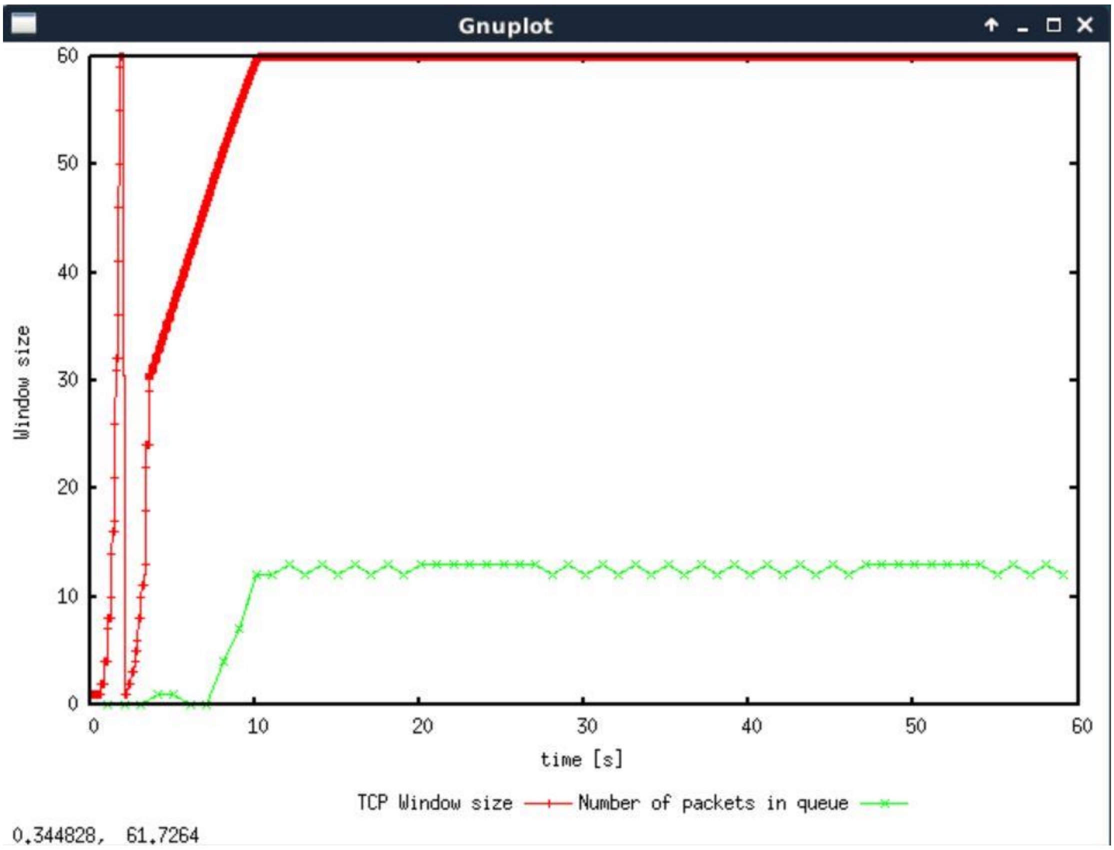
2. From the simulation script we used, we know that the payload of the packet is 500 Bytes. Keep in mind that the size of the IP and TCP headers is 20 Bytes, each. Neglect any other headers. The average throughput is about 190 packets per second. Considering every packet includes 500 bytes payload and 20 bytes IP header and 20 bytes TCP header, the throughput is 102600 bytes per second.



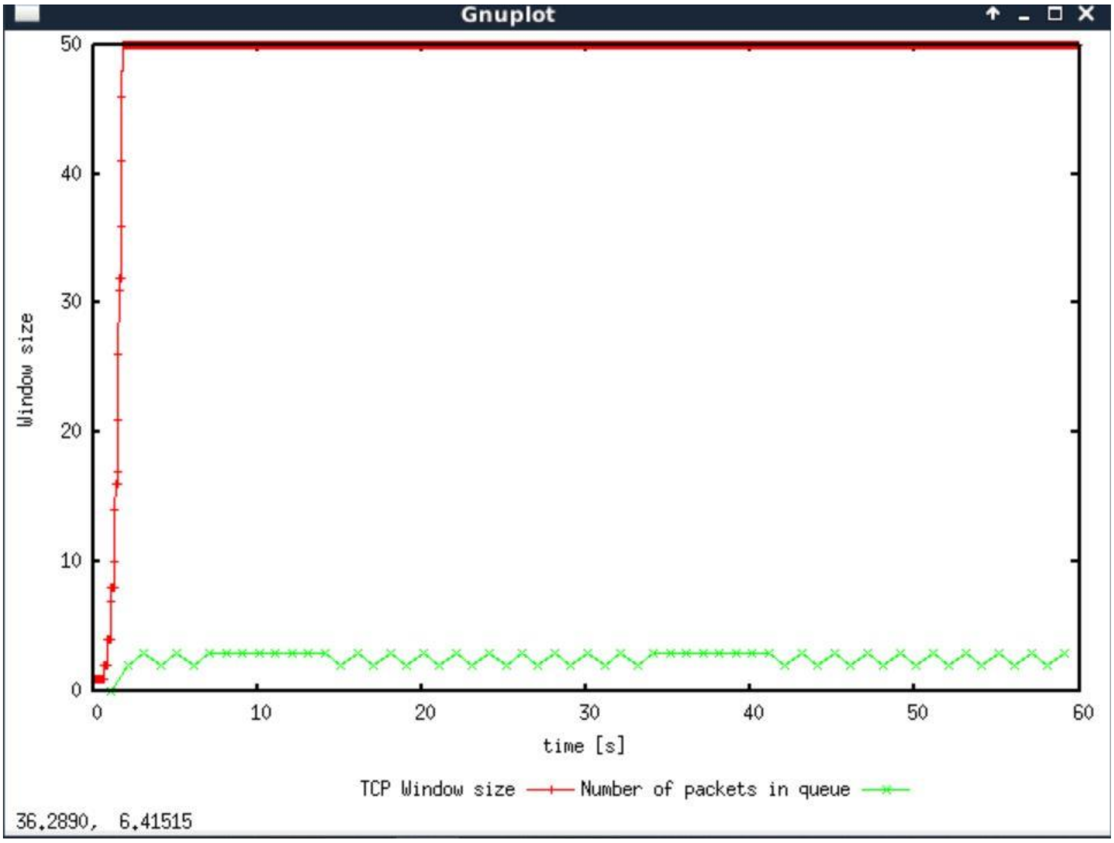
3. I tried 80 and 60 first. Though oscillation still appeared, they became more and more stable. When I tried 50, it stopped oscillating. To prove 50 is maximum size of congestion window, I tried 51. It oscillated again. So 50 is the maximum congestion window at which TCP stops oscillating to reach a stable behaviour. The average throughput is about 225 packets per second. Considering every packet includes 500 bytes payload and 20 bytes IP header and 20 bytes TCP header, the throughput is 121500 bytes per second. Compared to 1Mbps, which is 125000 bytes per second, the actual throughput is slightly smaller.



When congestion window is 80

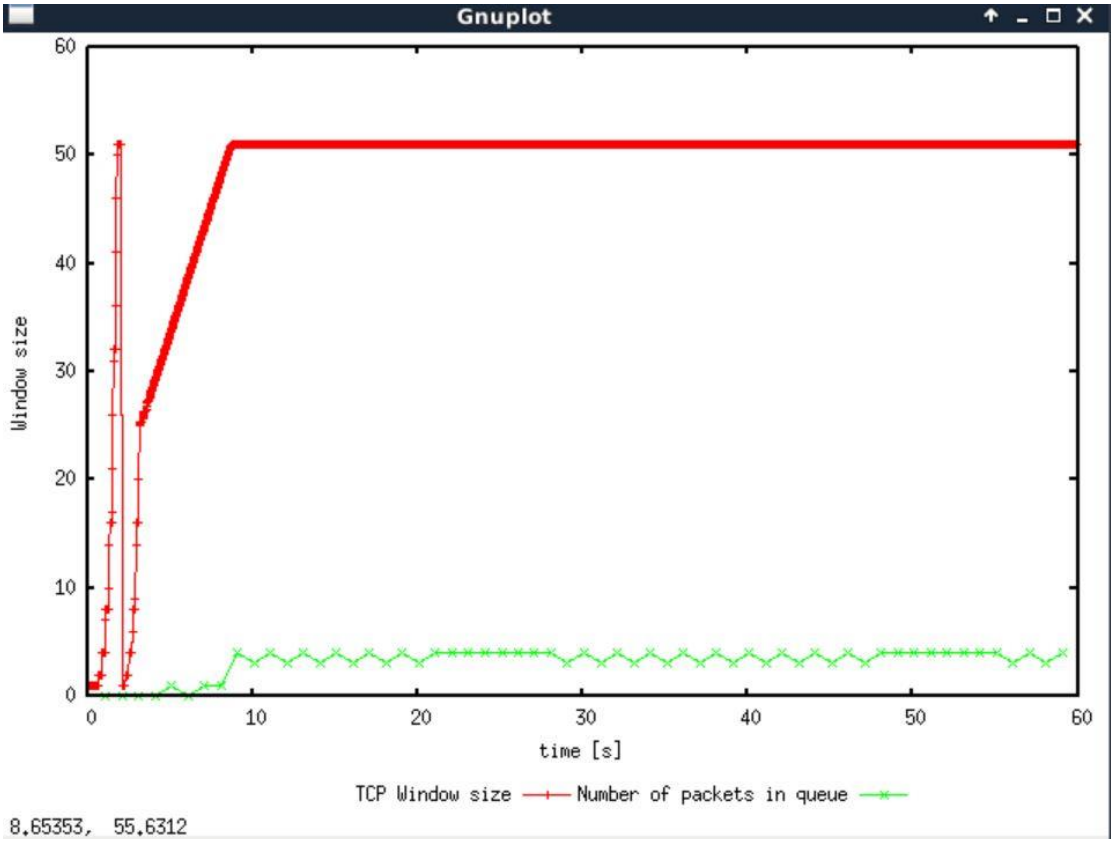


When congestion window is 60



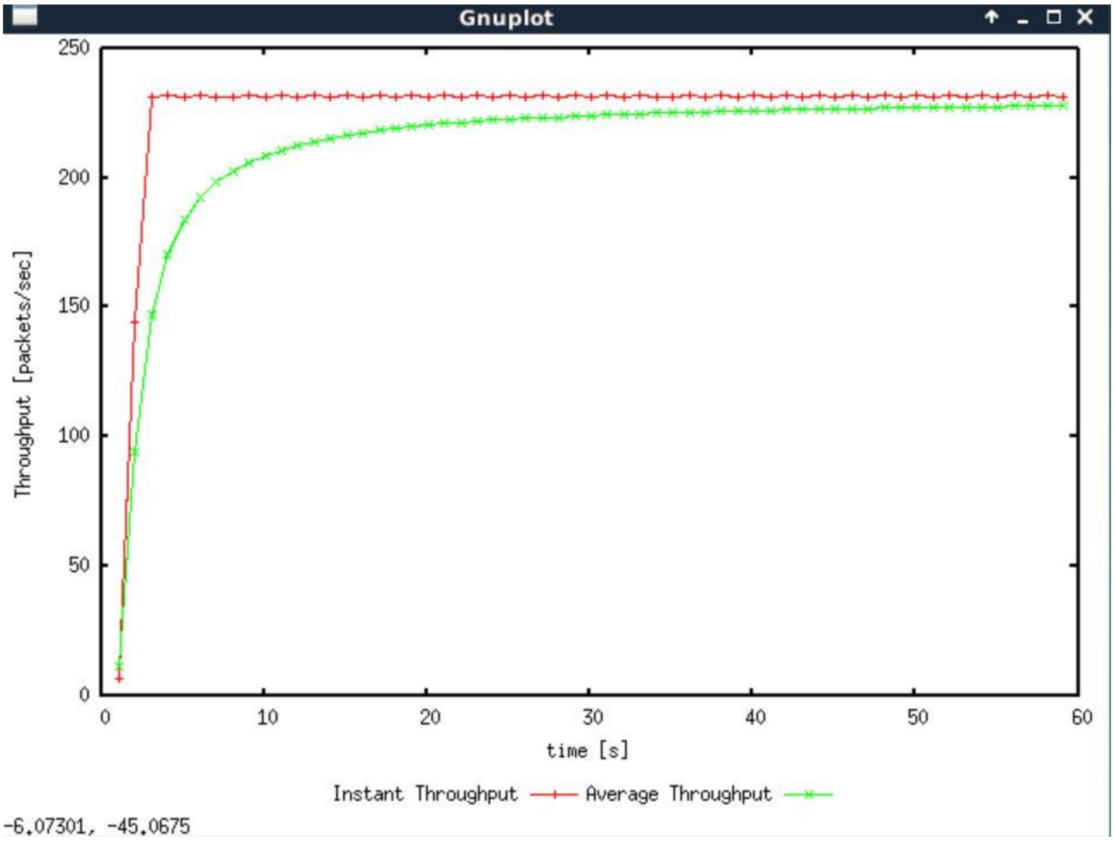
36.2890,  6.41515

When congestion window is 50



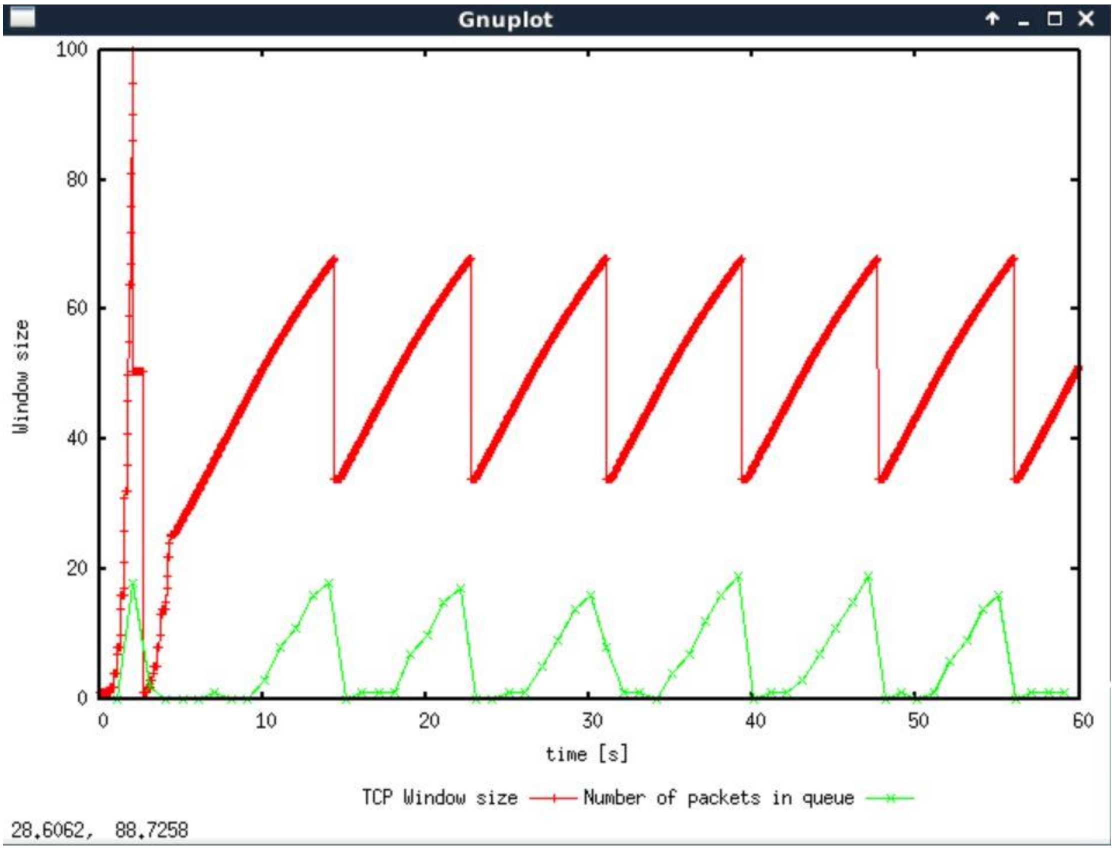
8.65353,  55.6312

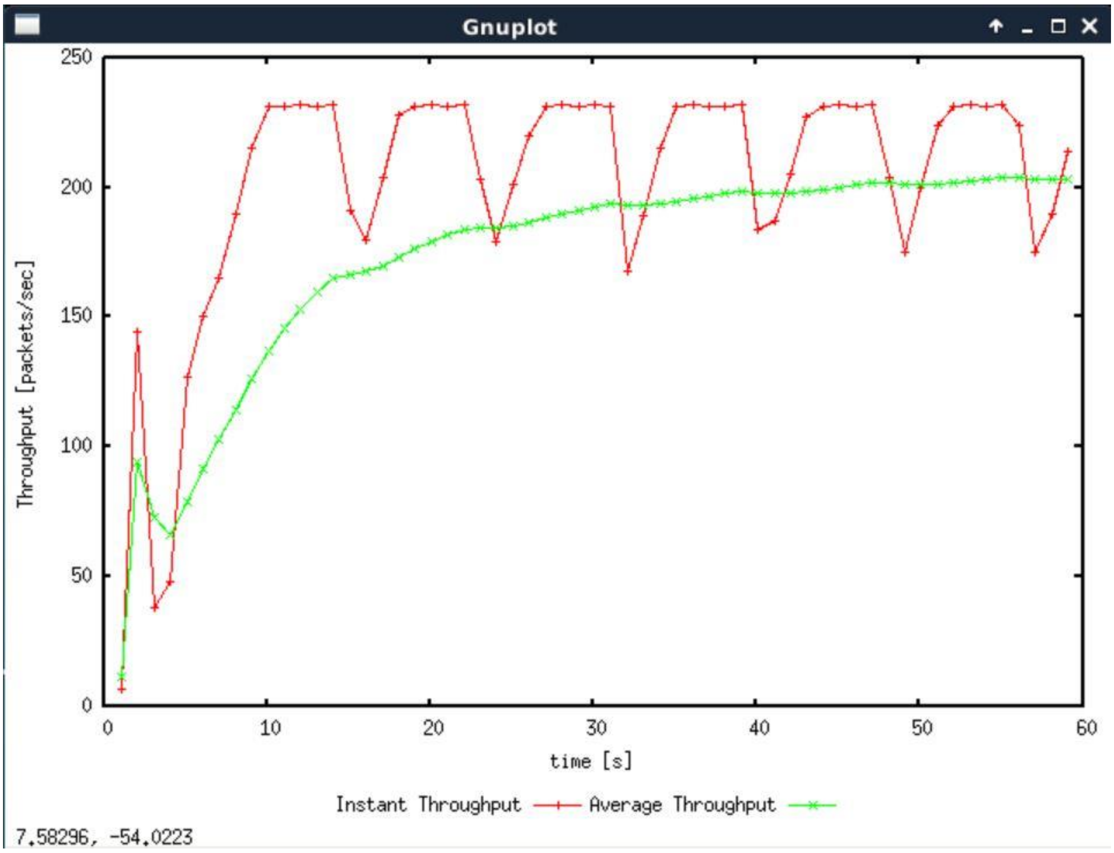
When congestion window is 51



Actual throughput when congestion window size is 50

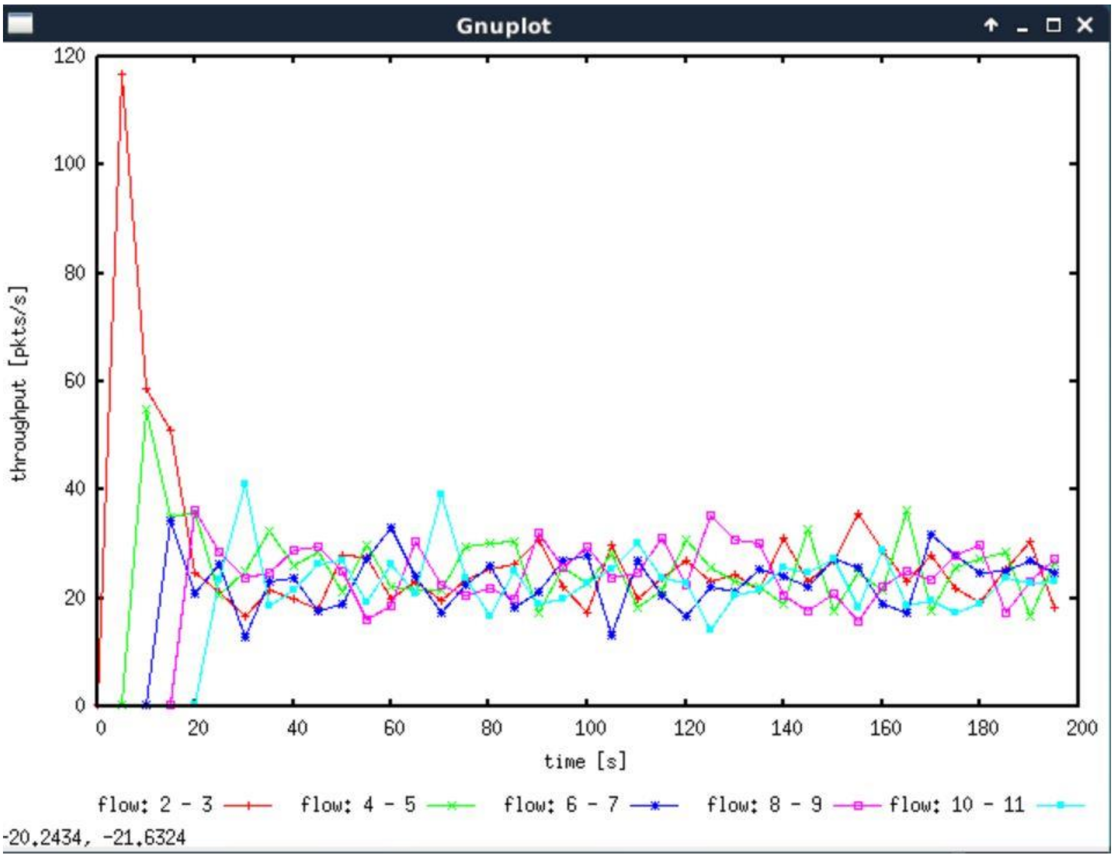
4. Unlike Tahoe, Reno do not decrease window size to 0 every times. Instead, it only decreases window size to 0 once, probably because timeout occurred. In other cases, it only halved window size when error, probably receiving duplicate ACKs, occurred. The average throughput is about 202 packets per second, which is larger than Tahoe.


Gnuplot
Window size
time [s]
0
10
20
30
40
50
60
0
20
40
60
80
100
TCP Window size
Number of packets in queue
28.6062, 88.7258


Gnuplot
Throughput [packets/sec]
time [s]
0
10
20
30
40
50
60
0
50
100
150
200
250
Instant Throughput
Average Throughput
7.58296, -54.0223

## Exercise 2

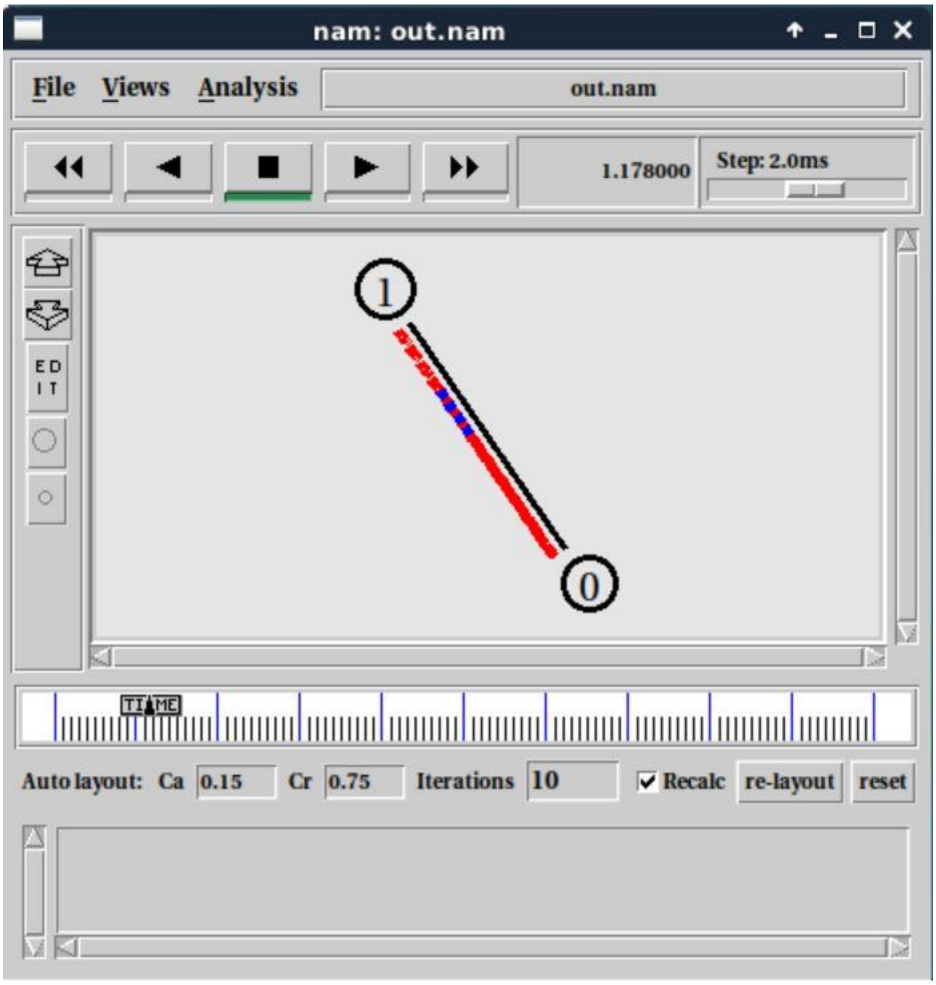
1. Yes. Flows get equal capacities of the common link. Though throughputs were different at beginning, they tended to be stable and equal over time.



2. When a new flow is created, the throughput of pre-existing flows decreases in order to share the common link. When a new flow start transmission in slow-start way, the congestion window size increases rapidly and occupy throughput. Therefore, throughput of other flows was sacrificed. This is fair since the throughput of flows should be determined by needs instead of created order.

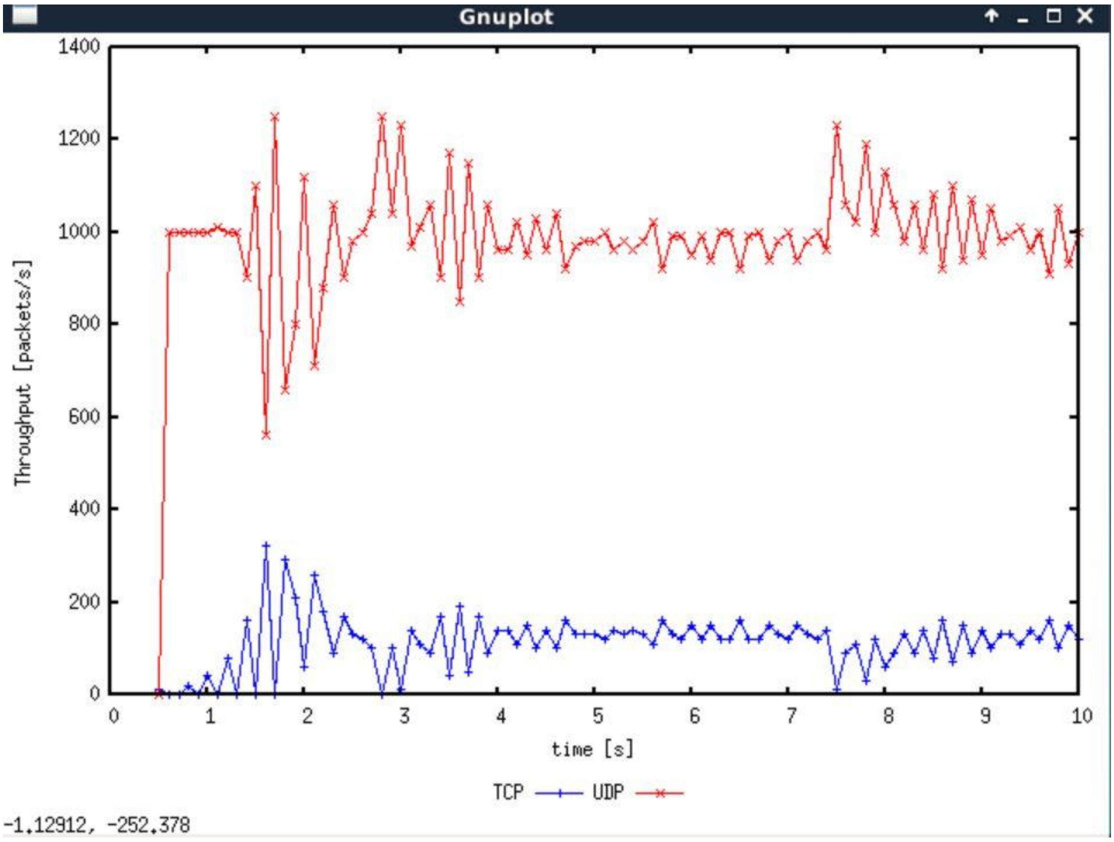
## Exercise 3

1. I expect the TCP flow act as it in above exercise (i.e., fluctuating) and the UDP flow simply sending packets as many as possible. TCP transmission has congestion control. So, it gradually sent more and more packets, which was blue flow's behavior. On the other hand, the red flow kept sending packets, which should be UDP's behavior.



At the beginning, blue sent few packets while red kept sending packets

2. TCP transmission use congestion control and connection to achieve stability. So, it gradually sent more and more packets, and would decrease window size when errors happen. Thus, the throughput of TCP is always relatively low. On the other hand, UDP do not provide any stability and kept sending packets no matter what happens. Therefore, the throughput of UDP is always relatively high.



3. when multiple flows are using one link, throughput was limited. Using UDP can guarantee maximum throughput was used, while the possibility of loss of packets and errors exist. Besides, we cannot handle with errors by using UDP. If everybody start using UDP instead of TCP, though bandwidth can be more fully utilized, more packets would lose, important information would not be guarantee to be received, the Internet become unstable.