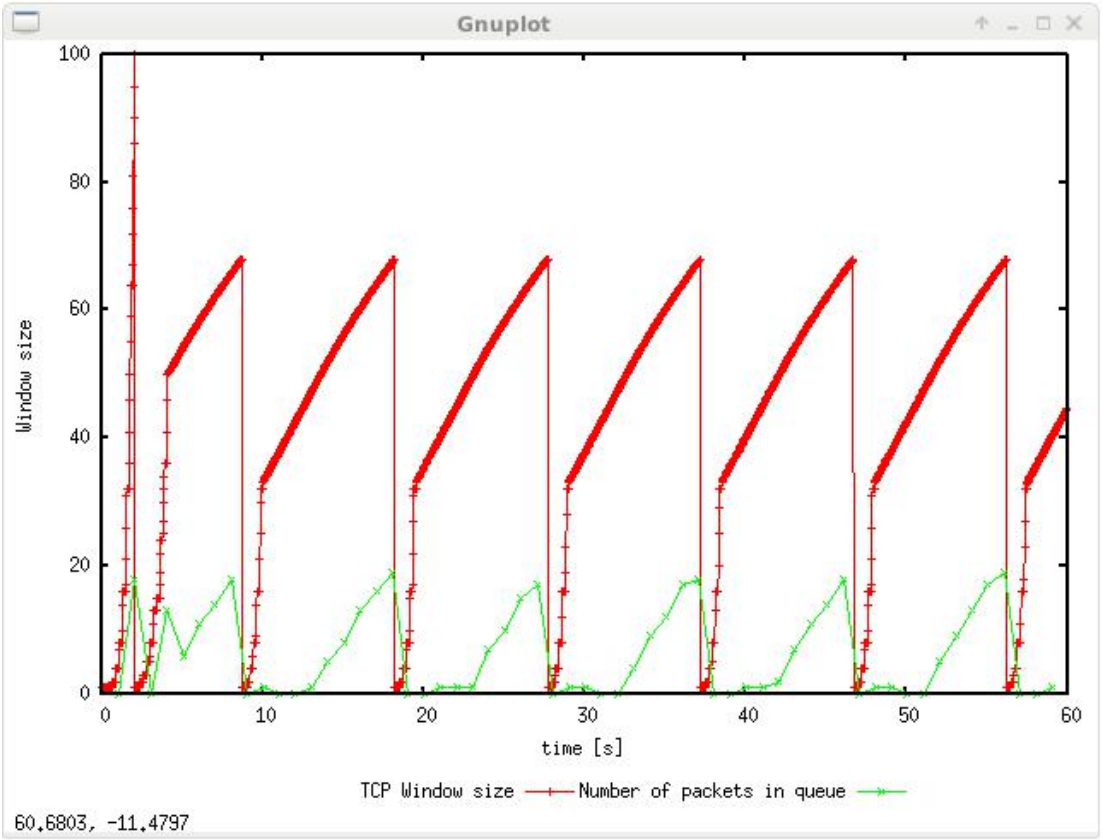
Student ID:z5195715 Name:Junyu Ren

Lab5

Exercise 1: Understanding TCP Congestion Control using ns-2

Question 1: What is the maximum size of the congestion window that the TCP flow reaches in this case? What does the TCP flow do when the congestion window reaches this value? Why? What happens next? Include the graph in your submission report.



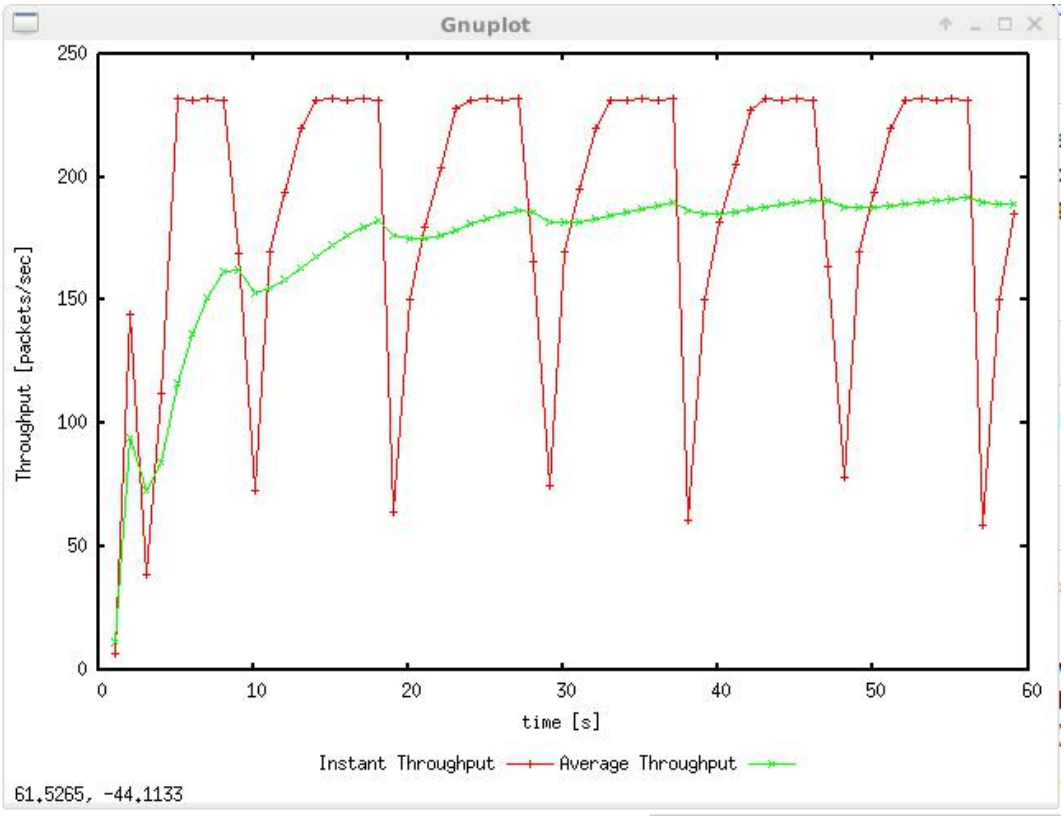
Maximum size of congestion window is about 100 packets.

When the congestion window reaches 100,the queue gets full because maximum queue capacity is 20. Packets then get dropped and congestion happens.The sender will cut down the window size to 1 and set the threshold to half of the original window size.This is determined by TCP Tahoe mechanism.

Next,the sender starts slow start and the window size increases until it reaches the threshold.It will get into congestion avoidance state with window size increase linearly until packets loss again.

Question 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

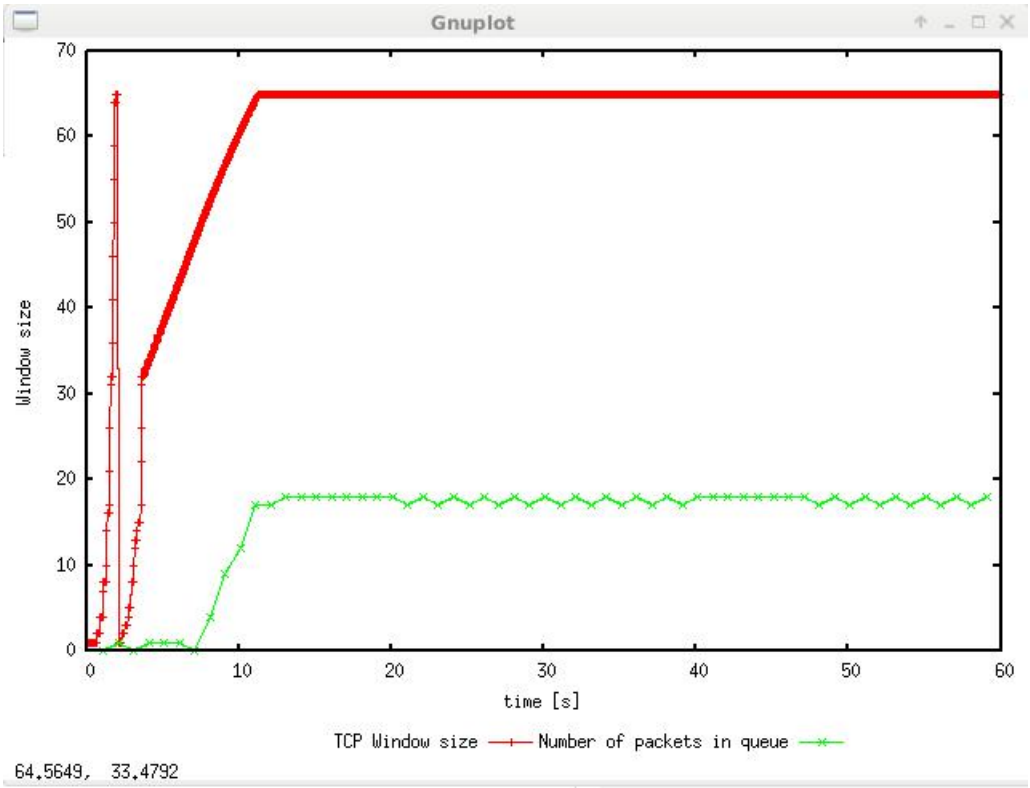
headers. What is the average throughput of TCP in this case? (both in number of packets per second and bps)



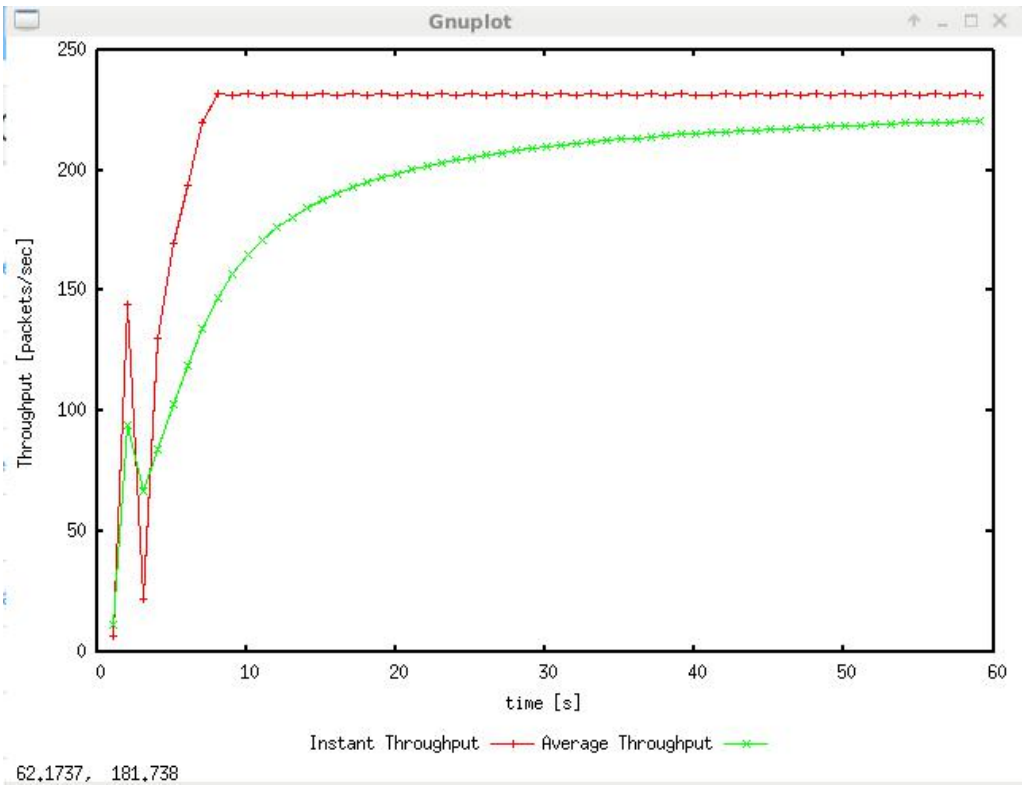
From the graph,the average throughput is roughly 190 pkt/s.
Every packet size = playload + IP header + TCP header = 500 + 20 + 20 = 540 bytes = 540 * 8 = 4320 bit
Average throughput (in bps) = 4320 * 190 = 820800 bps

Question 3: Rerun the above script, each time with different values for the max congestion window size but the same RTT (i.e. 100ms). How does TCP respond to the variation of this parameter? Find the value of the maximum congestion window at which TCP stops oscillating (i.e., does not move up and down again) to reach a stable behaviour. What is the average throughput (in packets and bps) at this point? How does the actual average throughput compare to the link capacity (1Mbps)?

When the parameter is bigger than 65, TCP reduce window size to 1 and then slow start whenever congestion happens like the graph in Q1.



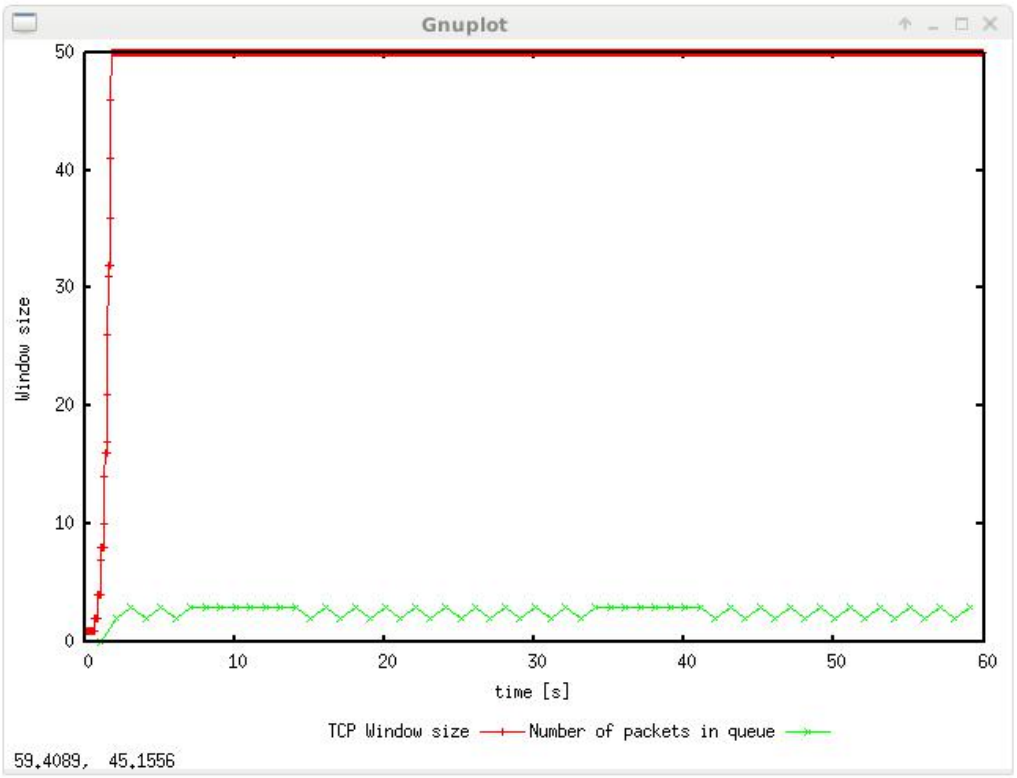
When the window size is 65, after the second slow start, TCP has an stable behavior which means the queue is never get full and no packet loss.



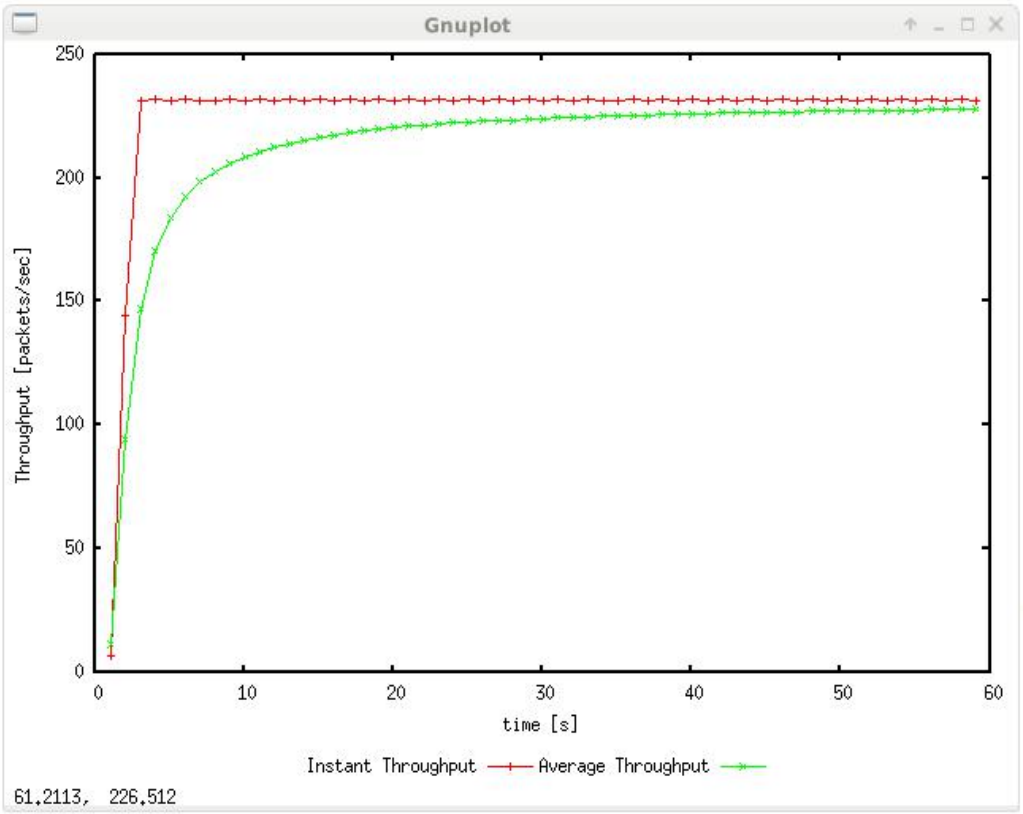
From this graph, average throughput for max window size 65 is about 220 pkt/s

Change it into bps = 220 * 540 * 8 = 950400 bps = 0.9504 Mbps (header included)

When the parameter is between 65 and 50, TCP's behavior is similar to max window size 65.



When the max window size is 50, just after first slow start, TCP goes stably.Moreover, the queue in this case is also relatively stable (although some minor fluctuations)and don't drop packets.

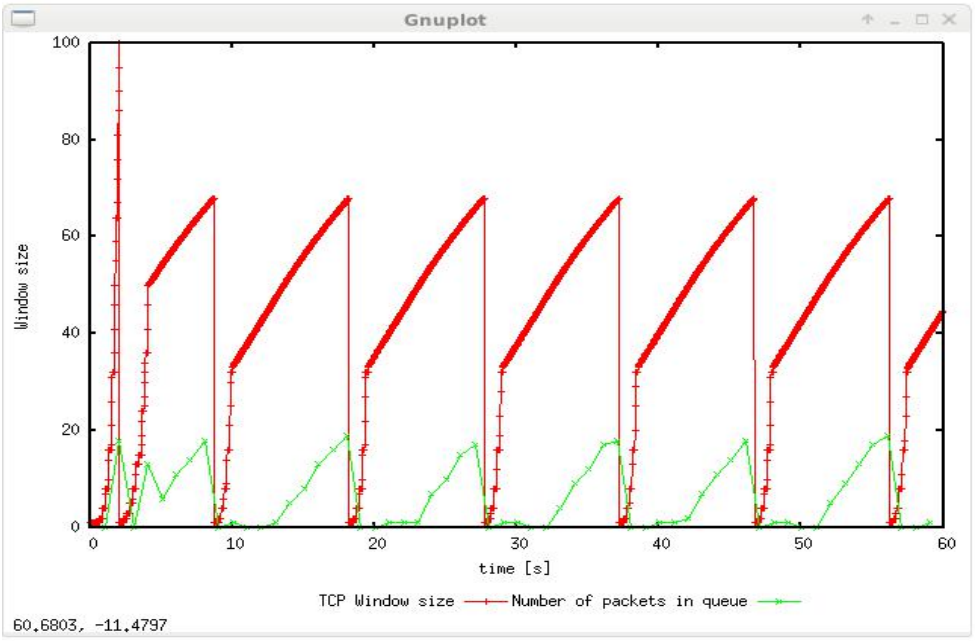


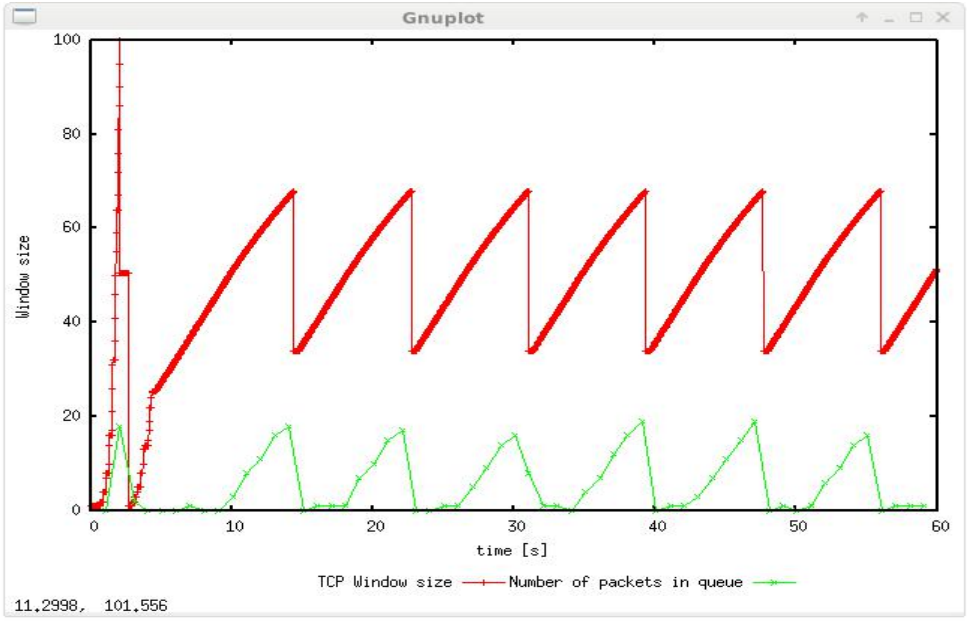
From the graph, the average throughput for max window size 50 is about 230 pkt/s

Change it into bps = 230 * 540 * 8 = 993600 bps = 0.9936 Mbps (header included)

When parameter is smaller than 50,TCP's behavior is similar to max window size 50.
Average throughput for max window size 65 and 50 are very close to link capacity threshold 1Mbps.So,when throughput is nearly equal to link capacity ,TCP tend to have a stable behavior.

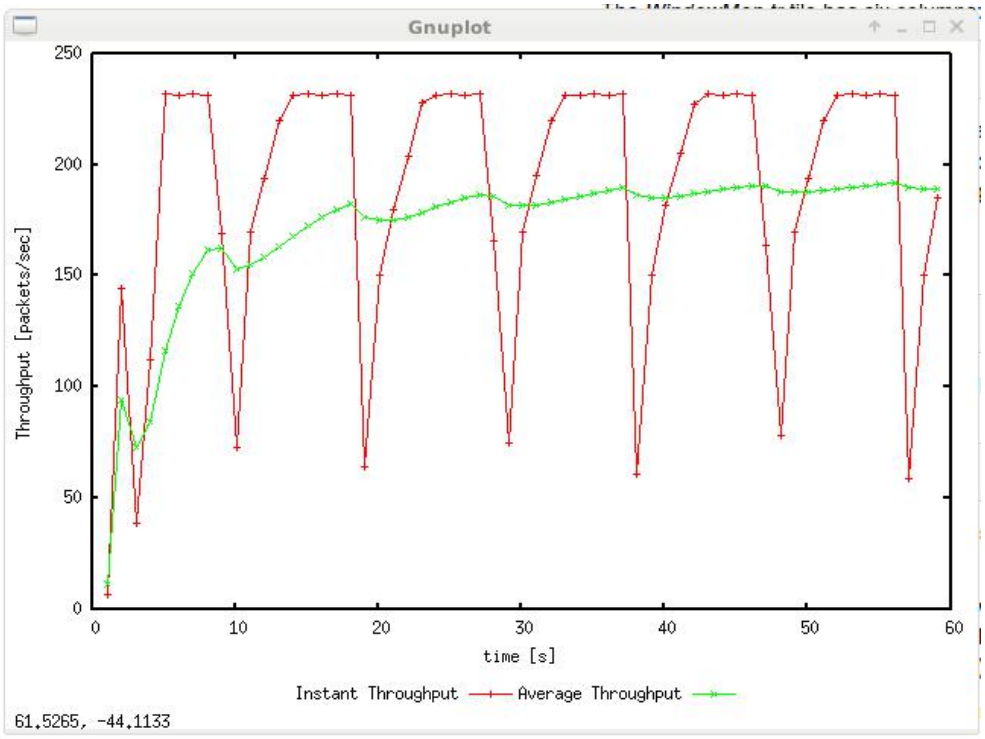
Question 4: Repeat the steps outlined in Question 1 and 2 (NOT Question 3) but for TCP Reno. Compare the graphs for the two implementations and explain the differences. (Hint: compare the number of times the congestion window goes back to zero in each case). How does the average throughput differ in both implementations?

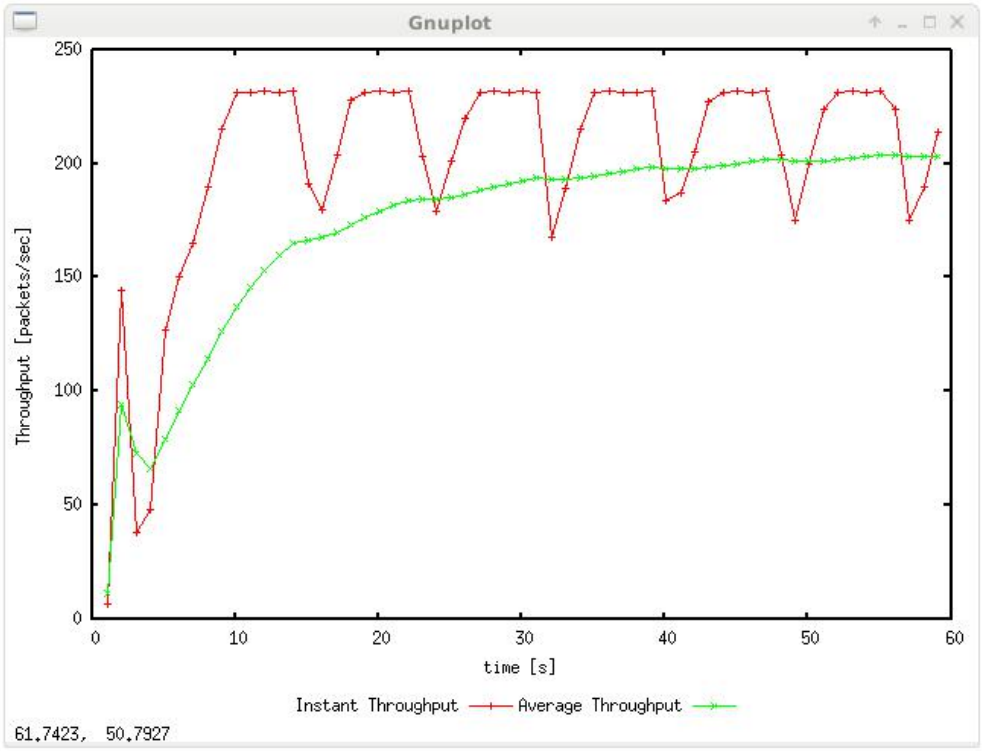




First graph is TCP Tahoe and second one is TCP Reno.For TCP Tahoe,window size cut down to 1 whenever packet loss is detected by triple duplicate ACK and timeout. But for TCP Reno,window

size will reduce to 1 when loss is found by timeout and cut down to half of original size when loss is found by triple duplicate ACK.So,for TCP Reno,most of the packet loss is detected by triple duplicate ACK.





The first graph is throughput of TCP Tahoe and second one is TCP Reno.

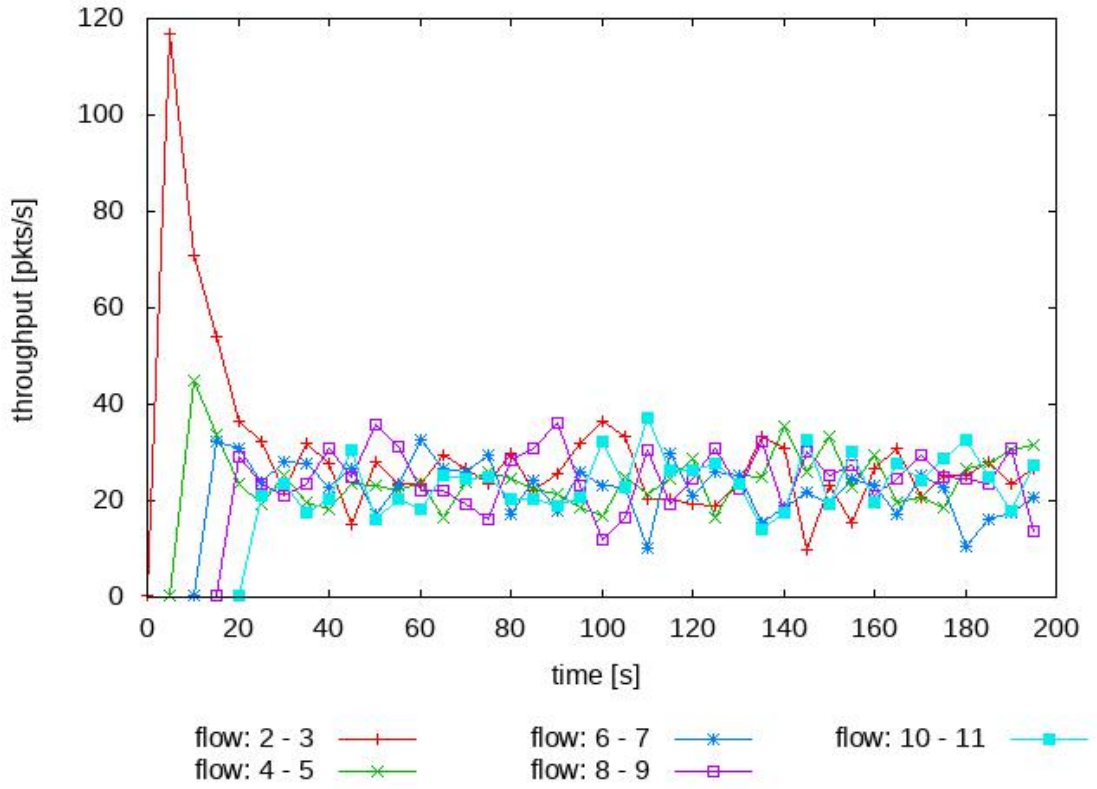
TCP Tahoe average throughput is about 190 pkt/s, TCP Reno is about 200 pkt/s.

Average throughput of TCP Reno is slightly greater than TCP Tahoe.Also,for instant throughput,

TCP Reno has smaller fluctuation than TCP Tahoe.Because when congestion happens, TCP Reno just cut down window size to 1/2 of original size but TCP Tahoe directly reduce it to 1 and then slow start.

Exercise 2: Flow Fairness with TCP

Question 1: Does each flow get an equal share of the capacity of the common link (i.e., is TCP fair) ? Explain which observations lead you to this conclusion.



Yes.

Because at the beginning 0-20s,because of the AIMD algorithm,5 flows have the similar throughput. When the link capacity is limited, the AIMD allows 5 flows to adjust window size.The 5 flows have similar response because they are exposed to the same network condition.

Question 2. What happens to the throughput of the pre-existing TCP flows when a new flow is created? Explain the mechanisms of TCP which contribute to this behaviour. Argue about whether you consider this behaviour to be fair or unfair.

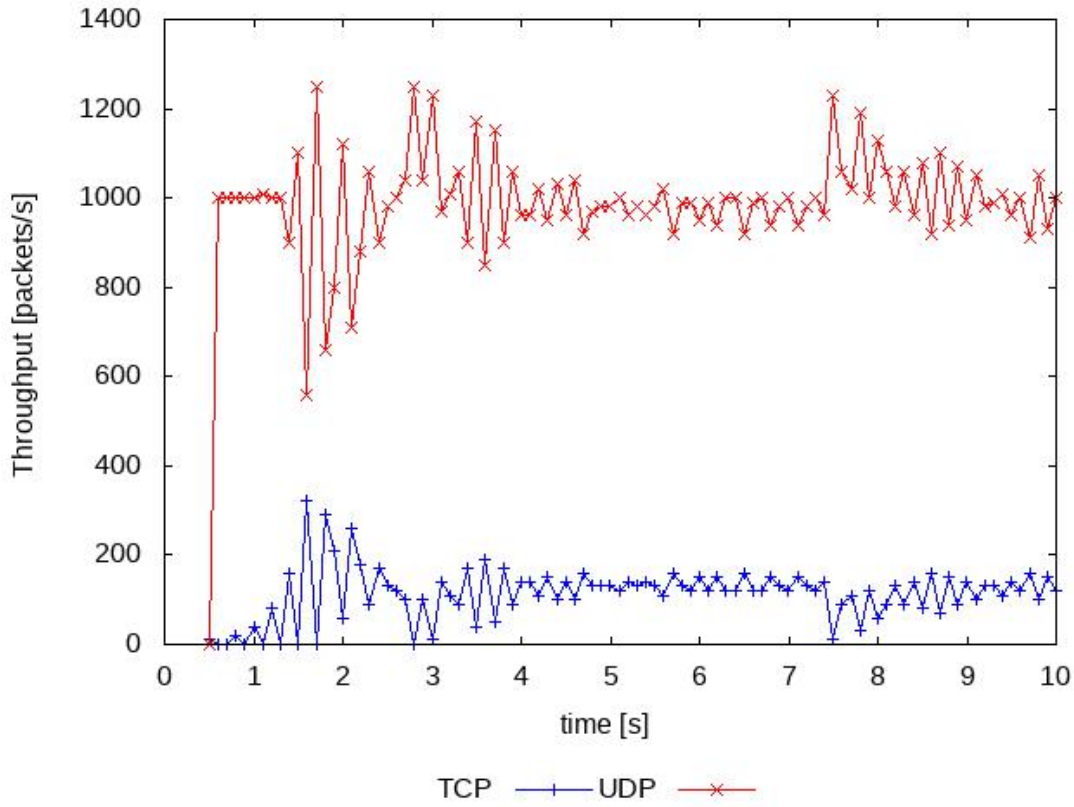
Throughput of pre-existing TCP flows will decrease.

This is because when a new flow come in, it will quickly enlarge congestion window by slow start.Some of the pre-existing flows find their packet loss and reduce their window sizes as a response.If window size reduced, throughput will reduced.

This is fair. Since total link capacity is restricted, more flows added in means smaller throughput for each one.

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? Can you guess which colour represents the UDP flow and the TCP flow respectively ?



I expected that UDP transmission rate will not be affected because UDP doesn't have any consideration about congestion control.But TCP can adjust the window size to fit the current network capacity.
Blue line is TCP, Red line is UDP.

Question 2: Why does one flow achieve higher throughput than the other? Try to explain what mechanisms force the two flows to stabilise to the observed throughput.

UDP achieves higher throughput than TCP.
UDP doesn't have congestion control, which means UDP transmit packets at a fixed rate without handling packet loss. UDP initial rate is far more than TCP. TCP have congestion control and when it detected packet loss,it will narrow window size again and again.So, gradually UDP is dominant in this network while TCP only have a very small throughput.

Question 3: List the advantages and the disadvantages of using UDP instead of TCP for a file transfer, when our connection has to compete with other flows for the same link. What would

happen if everybody started using UDP instead of TCP for that same reason?

Advantages:
UDP ignores congestion in network and will transfer at a fixed rate.This may reduce transmission delay.
Disadvantages:
UDP is not a reliable transfer.For example, it doesn’t handle packet loss

If everyone start to use UDP to transfer file,network links that support significantly high transmission speed are needed,which is a challenge to both technology and finance.