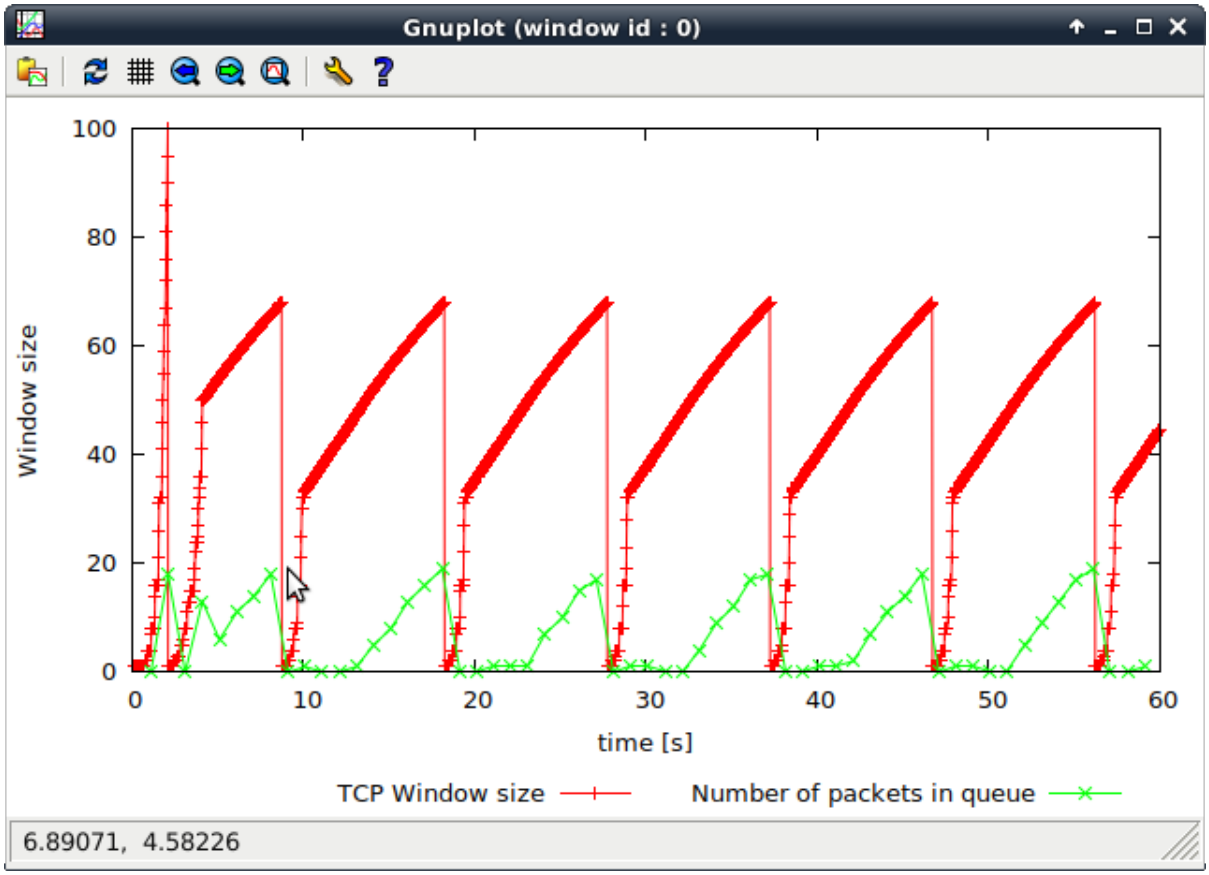
COMP9331
Lab Exercise 5
TCP Congestion Control and Fairness

Z5145114
Xiaodan Wang

Exercise 1: Understanding TCP Congestion Control using ns-2

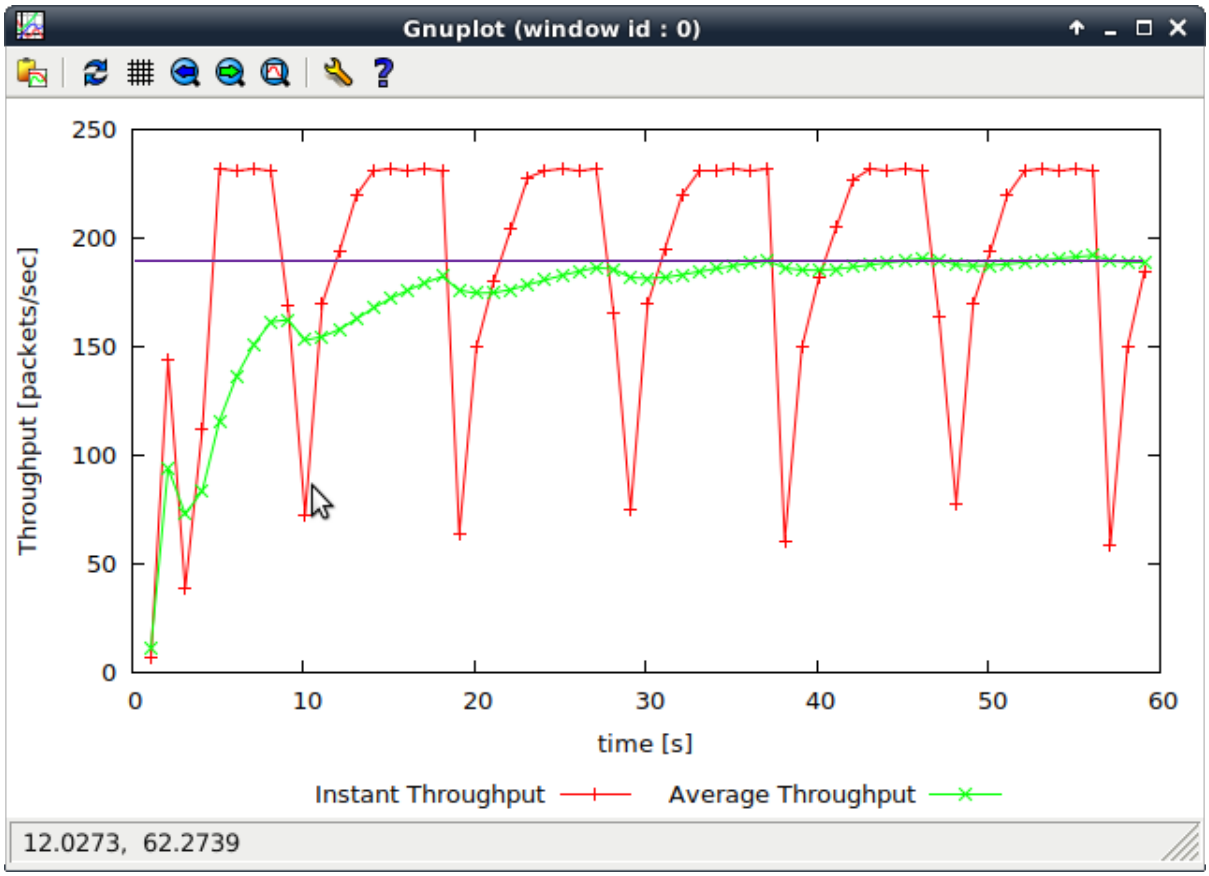
Question 1: Run the script with the max initial window size set to 150 packets and the delay set to 100ms.
The result is below.



What is the maximum size of the congestion window that the TCP flow reaches in this case?
The maximum size of the congestion window that the TCP flow reaches in this case is 100 packets.

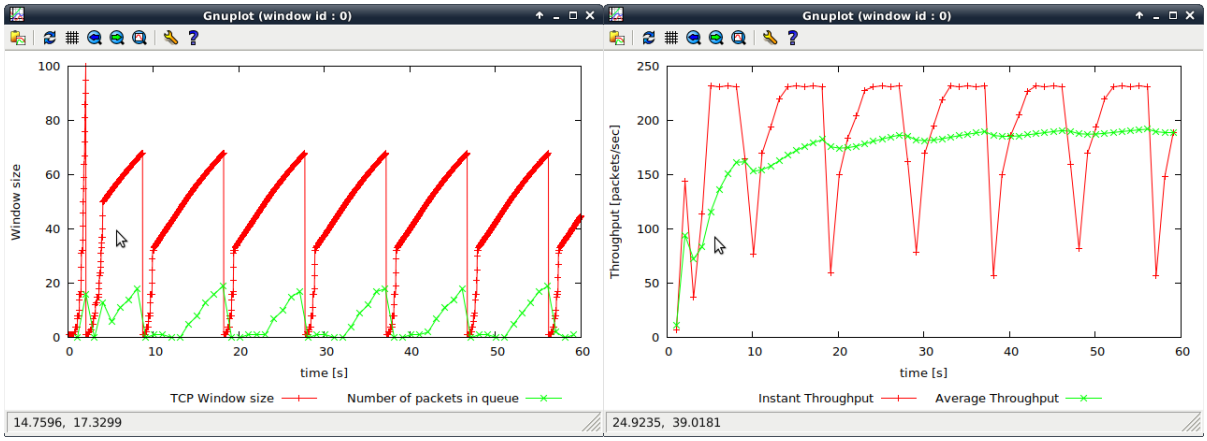
What does the TCP flow do when the congestion window reaches this value? Why? What happens next?
When the size of the congestion window reached 100 packets, the queue was full. Then, some packets were dropped and there was a congestion event at sender. Therefore, the sender reduced the congestion window to 1 and the threshold reduced to 50 packets. The TCP connection enter slow start. After that, the size of congestion window increased rapidly until reached the threshold. And then the connection changed into congestion avoidance phase.

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?

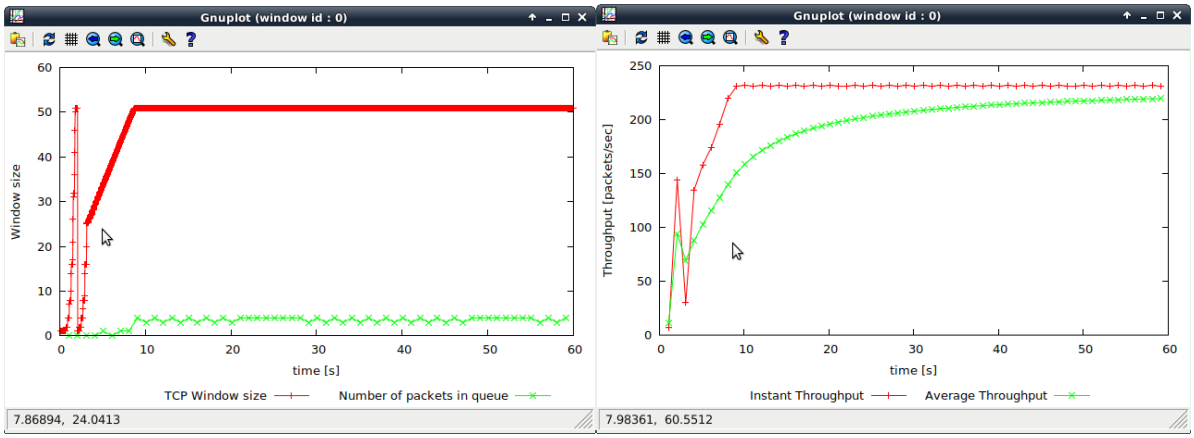


The average throughput is 190 packets per second approximately, the payload of the packet is 500 Bytes and the size of the IP and TCP headers is 20 Bytes, each.
So, the throughput can be calculated as:
(500 + 20 * 2) * 8 * 190 = 802.8Kbps
Each packet has 500 + 20 * 2 Bytes, which is (500 + 20 * 2) * 8 bits.
And in this case, average throughput is 190 packets per second.

Question 3: Rerun the above script, each time with different values for the max congestion window size but the same RTT. How does TCP respond to the variation of this parameter?
With the reduce of max congestion window size, the oscillation reduced.
When the max congestion window size changed to 100, the connection still behavior similarly. However, when reduced to 66, the queue never gets full and the oscillation stop after the first return to slow start.



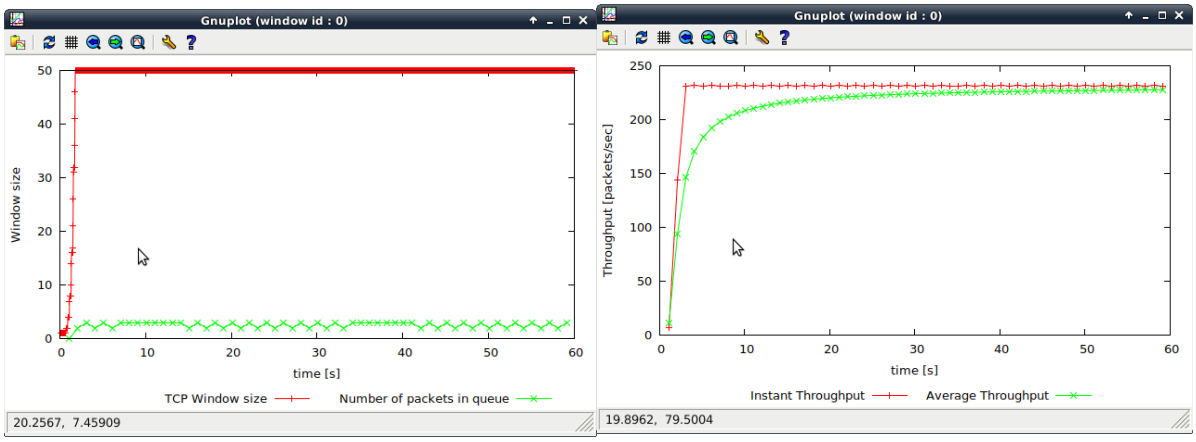
result for size 100



result for size 66

Find the value of the maximum congestion window at which TCP stops oscillating to reach a stable behavior.

When the maximum congestion window size is 50 packets, TCP stops oscillating to reach a stable behavior.

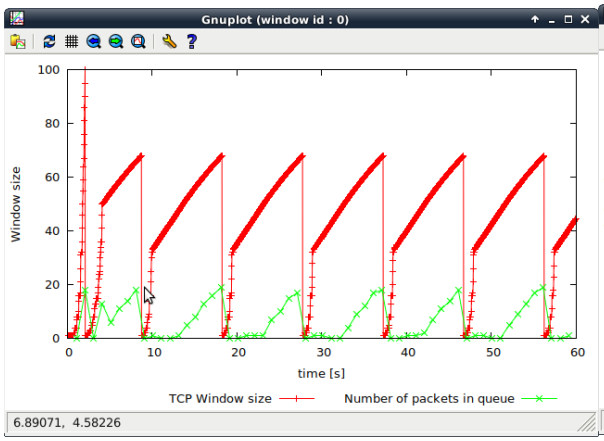


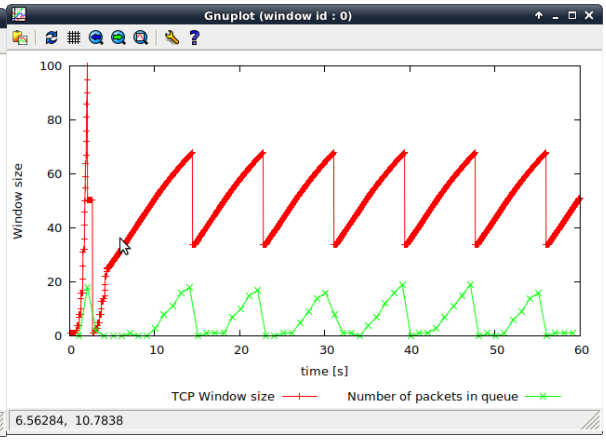
What is the average throughput (in packets and bps) at this point?

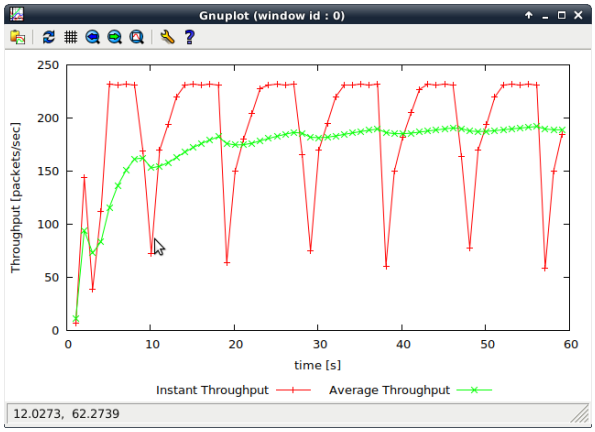
The average throughput at this point is:
Approximately 230 packets per seconds.
Which is 230 * (500 + 20 * 2) * 8 = 993.6 Kbps.

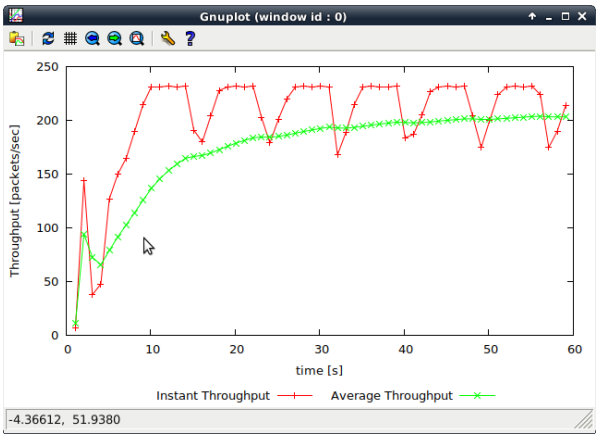
How does the actual average throughput compare to the link capacity (1Mbps)?
The actual average throughput is a bit smaller than the link capacity (993.6Kbps and 1Mbps).

Question 4: Repeat the steps outlined in Question 1 and 2 but for TCP Reno. Compare the graphs for the two implementations and explain the differences. How does the average throughput differ in both implementations?



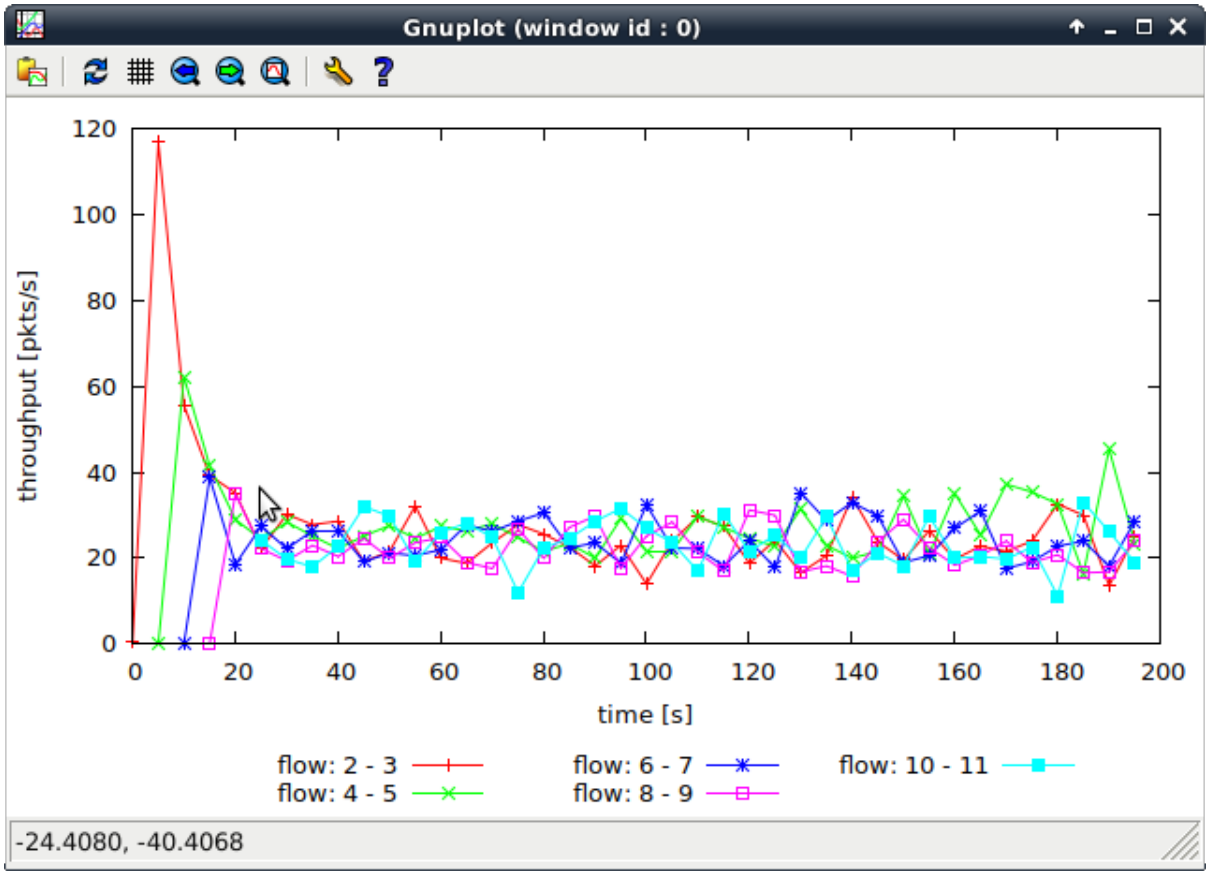






The left the one is TCP Tahoe, while the right one is TCP Reno.
As the graphs showed, in TCP Tahoe the number of times the congestion window goes back to zero is 7, while in TCP Reno is 1. The reason is in Reno, the sender halves its current congestion window and increase it linearly.
The throughput of TCP Reno is around 200 packets per second which is slightly higher than TCP Tahoe. The reason is TCP Reno has less slow start than TCP Tahoe.

Exercise 2: Flow Fairness with TCP

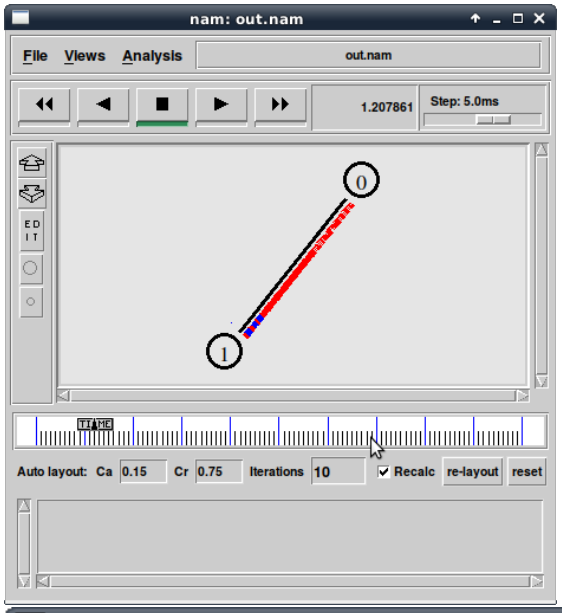


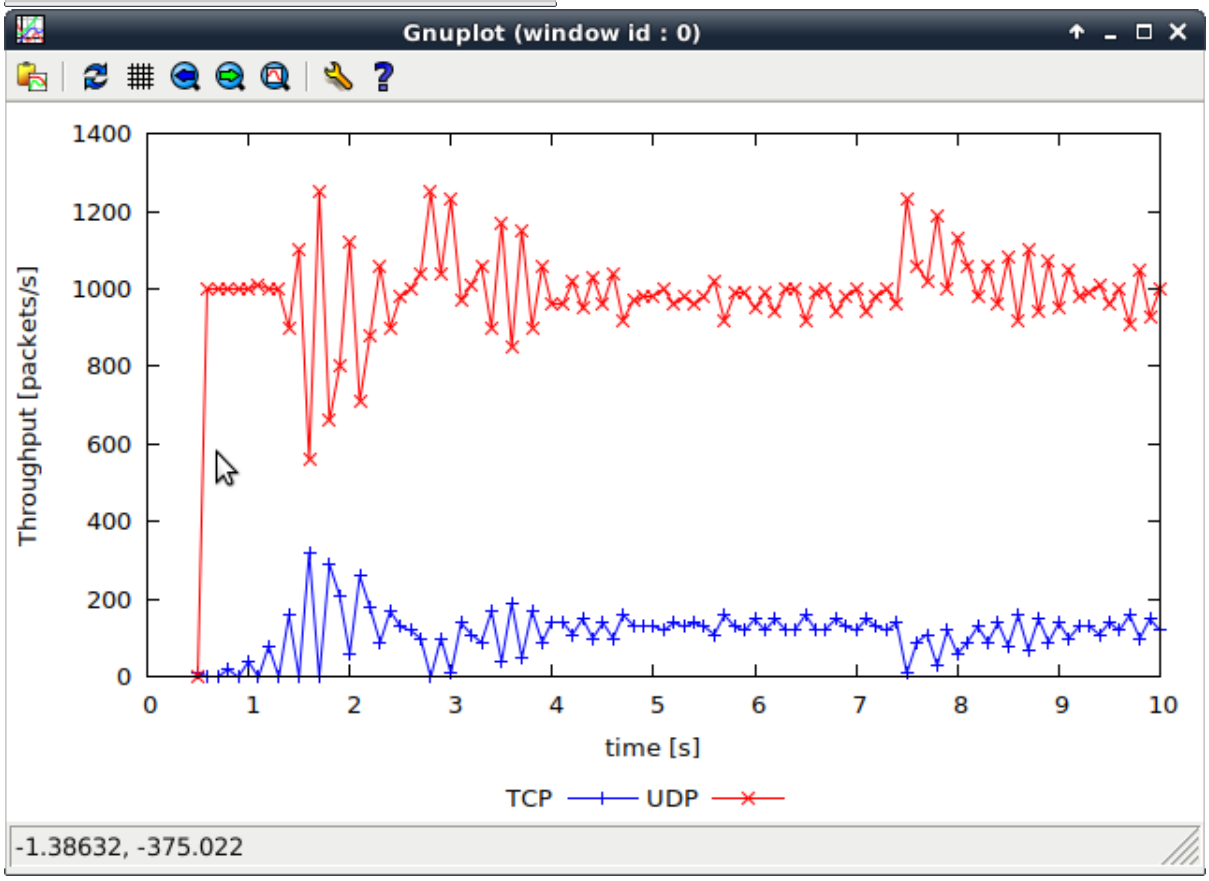
Question 1: Does each flow get an equal share of the capacity of the common link?
I think each flow get an equal share of the capacity of the common link.
The reason is that, after 20 seconds when all five connections have commenced, the throughput for five connections is similar.

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 behavior. Argue about whether you consider this behavior to be fair or unfair.
It is easy to observe from the graph that when a new flow is created, the throughput of the pre-existing TCP flows is reduced. The reason is the once the new flow created, it will create congestion on the link and all existing TCP connections detect losses through duplicate ACKs and timeouts. Then, they will reduce their congestion window.
I think this behavior is fair as every TCP flows share the same throughput.

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?
As UDP do not use any congestion control, the UDP flow will not reduce its transmission rate even the queue is full.
However, the throughput of TCP will be less because it will reduce the transmission rate for congestions.
The results are below.





Question 2: Why does one flow achieve higher throughput than the other? Try to explain what mechanisms force the two flows to stabilize to the observed throughput.
The higher throughput should belong to UDP.
Because UDP is not restricted by the congestions so it has a better transmission rate. But on the other hand, it may occur the data loss.

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?

The advantage of UDP is faster. As there is no congestions in UDP, the sender will concentrate on sending messages and will not be interrupted. It is time saver. However, without ACKs, the data may lost and sender cannot know it. The reliability of UDP is low. The advantage of TCP is reliable but it slower than UDP.
If everybody started using UDP instead of TCP, I think that some parts of the internet would encounter congestion collapse.