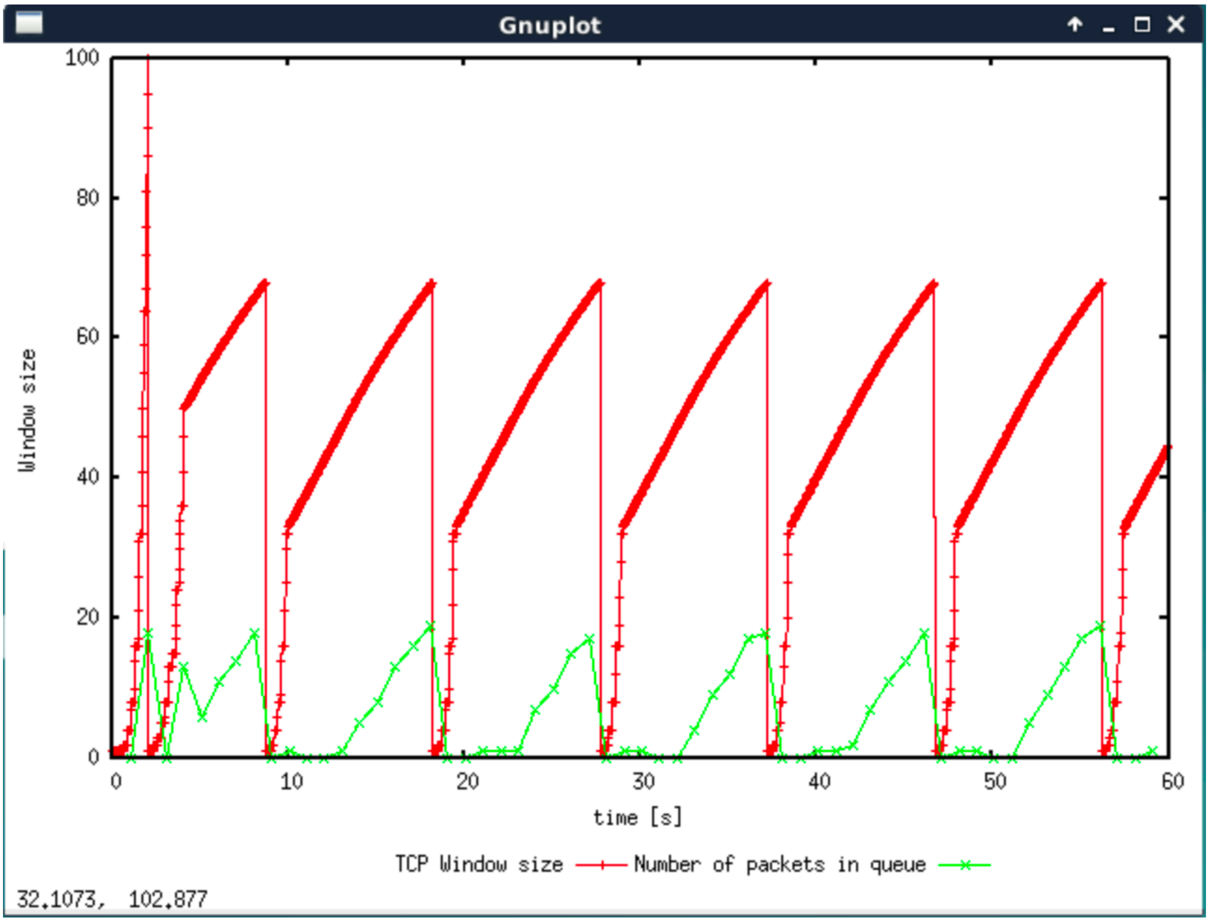
Name: Minh Khai Tran
zID: z5168080

Exercise 1:

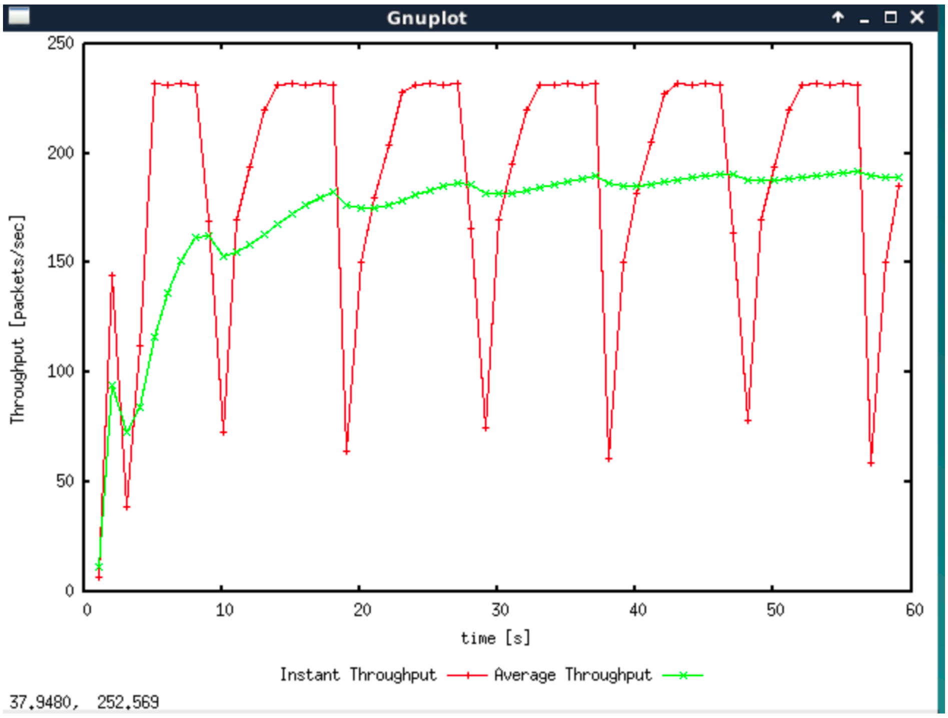
Question 1: What is the maximum size of the congestion window that the TCP flow reaches in this case? What does the TCP flow do when the congestion window reaches this value? Why? What happens next? Include the graph in your submission report.



- As we can see from the graph, the maximum congestion window size is 100 packets.
- When the congestion window reaches to maximum 100 packets, the queue is full (20 packets), so packet dropped happens. Hence TCP flow will start again with congestion window size of 1 with the slow-start mechanism.
- So, when packet drop happens (when the queue is full), the congestion window size goes to 1 and TCP flow will in the slow-start mode again. Also, the threshold will be half of the window size of the previous packet-drop event.

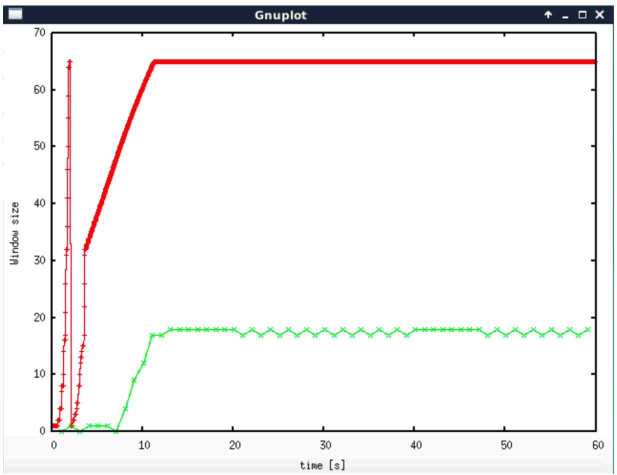
Question 2: From the simulation script we used, we know that the payload of the packet is 500 Bytes. Keep in mind that the size of the IP and TCP headers is 20 Bytes, each. Neglect any other headers. What is the average throughput of TCP in this case? (both in number of packets per second and bps).

- The average throughput of TCP is: $\sim 185 \frac{packets}{s}$
- Size of each packet: $500\ bytes + 20\ bytes\ TCP\ header + 20\ bytes\ IP\ header = 540\ bytes$
- Therefore, the average throughput of TCP is $(185 * 540 * 8) = 799200\ bps$



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)?

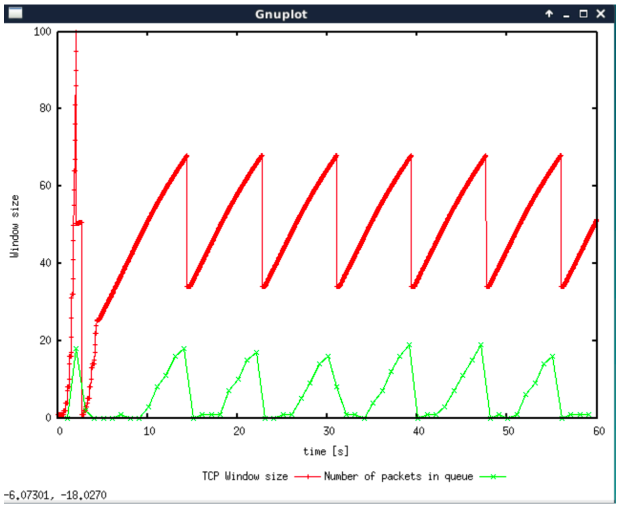
- As we know, the maximum TCP congestion window size when max_cwnd = 150 is 100. So the behaviour when we run the command with the max_cwnd > 100 will be the same.
- From the first graph, as we see the average of congestion window size when the packet-dropped happens is 65. So I choose the maximum congestion window at which TCP stops oscillating is 65, as showed the below graph

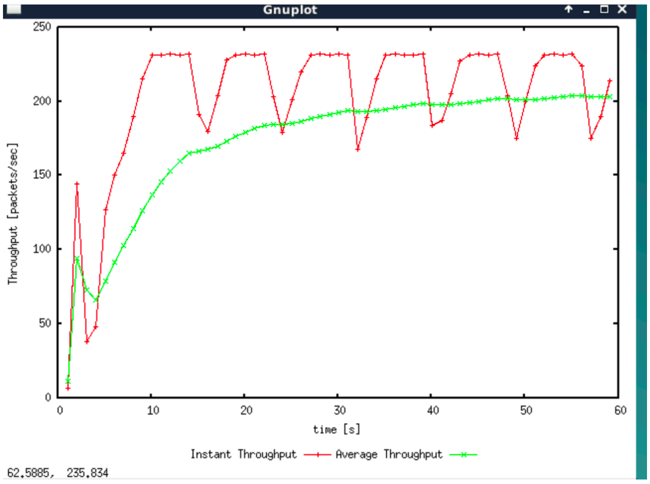


- At this point, the average throughput is $220\ pps = (220 * 540 * 8) = 950400\ bps$
- The average throughput is slightly less than link capacity, that is the reason why we do not have packet-dropped happens.

Question 4:

- First of all, the number of time that the congestion window goes back to zero in this case (TCP Reno) is 1. It was 7 time in the case of TCP Tahoe.
- With TCP Reno, the TCP does not enter to the start-slow phase when loss occurs.
- It only enters to start-slow phase when it detect time-out case, otherwise it enters congestion avoidance phase when triple-acks detected and the congestion window size reduces to half.





- The average throughput is

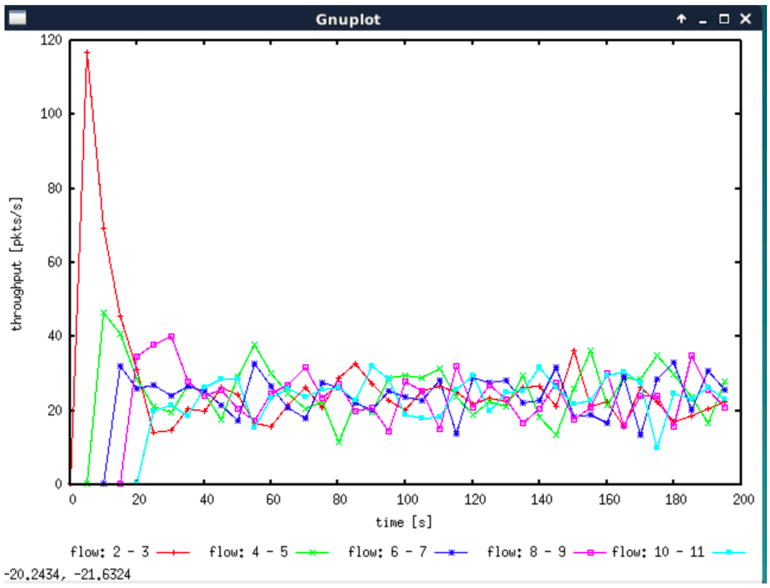
$200\ pps = (200 * 540 * 8) = 864000\ bps$

- The previous throughput of TCP Tahoe was $799200\ bps < 864000\ bps$
- Therefore, TCP reno performs better.

Exercise 2:

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.

- As we see, at the start, the first spike of each flow is different from each other.
    - Flow:2-3 is ~120pps
    - Flow: 4-5 is ~ 50pps
    - Flow: 6-7 is ~30pps
    - Flow: 8-9 is ~35pps
    - Flow: 10-11 is ~25pps
- But eventually, the pikes of each flow are slight the same with each other.
- So, we can conclude that:
    - Each flow does get an equal share of the capacity of the common link.



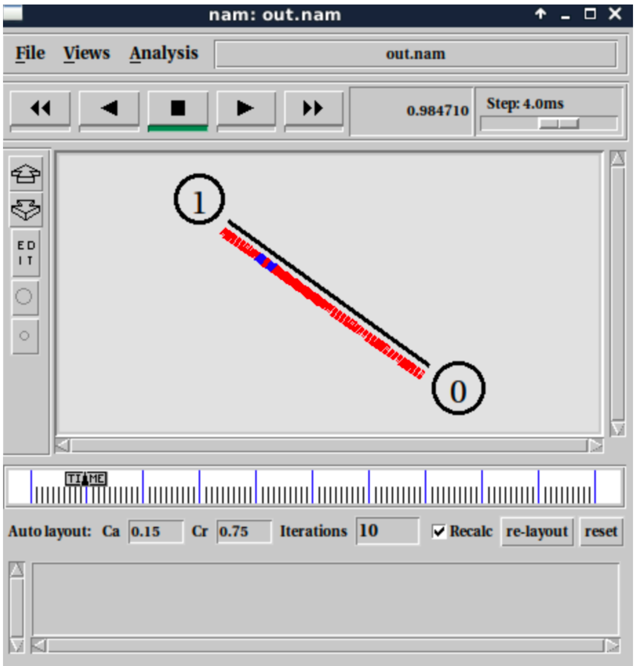
Question 2: What happens to the throughput of the pre-existing TCP flows when a new flow is created? Explain the mechanisms of TCP which contribute to this behaviour. Argue about whether you consider this behaviour to be fair or unfair.

- When the new flow is created, the congestion through put of the pre-existing TCP flows start to drop down so that the new flow is able to have bandwidth so that it can start to transfer data over the link. Eventually, they are all averaged out over time.
- I consider this behaviour to be fair.

Exercise 3:

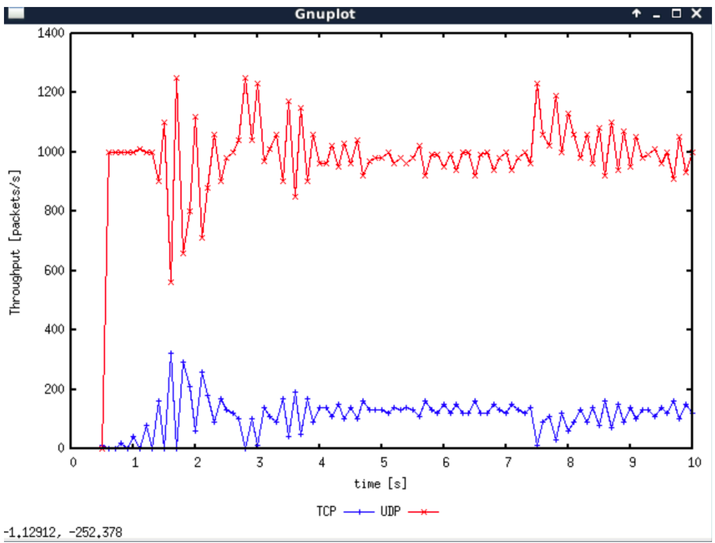
Question 1: How do you expect the TCP flow and the UDP flow to behave if the capacity of the link is 5 Mbps?

- As the picture on the left, I can tell that the number of blue packets increasing exponentially. So blue packets represent TCP flow.
- Also, at each time, the number of red packets larger than number of blue packets
- Therefore, I expect that UPD flow takes more throughput than the TCP flow



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.

- The UDP throughput is higher than TCP throughput.
- Because, as UDP we need to transfer as fast as possible, also in UDP we do not have congestion control, so we try to send as many packets as possible.



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:
  - The client will receive the file faster.
- Disadvantages:
  - The quality of the file is not guaranteed same as the original file.
  - The link could end up collapsing and blocking due to there is no congestion control. Eventually, the whole network will be blocked and collapsed.
- If everybody started using UDP instead of TCP, the whole network will be blocked, packets will be dropped as the queue is full.