Exercise 1: Understanding TCP Congestion Control using ns-2

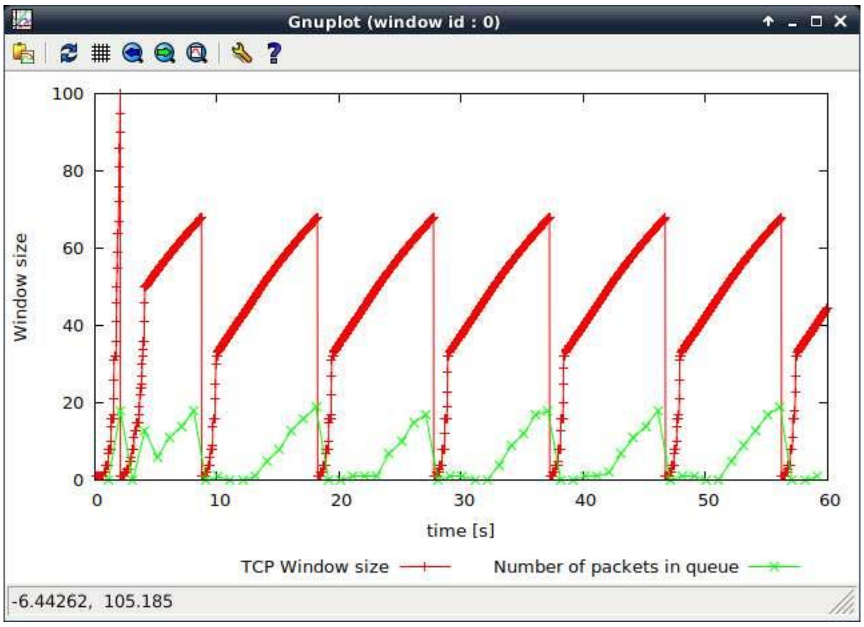
Question 1: Run the script with the max initial window size set to 150 packets and the delay set to 100ms (be sure to type "ms" after 100). In other words, type the following:

```
$ns tpWindow.tcl 150 100ms
```

In order to plot the size of the TCP window and the number of queued packets, we use the provided gnuplot script Window.plot as follows:

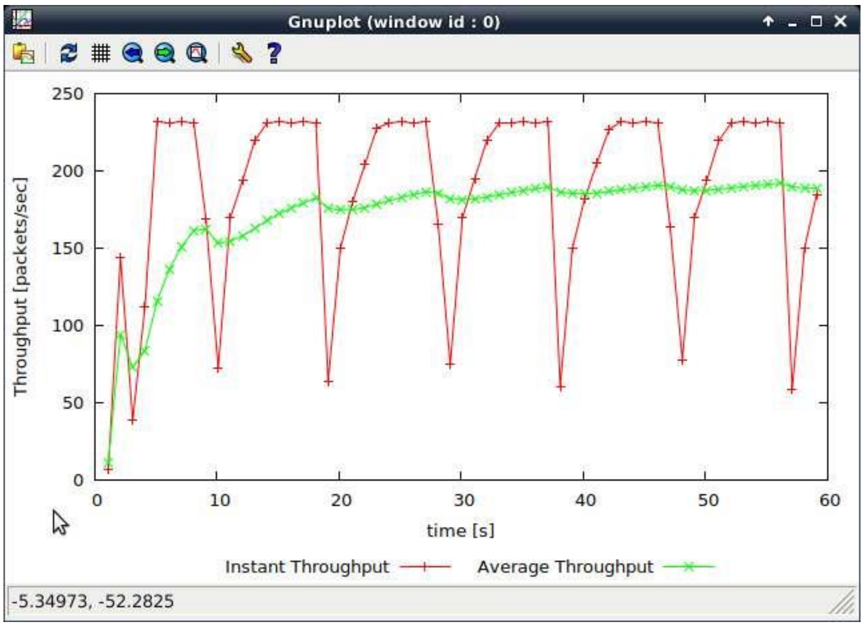
```
$gnuplot Window.plot
```

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.



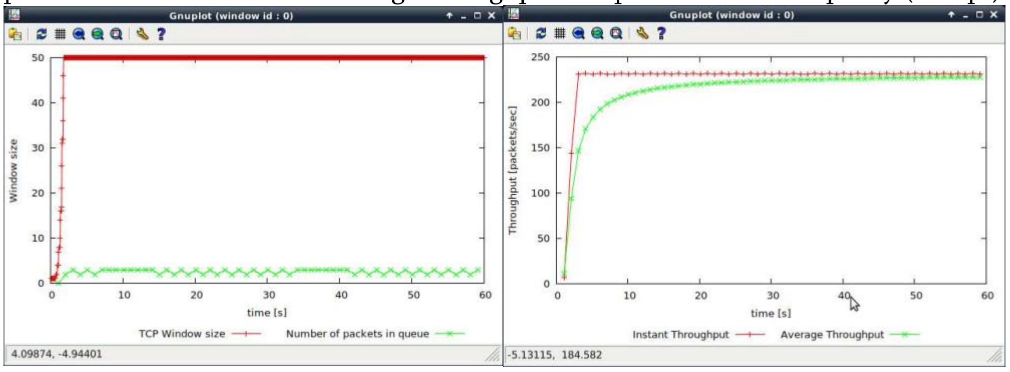
From the graph we can see that although we set the maximum congestion window to 150 packet but the it only grows up to 100 packet. this is because the maximum size of queue is only 20 packets, and any additional packets are droped. When congestion reaches 100, TCP will stop increasing the window size and reset it from 1,Meanwhile, the value of ssthresh has been set to the half of the congestion window. So when the next congestion window reaches the ssthresh, it will apply slow-start algorithm again, and the next cwnd will start from 1 and increase dramaticly until it reaches the ssthresh again, after that, seting half of the ssthresh as previous one again, and apply slow-start.

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)



As the graph shows, the average throughput is around 190 packets / sec and(190*500*8) 760kbps(header payload removed).

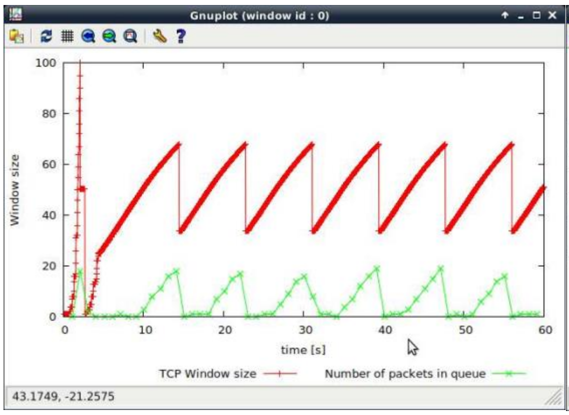
Question 3: Rerun the above script, each time with different values for the max congestion window size but the same RTT (i.e. 100ms). How does TCP respond to the variation of this parameter? Find the value of the maximum congestion window at which TCP stops oscillating (i.e., does not move up and down 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)?

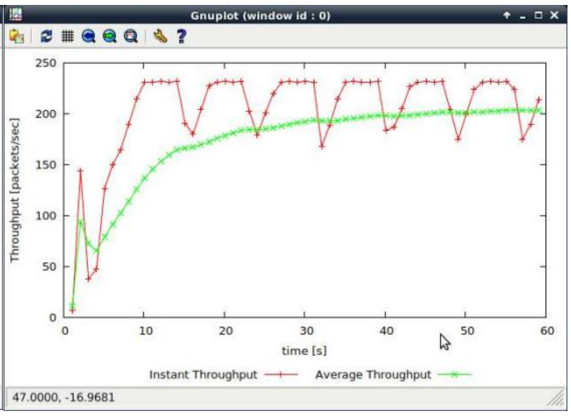


I set the value of max cwnd as 50, and get the two graphs showing above, at the beginning, cwnd started from 1 and increased dramatically up to 50 which the max value that I set, after that, it stops oscillating and keeps in 50.
The average throughput of this point is around 240 packet/sec
The actual throughput in this case is (240*500*8)/1000000 = 0.96Mbps
Link capacity is the maximum transmission rate and actual throughput can not exceed the value of link capacity.

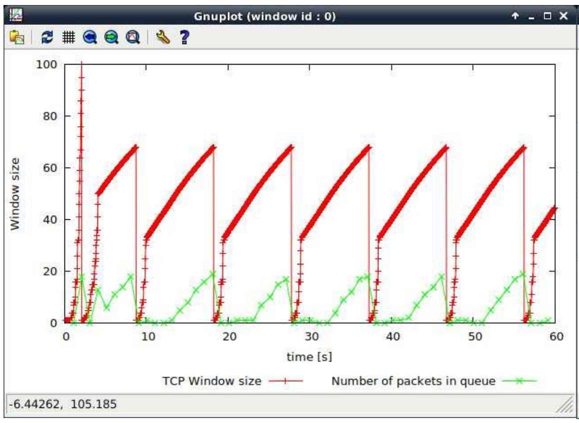
Question 4: Repeat the steps outlined in Question 1 and 2 (NOT Question 3) but for TCP Reno. Compare the graphs for the two implementations and explain the differences. (Hint: compare the number of times the congestion window goes back to zero in each case). How does the average throughput differ in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

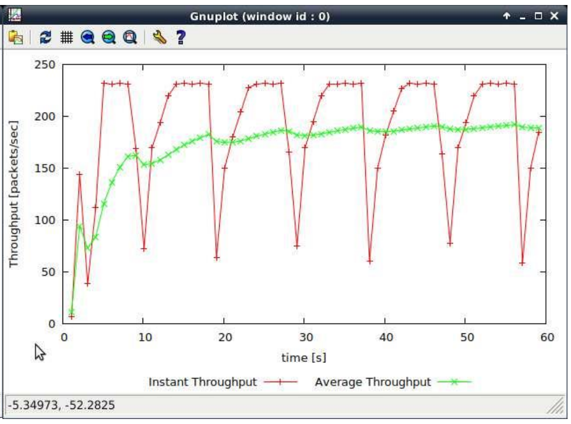
TCP Reno





TCP Tahoe

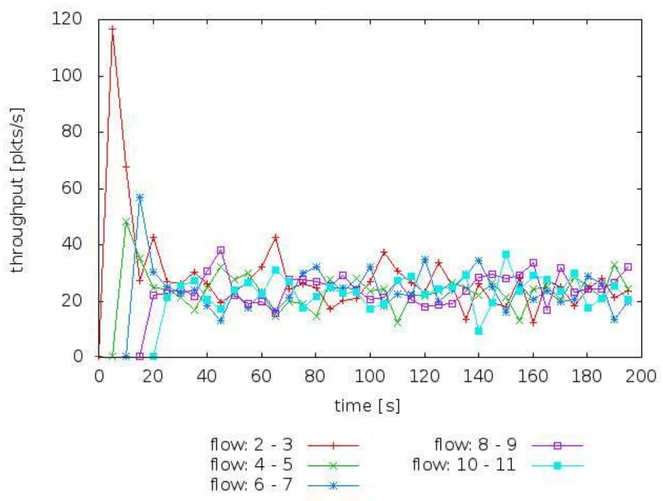




The difference between TCP Reno and TCP Tahoe is that Reno apply the Fast Recovery algorithm but Reno do not, as we can see from graph above, after sender received three consecutive acknowledgments, it will execute AIMD algorithm, and set ssthresh to half of previous one, and then it will not use slow-start algorithm instead setting cwnd to the ssthresh that have already changed and execute avoidance algorithm which allow cwnd increasing linearly.

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.



As the graph shows , even there are same fluctuations, the throughput for all connections is similar, this is because AIMD algorithm will adapt the window size for achieving TCP fairness. So when many flows share one link, since they experience the same states of the network, they will react in same manner.

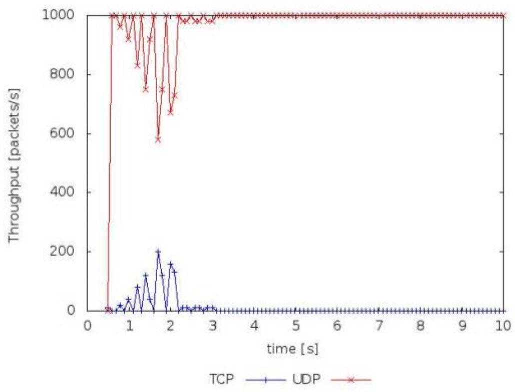
Question 2. What happens to the throughput of the pre-existing TCP flows when a new flow is created? Explain the mechanisms of TCP which contribute to this behaviour. Argue about whether you consider this behaviour to be fair or unfair.

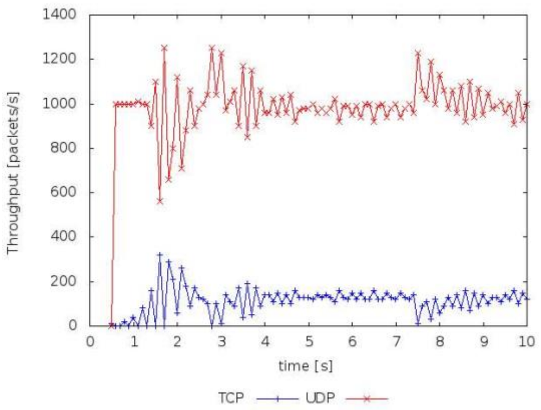
When the new flow is created, it will increase its congestion window size quickly and TCP strategy will try to adapt the congestion window in order to balance the network.

It is fair because if there is no throughput shared after new flow come into the link, it can not be sent until the pre-existing TCP transmission finished.

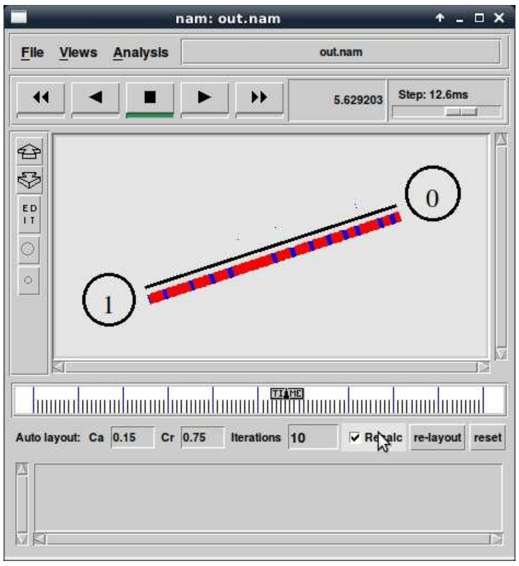
Exercise 3: TCP competing with UDP

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 the graph shows, UDP will be not restrain by any congestion control, and it will ignore every response of congestion, unlike TCP every congestion response will invoke AIMD to reduce its transmission rate, so according above, UDP take priority.





4Mbps 5Mbps



Blue is TCP
RED 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.

TCP is the reliable and stable protocol, which means that there are many constrains during the transmission , but UDP only need to send datagrams to the link without any constrains, and this is not fair to the TCP, because since TCP find any congestion response, it will reduce its transmission rate, so the throughput of UDP could be larger than TCP.

TCP employ the congestion and flow control.

UDP is rely on the capacity of the Link.

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?

TCP

Advantage:
stable and reliable
Disadvantage:
many constraints for keeping reliable transmission so that the speed of transmission could be very slow

UDP
Advantage:
1.the speed is very fast and Much faster than TCP.
Disadvantage:
1. There are no guarantees with UDP.
2. UDP has no flow control, Congestion Control, implementation is the duty of user programs.

If everyone use UDP instead of TCP, the high rate of packet loss will happened, if people want to download the file , due to the high packet loss, they are not likely to receive the entire documents that they are request.