

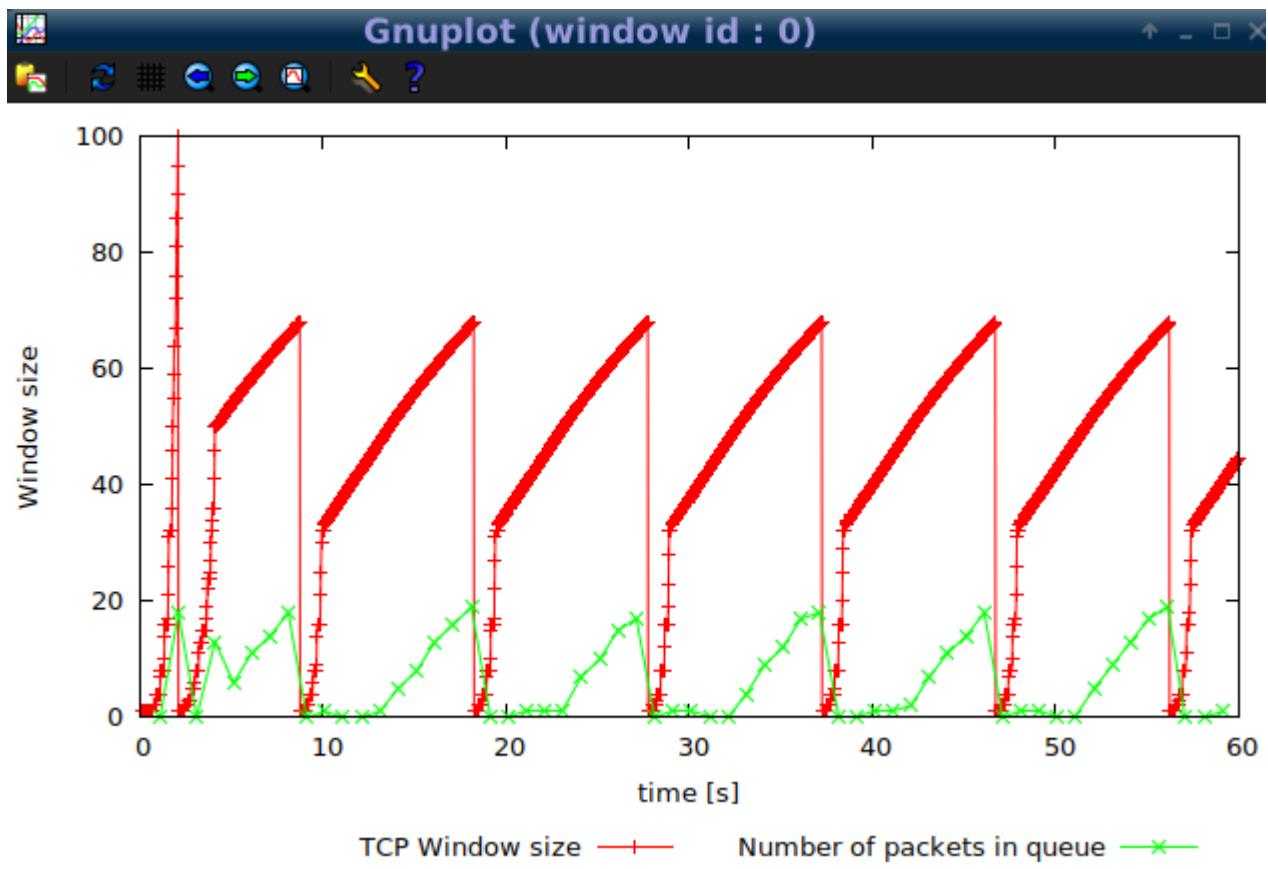
# Exercise 1: Understanding TCP Congestion Control using ns-2

**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.

The maximum size of the congestion window is 100 packets in slow start.

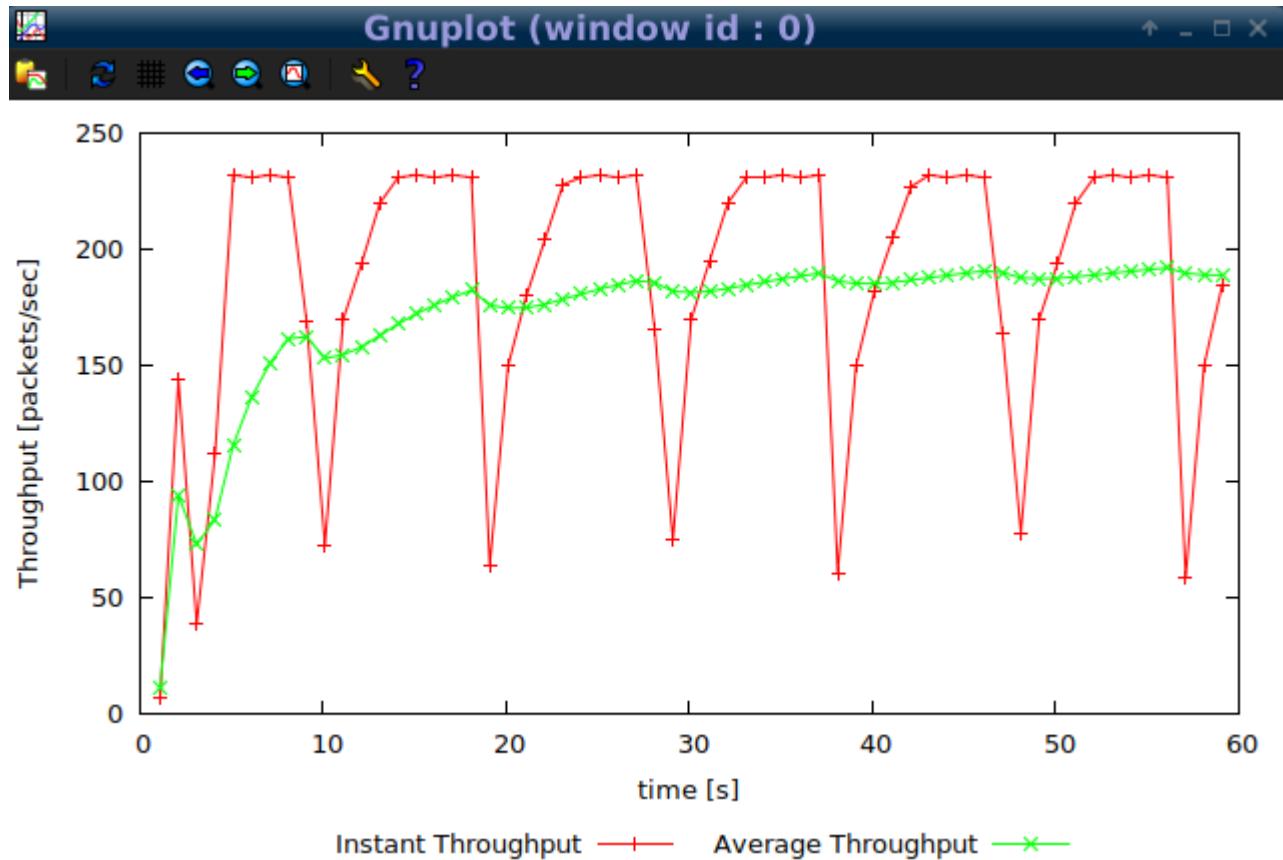
The queue is full because the size of queue is 20 packets. When reaching that value, some packets are dropped and the ssthresh is set to the half of the previous value, which is 50 packets.

It will come back to the slow start. And then reach to the congestion avoidance phase again. It will oscillate between the slow-start phase and the congestion avoidance phase.



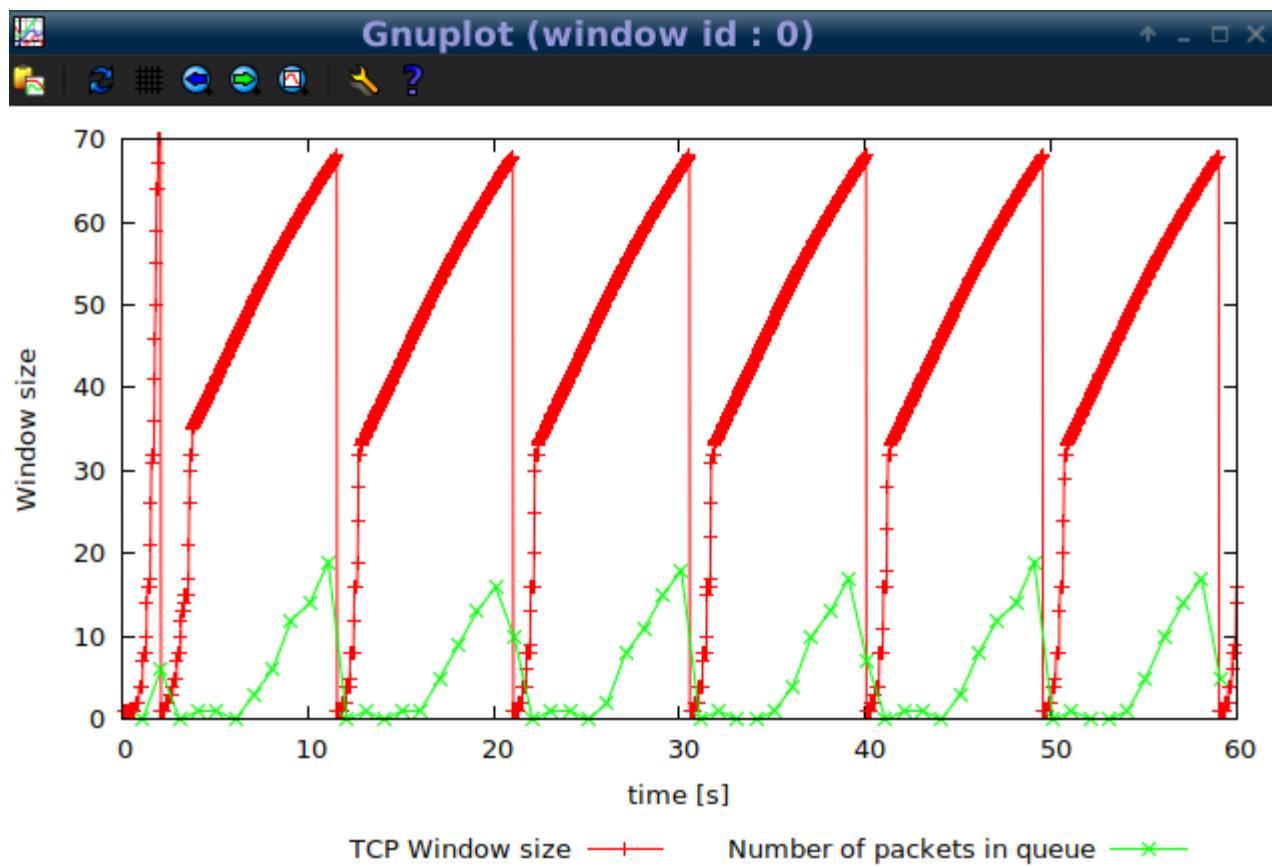
**Question 2:** What is the average throughput of TCP in this case? (both in number of packets per second and bps)

The average throughput in packets is about 185 packet per second. IP and TCP headers is  $20 + 20 = 40$  bytes. The payload is 500 bytes. Therefore, the average throughput in bps is  $(40 + 500) * 8 * 185 = 799,200$  bps.

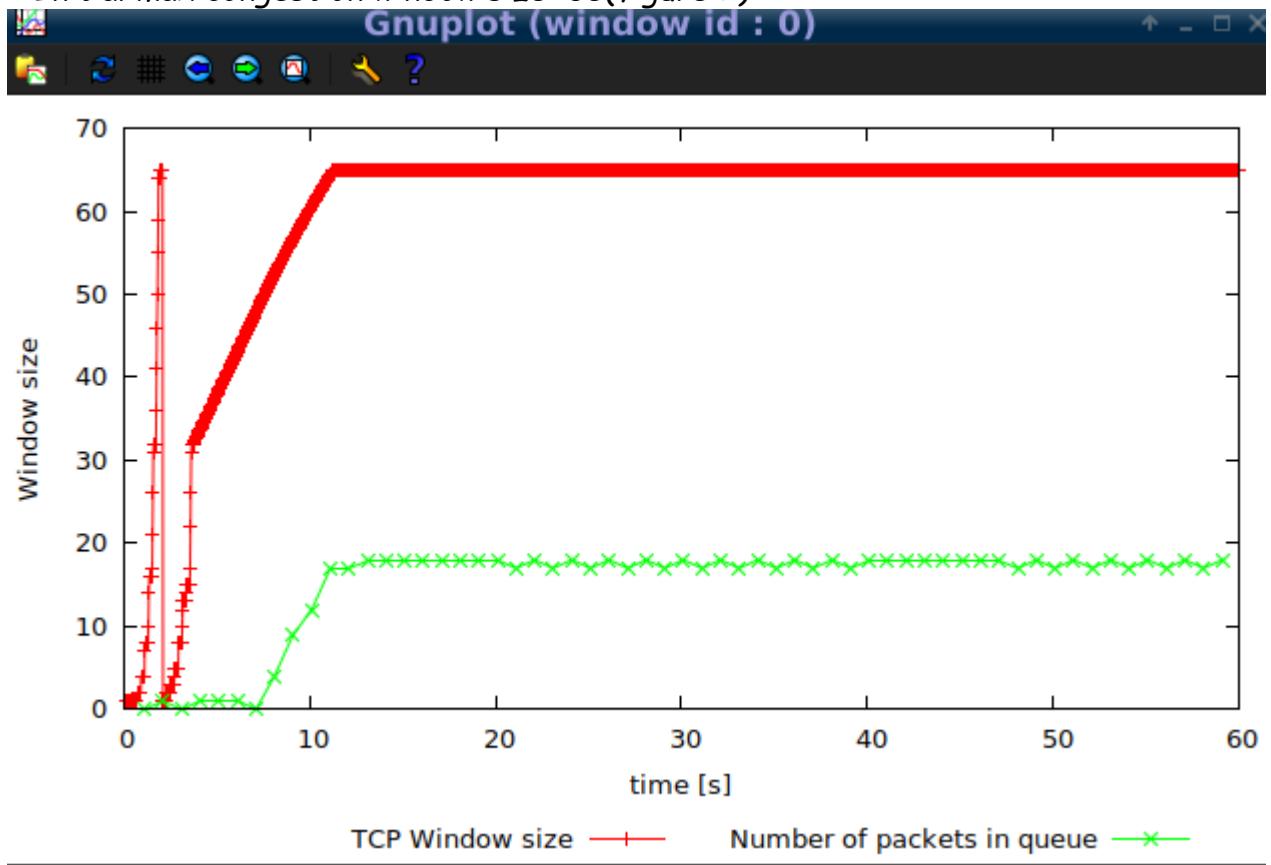


**Question 3: How does TCP respond to the variation of this parameter? Find the value of the maximum congestion window at which TCP stops oscillating (i.e., does not move up and down again) to reach a stable behaviour. What is the average throughput (in packets and bps) at this point? How does the actual average throughput compare to the link capacity (1Mbps)?**

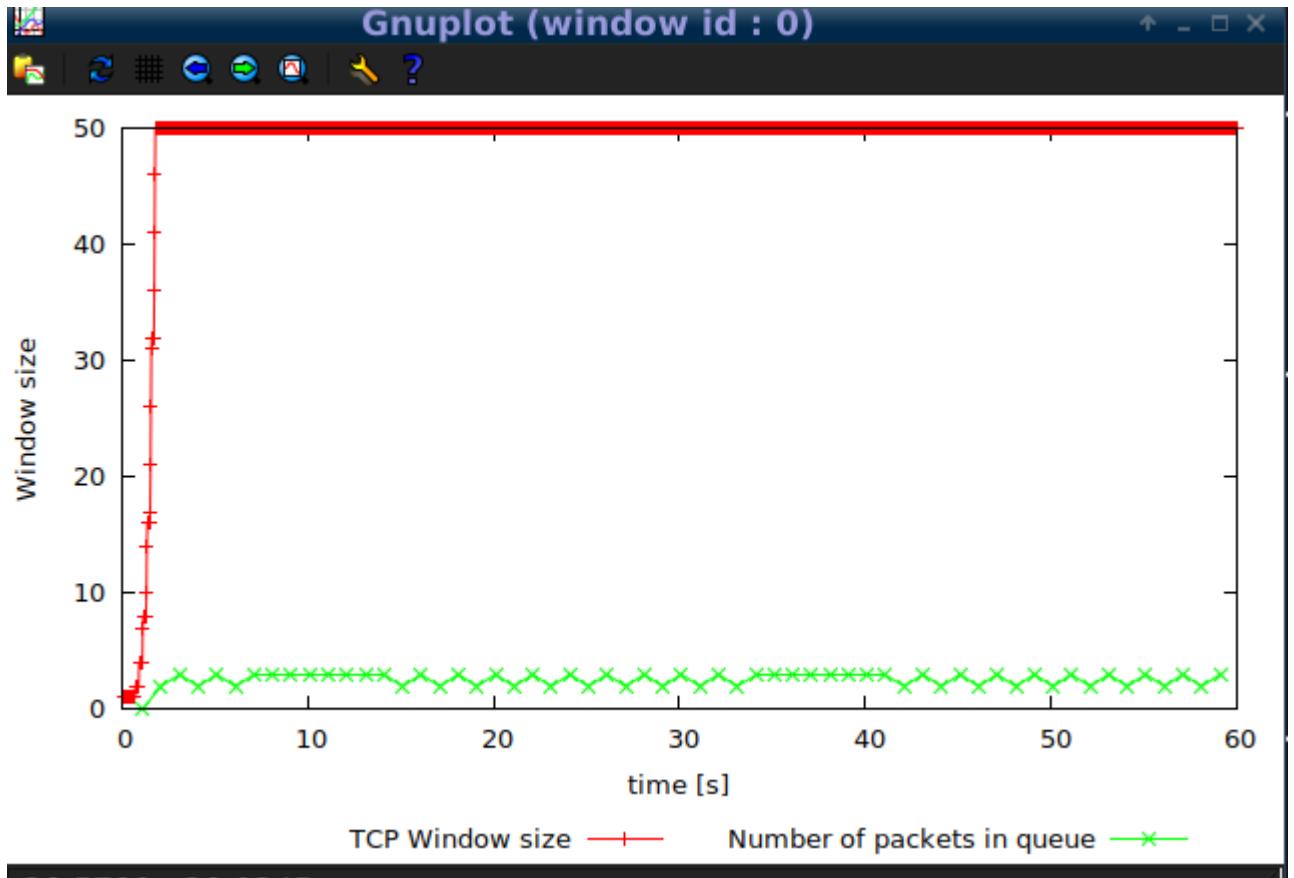
Initial max congestion window size: 70 (figure 1)



Initial max congestion window size: 65(figure 2)



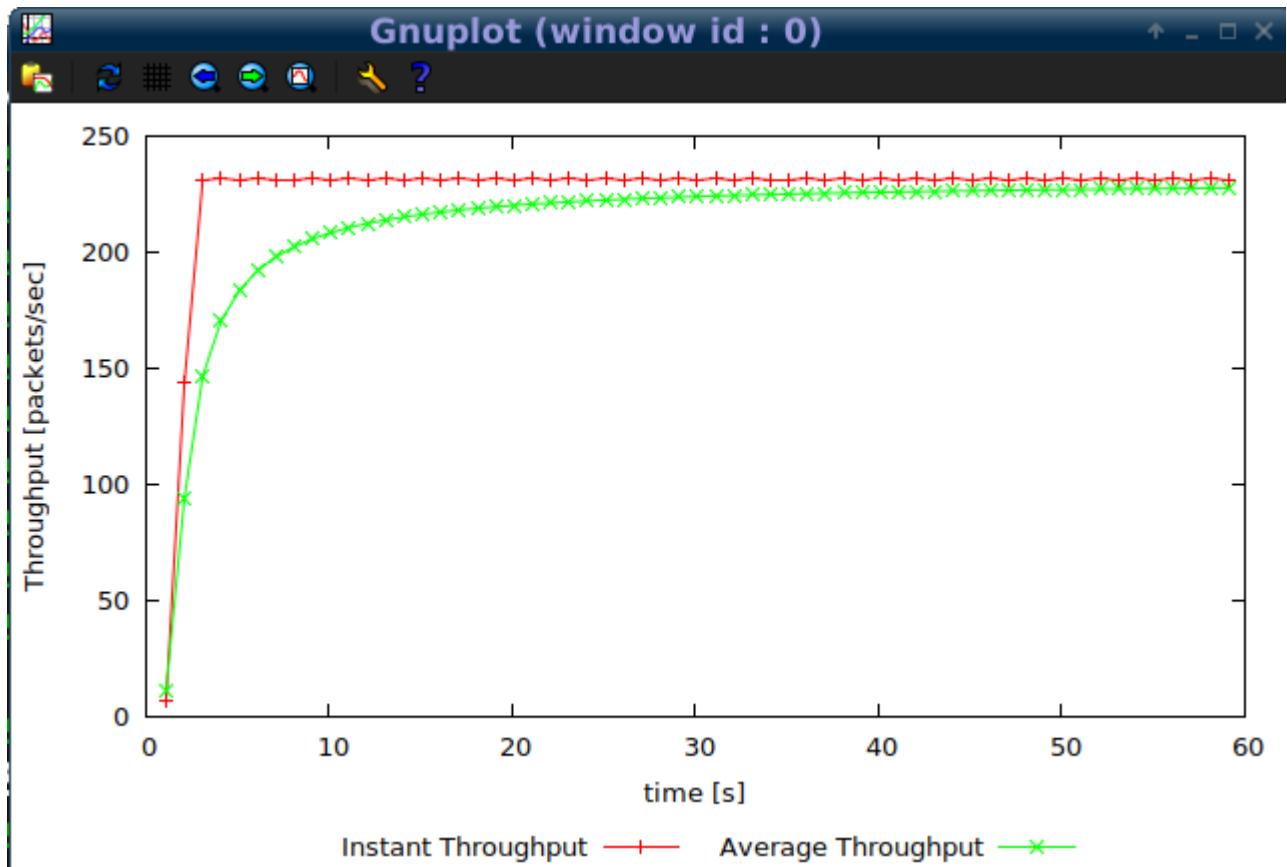
Initial max congestion window size: 50 (figure 3)



In figure 1 (with 66 packets of max window size), TCP oscillates between the slow-start phase and the congestion avoidance phase. In figure 2 (with 66 packets of max window size), the oscillations stop after the second slow-start.

In figure 3, TCP reaches to 50 and stops the oscillating (with 50 packets of max window size). From figure 4, the average packet throughput is about 225 packets per second. Thus,  $225 * 8 * 540 = 972,000$  bps, which is near the link capacity(1Mbps).

Throughput(*figure 4*)

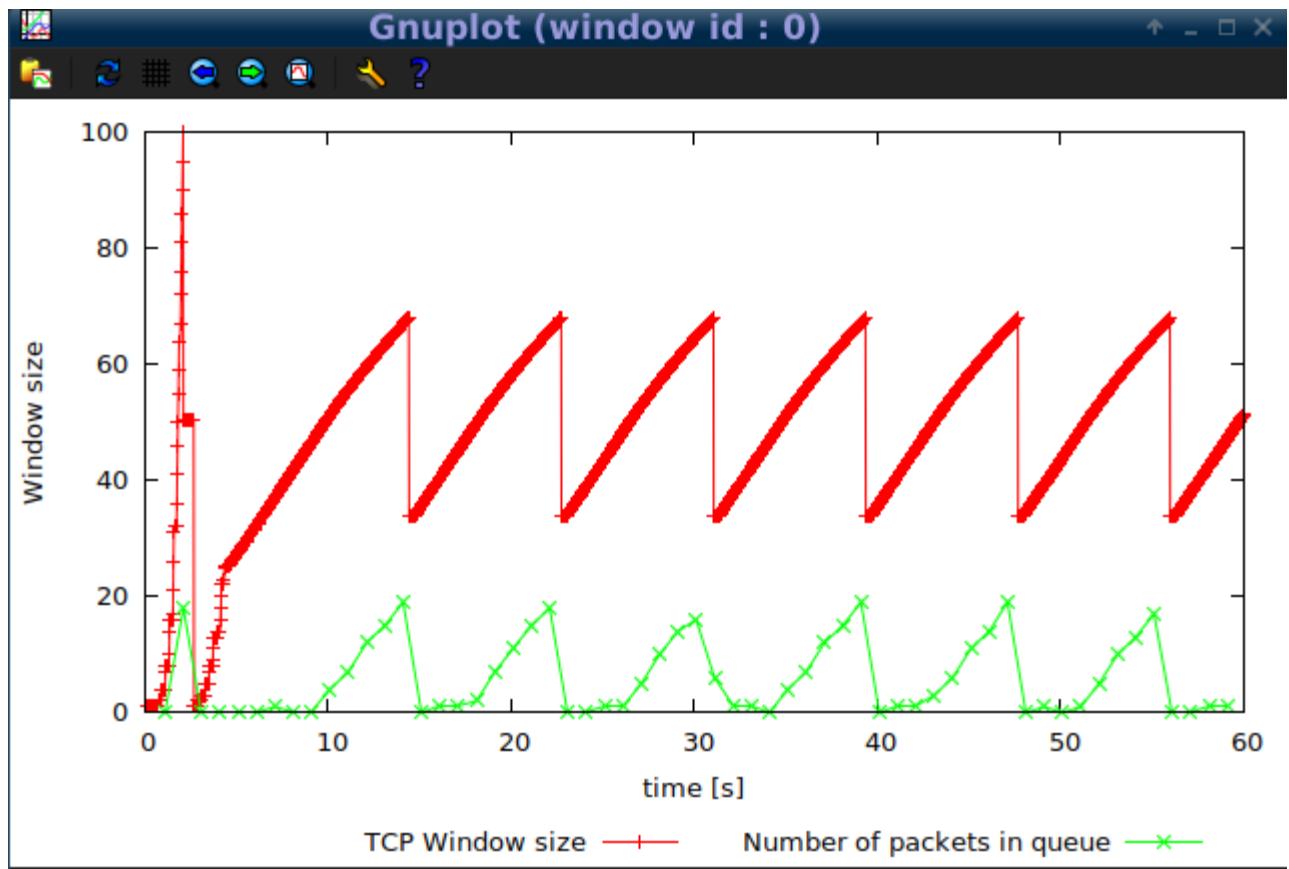


#### Question 4: How does the average throughput differ in both implementations?

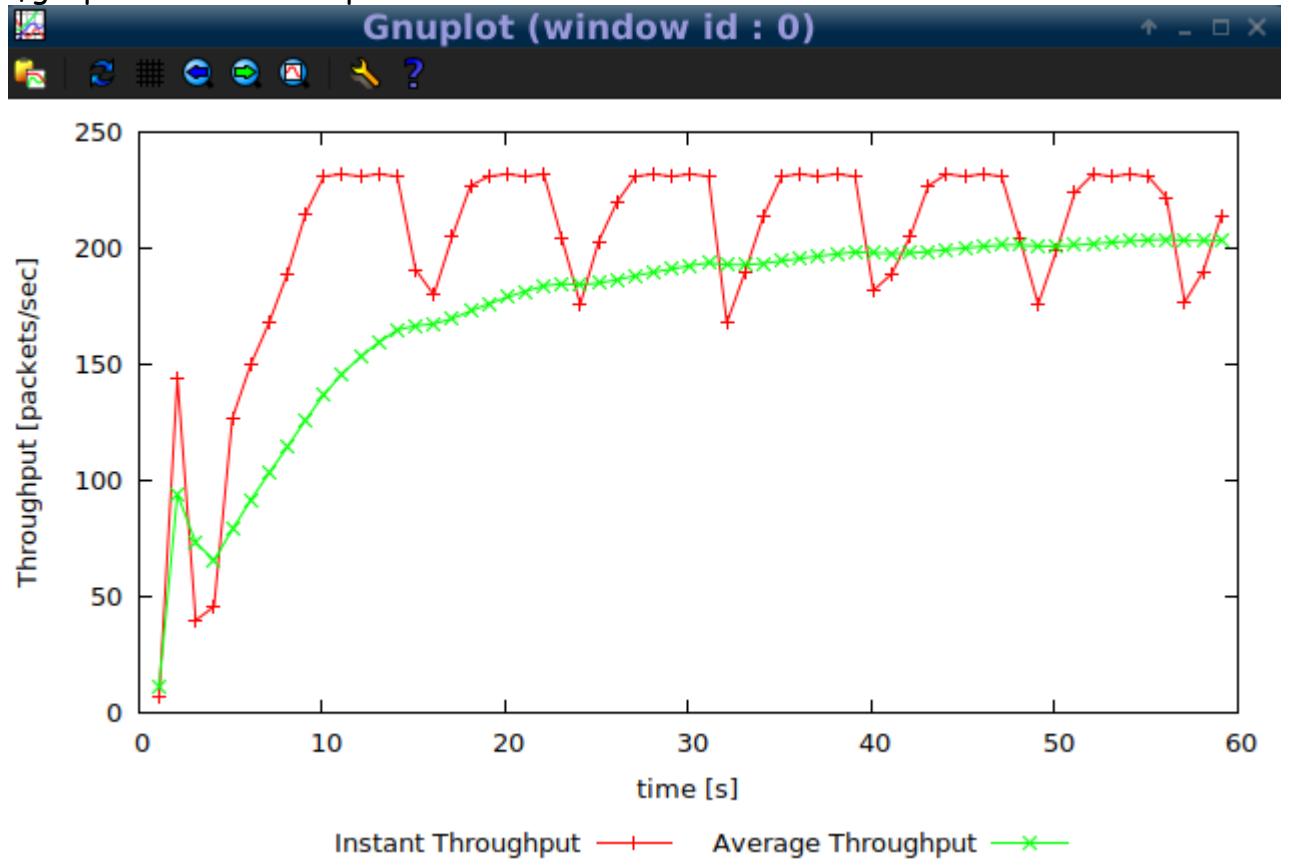
The maximum size of the congestion window: 150. When a loss happens, the TCP Reno does not drop the window size to 1 MSS. It drops to the half and then enters to congestion avoidance. This is called fast recovery and it could prevent 'pipe' from emptying after fast retransmit. The throughput is about 200 packet per second and then there is  $200 * 8 * 540 = 864,000$  bps.

Replace set tcp0 [new Agent/TCP]  
to set tcp0 [new Agent/TCP/Reno]

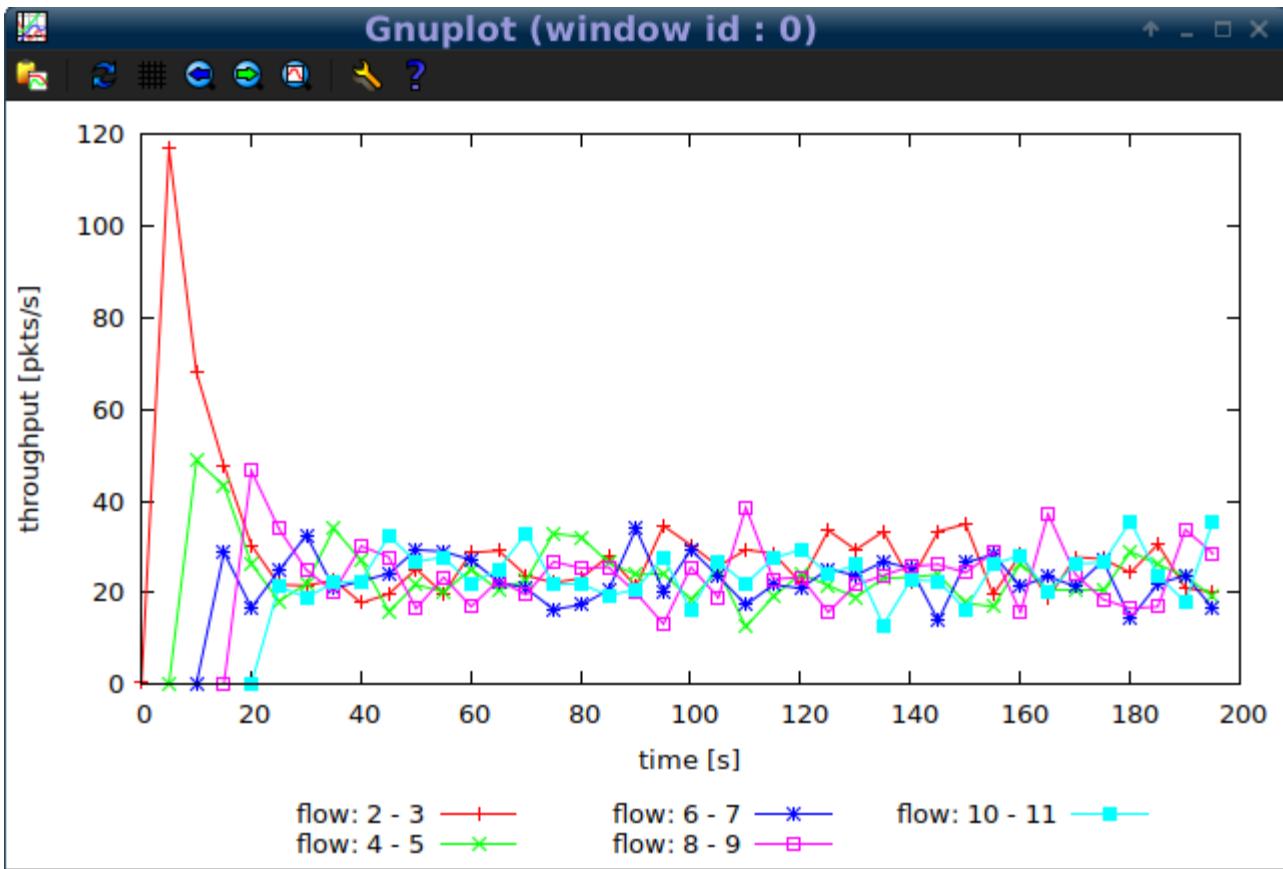
```
$ns tpWindow.tcl 150 100ms
$gnuplot Window.plot
```



\$gnuplot WindowTPut.plot



## 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. The throughput of node 2 is about 120 at the beginning and it decreases whenever a new connection of other nodes starts. The first few nodes will do the same behaviour. The throughput of nodes oscillate in a similar range after all connections of nodes start.

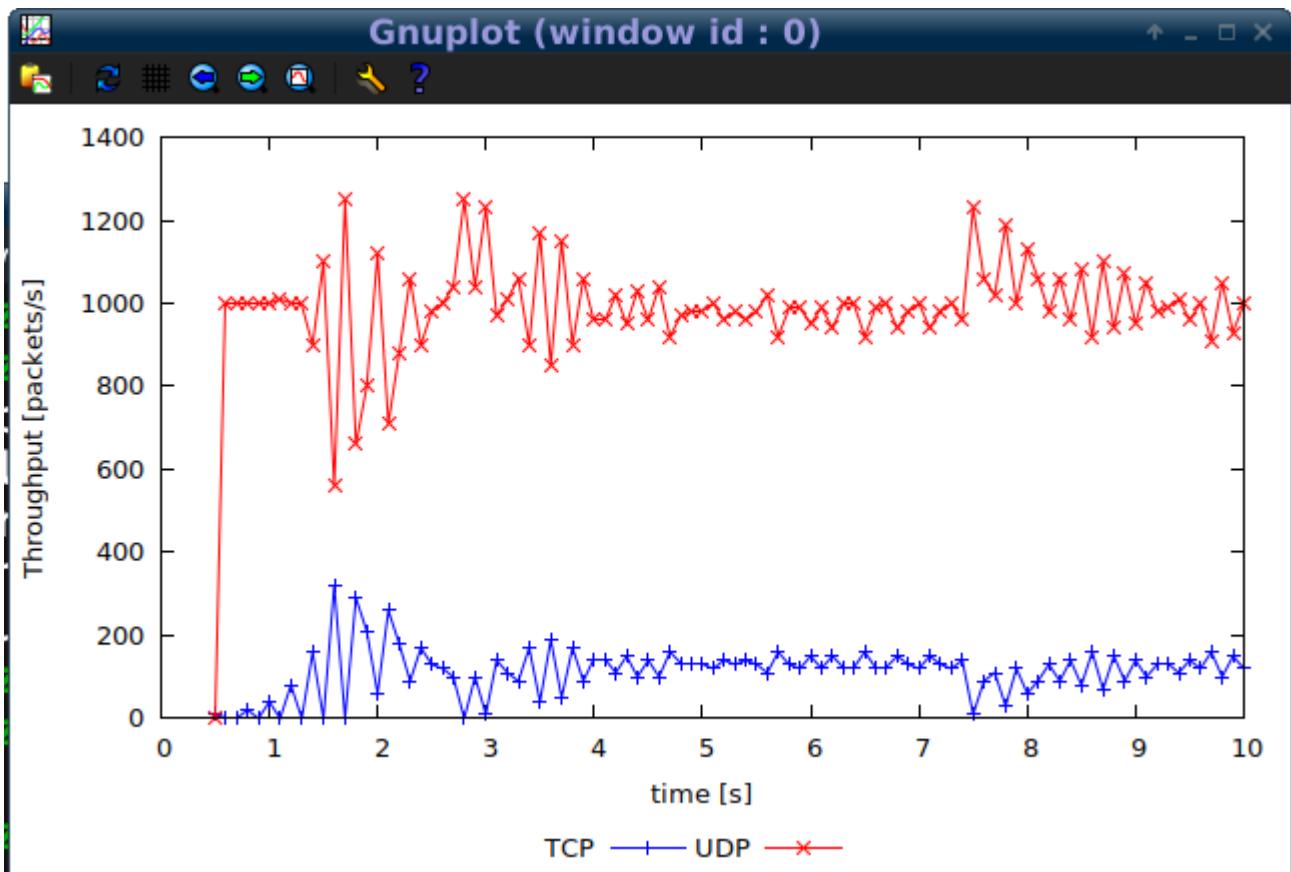
**Question 2. What happens to the throughput of the pre-existing TCP flows when a new flow is created? Explain the mechanisms of TCP which contribute to this behaviour. Argue about whether you consider this behaviour to be fair or unfair.**

From the picture, the first few connections has pretty high throughput and the throughput decreses a lot when a new flow is created. After all flows are created, they are average over time.

The congestion control mechanisms will contribute to this behaviour.

I think this is a fair behaviour. Since all flows are under the same network condition, the same mechanism will be used. In addition, each connection adjusts the size of the connection window when setting up a new connection to allow sharing of the common link.

## 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 expect the UDP throughput is higher than the TCP throughput if the capacity of the link is 5Mbps. The red line represents the UDP and the blue line is the TCP.

**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 does not have congestion control maner and TCP has congestion contol. Therefore, the UDP transfers the packet at relatively constant speed regardless of dropping packets in the transmission. And the TCP will detect congestion abd reduce the congestion window size.

**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: transfer the file fast because UDP could dominate the throughput in a shared link

Disadvantages: it will cause congestion without congestion control mechanisms  
If everyone started using UDP, the network will collapse.