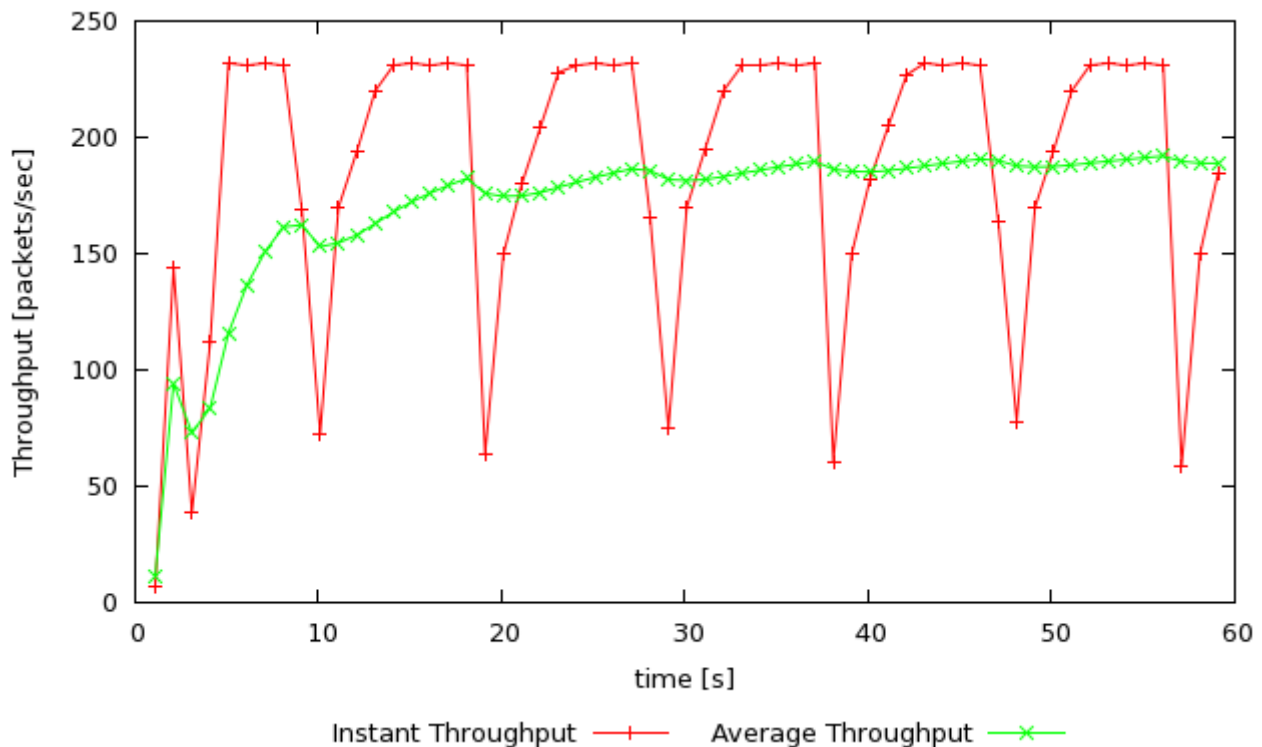


## Lab 5 Report

by Nicholas Puglia z5115237

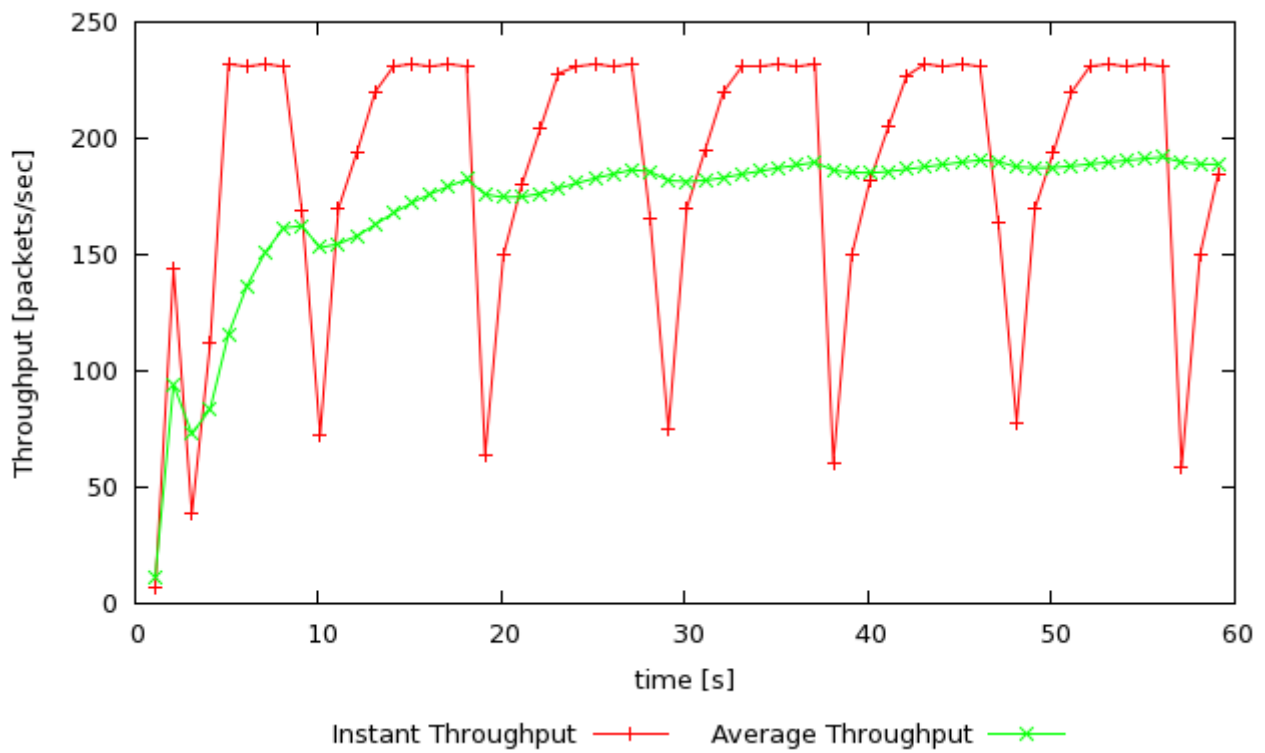
### Exercise 1

1. The maximum size of the congestion window is 100. When the congestion window reaches this value, there is a timeout and the congestion window returns to its starting value. The TCP flow then commences slow start which increases at an exponential rate until reaching ssthresh which is now half of the initial ssthresh. After the congestion window reaches this number it begins increasing the window at a linear rate. This is known as congestion avoidance.
2. From the graph when running the command `gnuplot WindowTPut.plot` the average throughput is approximately 190 packets/sec

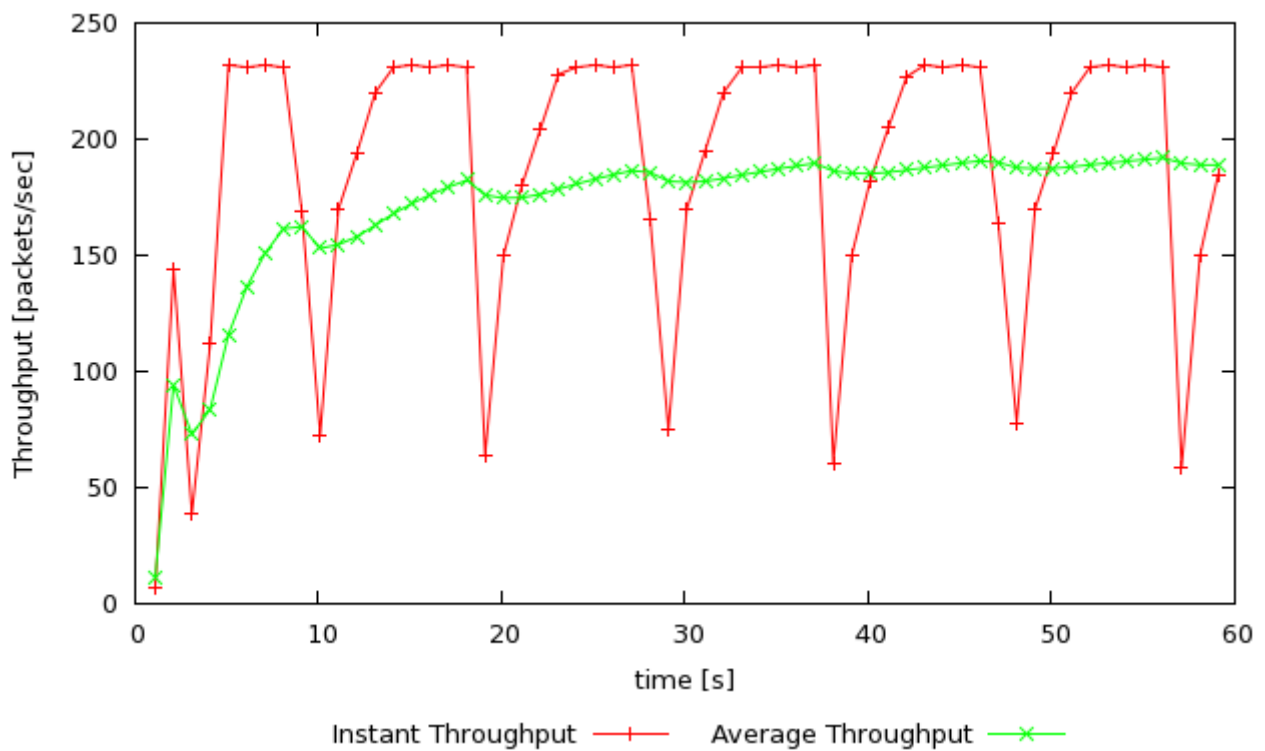


ns tpWindow.tcl 150 100ms

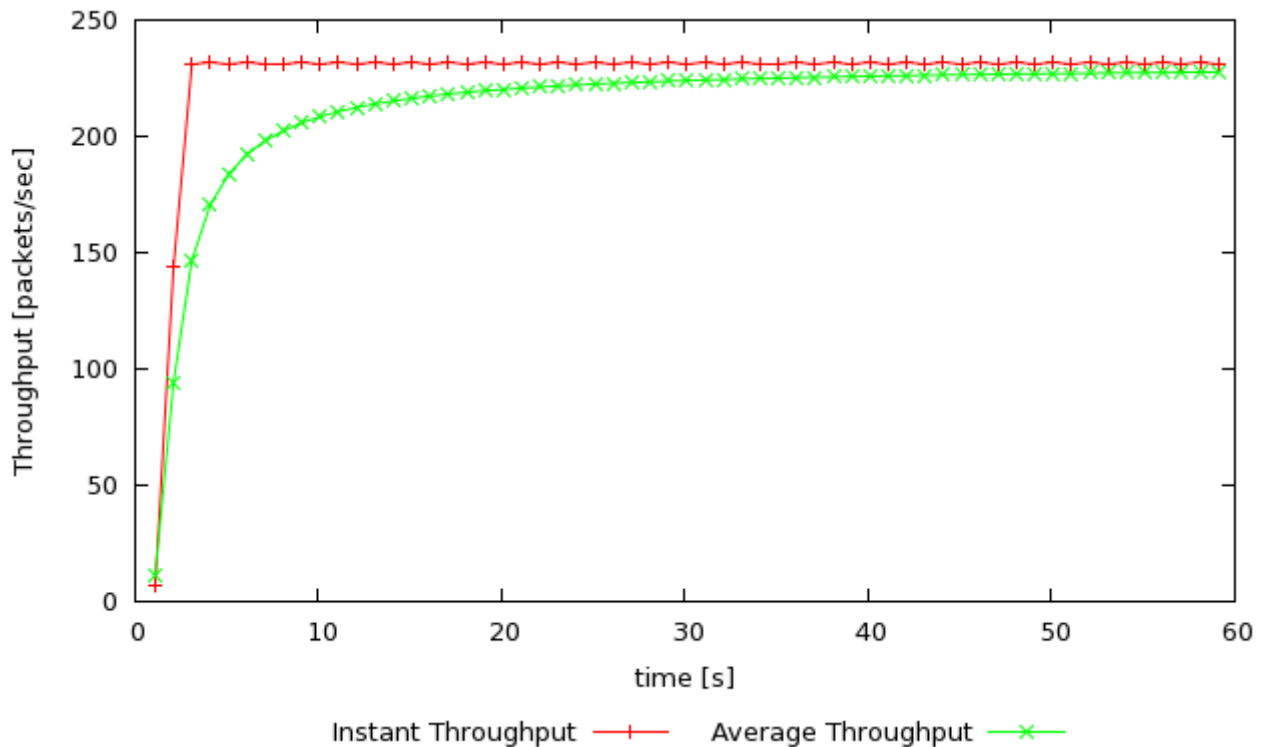
3. When changing the max congestion window the throughput graph still remains the same. The maximum congestion window when throughput is stable is 50. The average throughput at this point is approximately 230 packets/sec. If each packet is 540 bytes and 230 packs are transmitted in 1 second then the total bytes transmitted in 1 second is 124200. Comparing this to the link capacity of 1000000 bps. Only 12.42% of the link capacity is used.



ns tpWindow.tcl 250 100ms



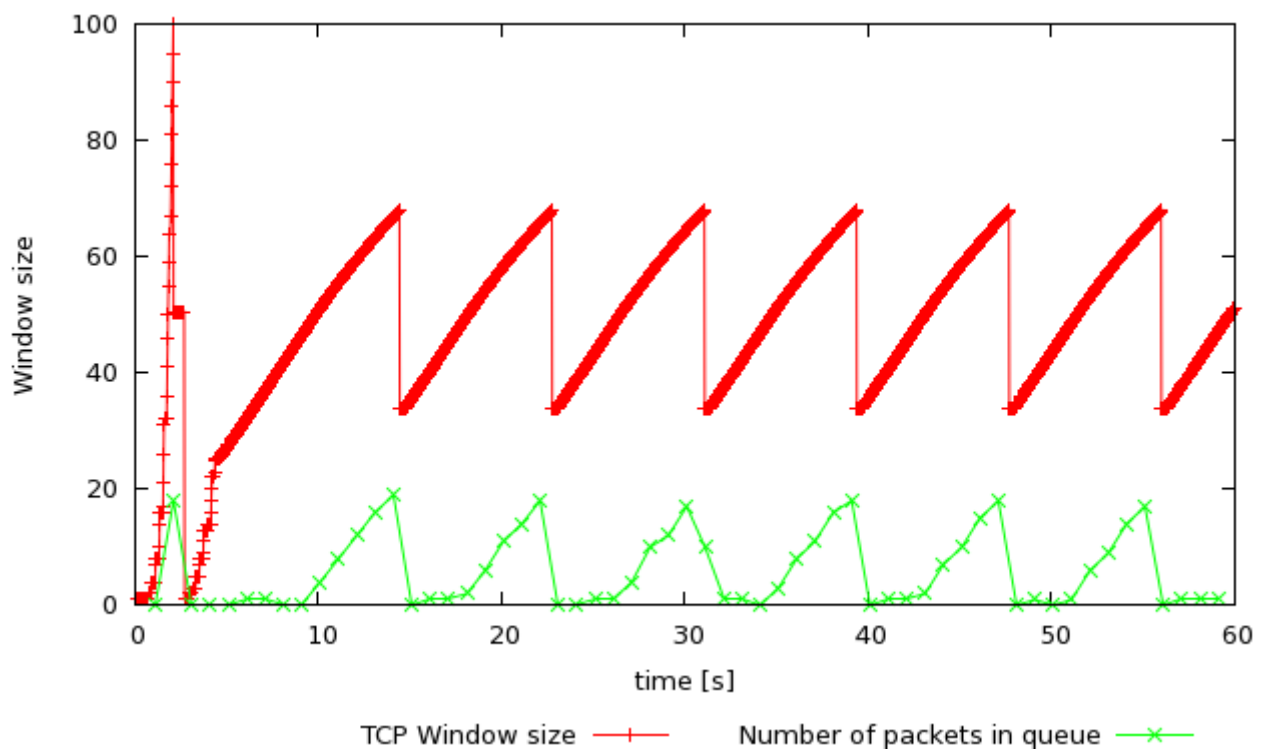
ns tpWindow.tcl 350 100ms



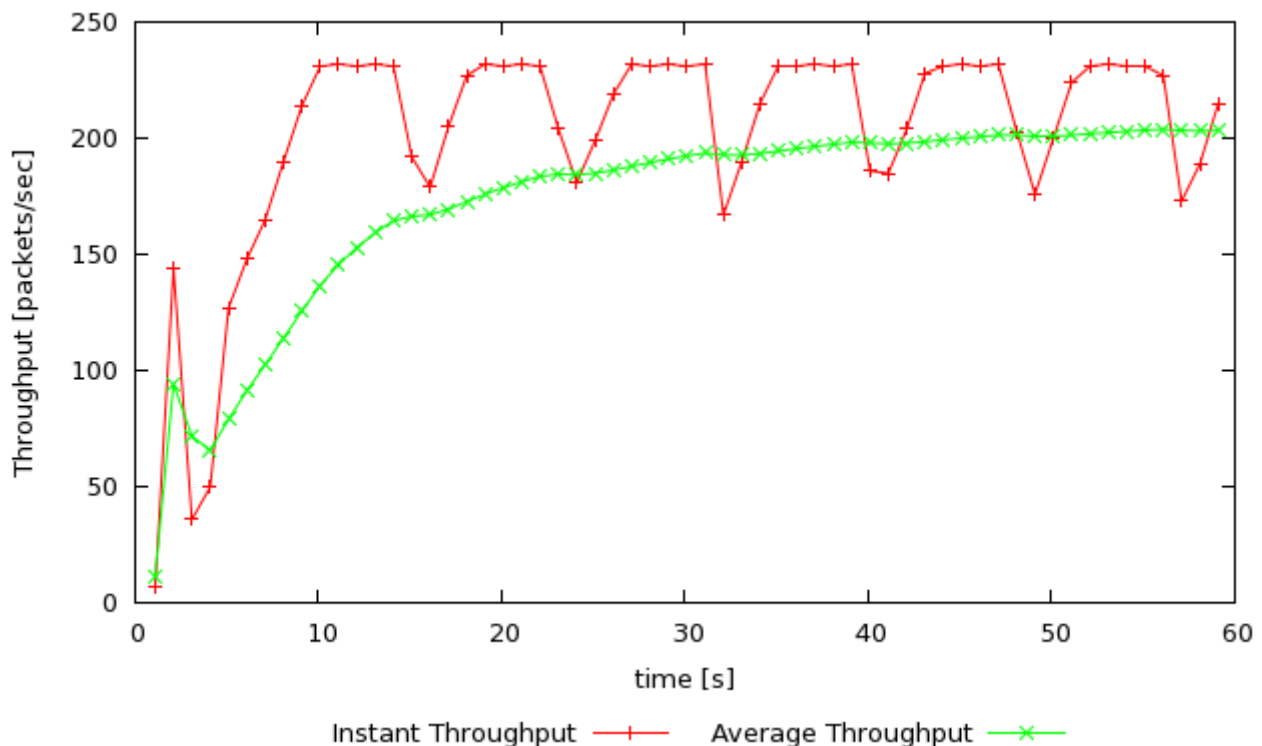
ns tpWindow.tcl 50 100ms

4. The differences in TCP Tahoe and Reno is the amount of times the congestion window goes back to zero. In TCP Tahoe, the congestion window is reset to zero seven times, whereas in TCP Reno only resets to congestion window once. The average throughput therefore for TCP Reno would be higher as the congestion window is not set to zero as much as TCP Tahoe. This is evident in the graph as the average throughput for Tahoe is approx 130 packets/sec while Reno is 200 packets/sec.

#### TCP RENO



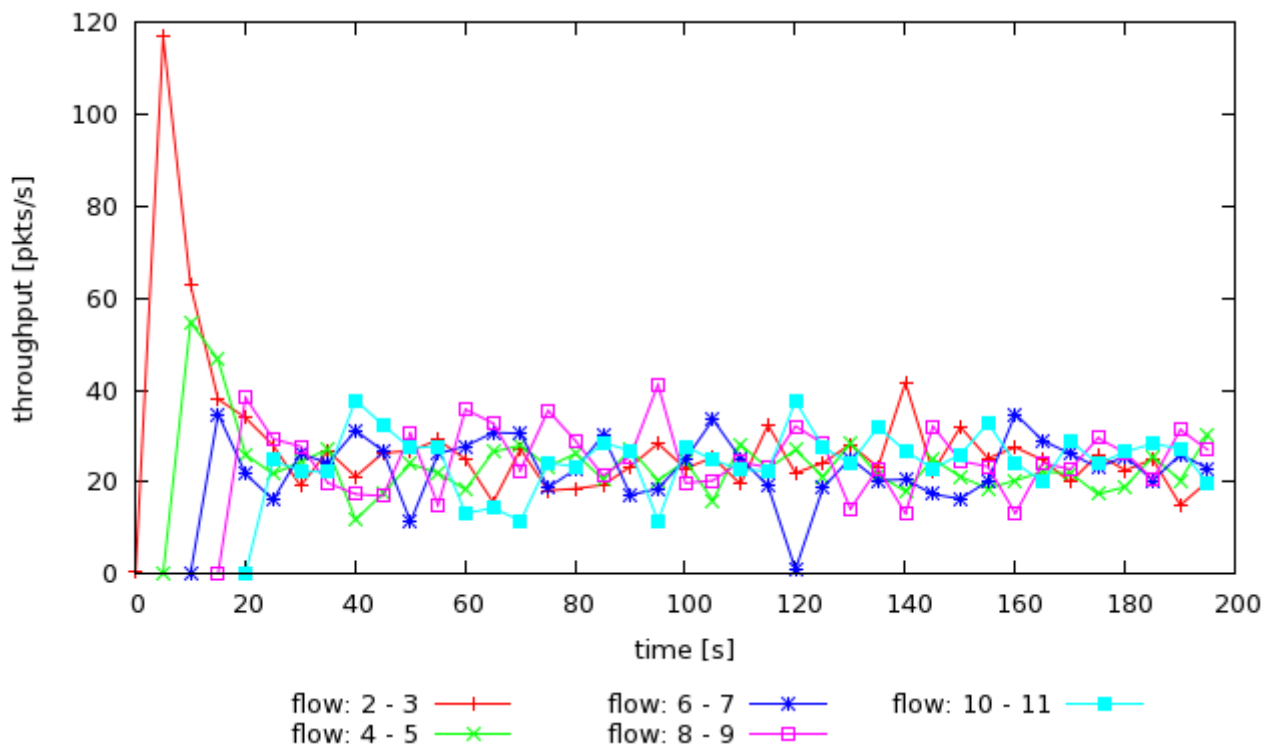
ns tpWindow.tcl 150 100ms



ns tpWindow.tcl 150 100ms

## Exercise 2

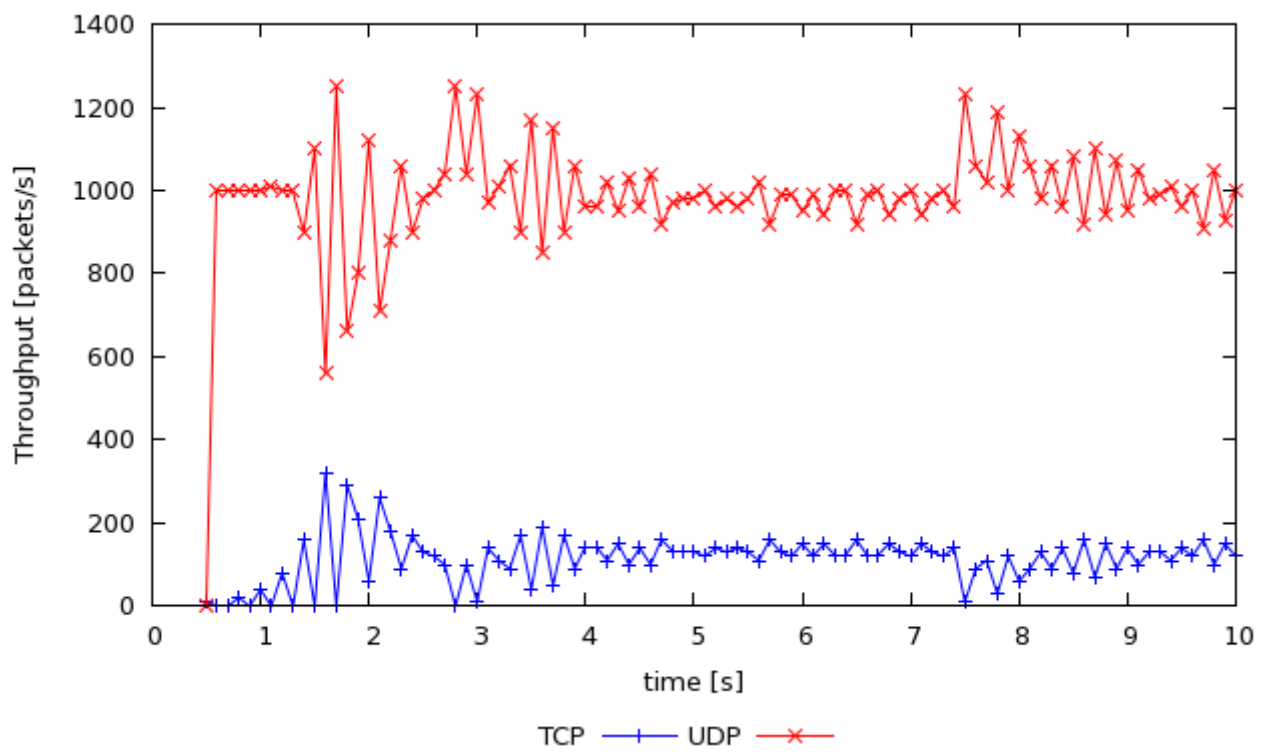
1. I think that TCP is reasonably fair. In the graph of fairness\_pps each flow shares an approximate equal amount if the common link. However they all do not have to exact amount of throughput, rather each flow fluctuates around an equal share of throughput.
2. When new flows are creates the existing TCP flows experience a decrease in throughput to account for a demand of throughput from other flows. I think that this behaviour is fair as it accounts for all TCP flows and treats them equally. The TCP mechanism that contributes to this behaviour is AIMD. This is additive increase and multiplicative decrease. By using this strategy, the throughput of flows on a common link will average to  $1/N$  packets/sec for each flow if  $N$  number of TCP session share a link.



ns tcp\_fairness.tcl

### Exercise 3

1. I expect the TCP flow will experience less throughput than the UDP flow as TCP tries to implement congestion control and hence the loss of packets, while UDP does not.
2. The UDP flow experiences higher throughput than the TCP flow due to the congestion control of TCP as well as TCP fairness. For TCP the mechanism that forces the throughput to stabilise is AIMD which is additive increase and multiplicative decrease. This means that the throughput will converge to the  $1/N$  packets/sec where  $N$  is the number of TCP sessions on a link.
3. The advantages of TCP congestion control is the idea of fairness across the network. This means that all users using the same network are able to receive a fair share of throughput. An advantage of UDP is increased throughput for users however, UDP does not implement any form of congestion control and therefore, one main disadvantage is that there is no sense of fairness in terms of users sharing throughput. TCP when compared to UDP has a disadvantage of slower throughput however, in a large network TCP's benefits outweigh its negatives. If everyone started using UDP instead of TCP the network would get congested very quickly and as there are no means to reduce this congestion, many packets will be lost, resulting in a very inefficient file transfer system.



ns tp\_TCPUDP.tcl 5Mb