Written by XiaoHu z5223731
Exercise 1 Answer:

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:

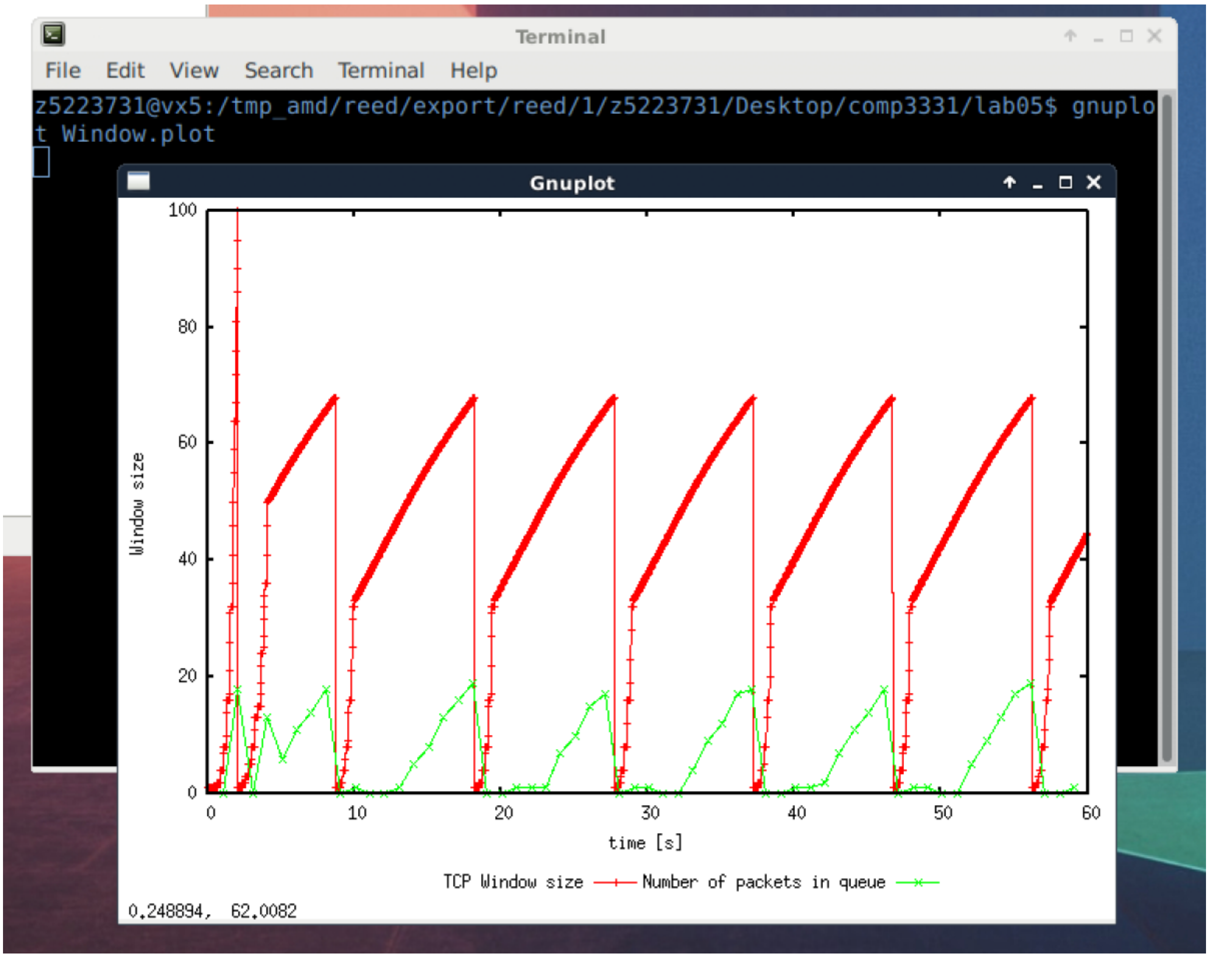
```
$ns tpWindow.tcl 150 100ms
```

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.

Answer:



Maximum size of the congestion window the TCP flow reached is 100, the TCP flow reduce the window size equal to 1 and the threshold is set to half of 100. The reason is when the

TCP flow reaches the value, the queue will be full and packets will be dropped and this make the timeout or triple dup ACK.
Then it will start a new slow-start, and keep increasing till it reach the threshold then it will do AMID till the queue is full and drop the packets again, after that it will reduce the window size to 1 and do a new slow-start again. It will do the above process over and over again.

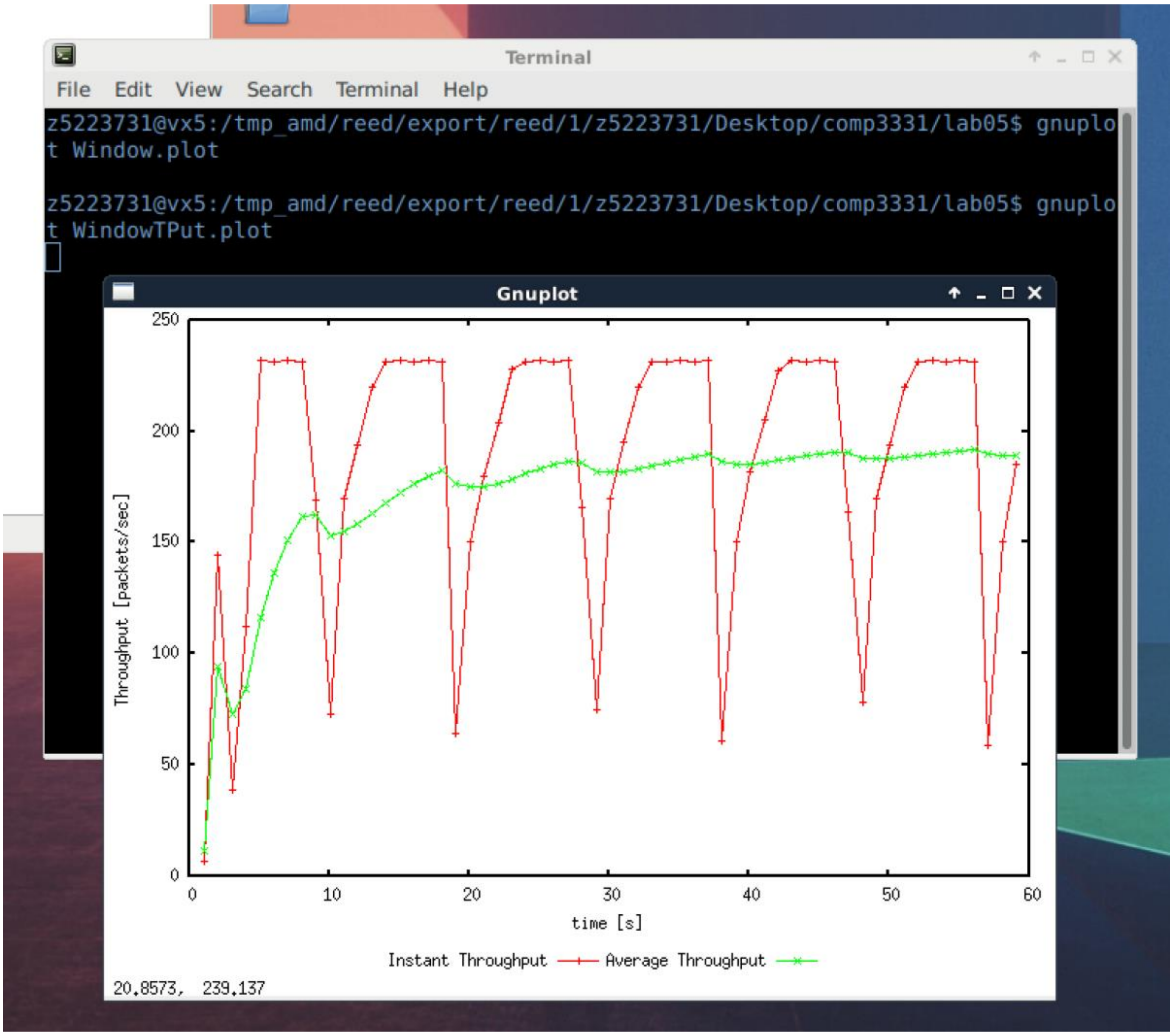
Question 2: From the simulation script we used, we know that the payload of the packet is 500 Bytes. Keep in mind that the size of the IP and TCP headers is 20 Bytes, each. Neglect any other headers. What is the average throughput of TCP in this case? (both in number of packets per second and bps)

You can plot the throughput using the provided gnuplot script WindowTPut.plot as follows:

```
$gnuplot WindowTPut.plot
```

This will create a graph that plots the instantaneous and average throughput in packets/sec. Include the graph in your submission report.

Answer:



The average packets per second is approximately 190 from the graph.

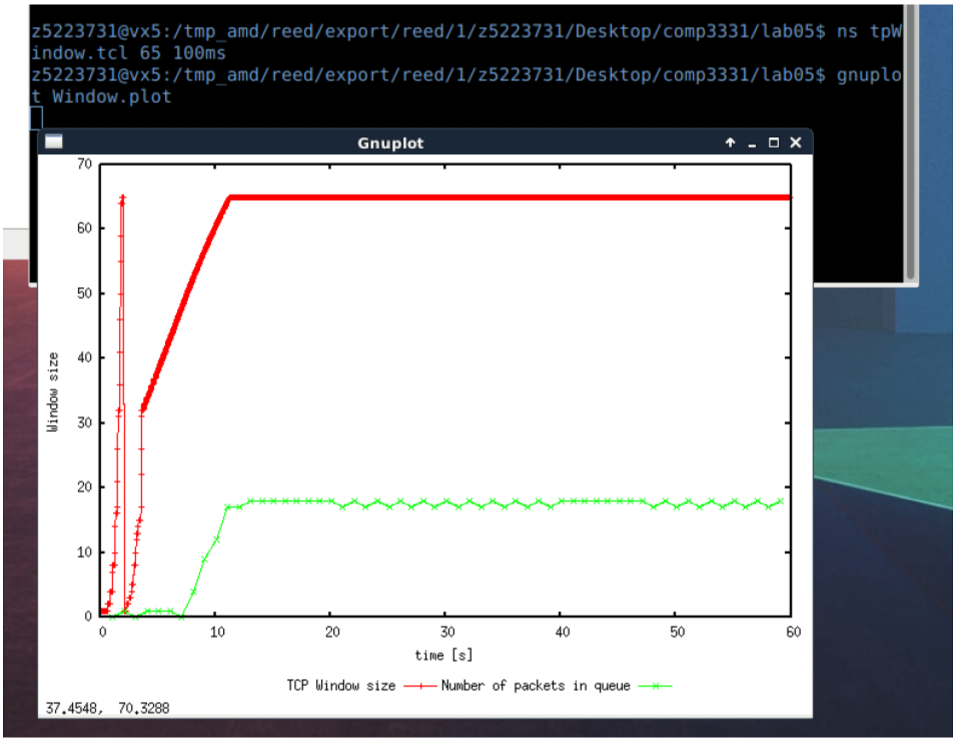
Average throughput:
Include header and payload.
190 * (500 + 20 + 20) = 102.6kb/s = 820.8kbps

Average throughput:
Without header.
190 * 500 = 95kb/s = 760kbps

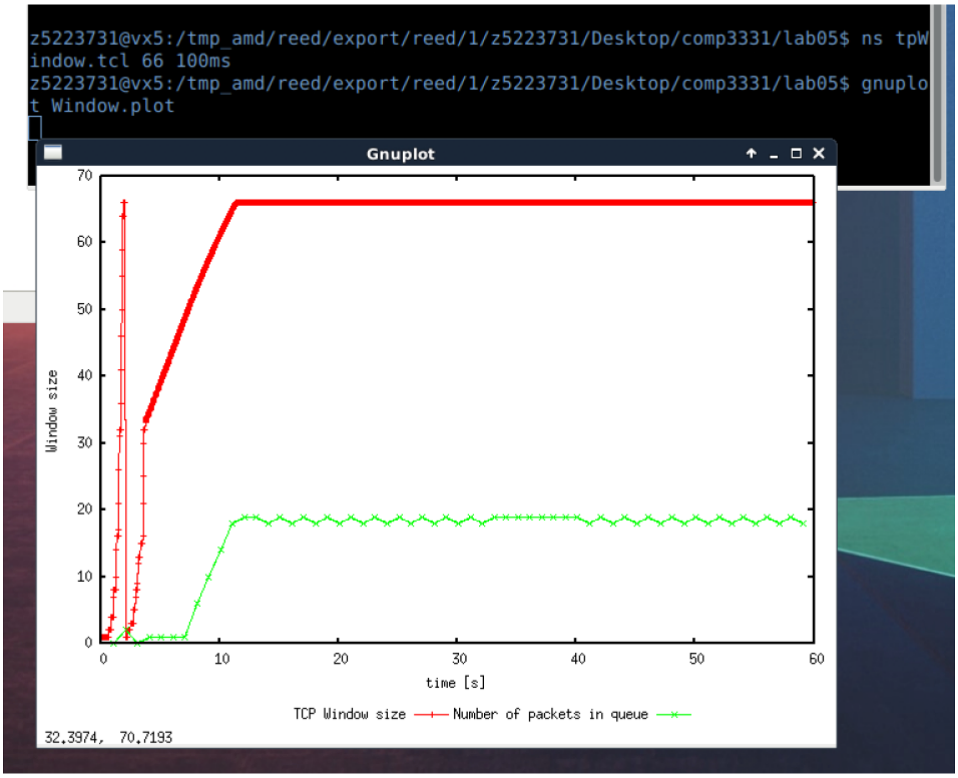
Question 3: Rerun the above script, each time with different values for the max congestion window size but the same RTT (i.e. 100ms). How does TCP respond to the variation of this parameter? Find the value of the maximum congestion window at which TCP stops oscillating (i.e., does not move up and down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

Answer:

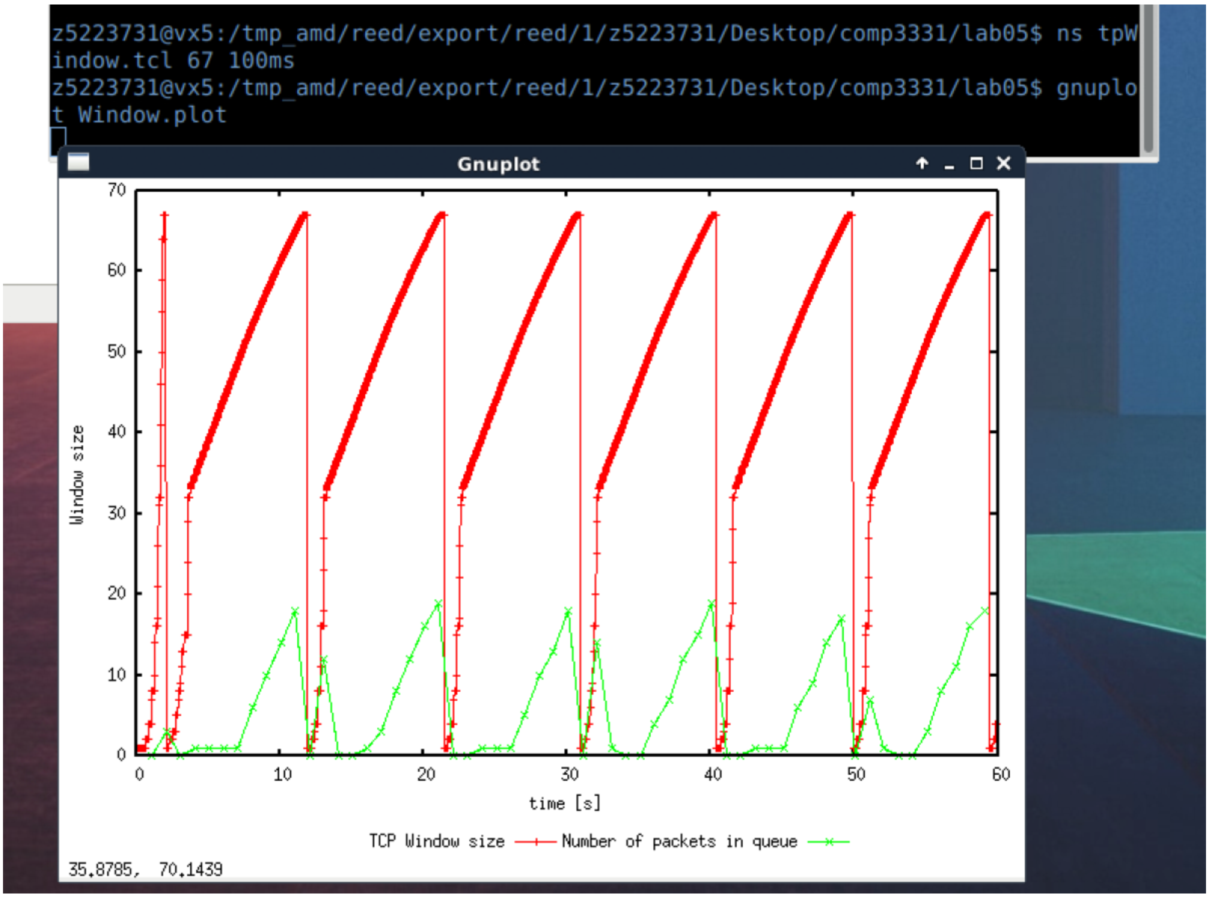
Window size 65:



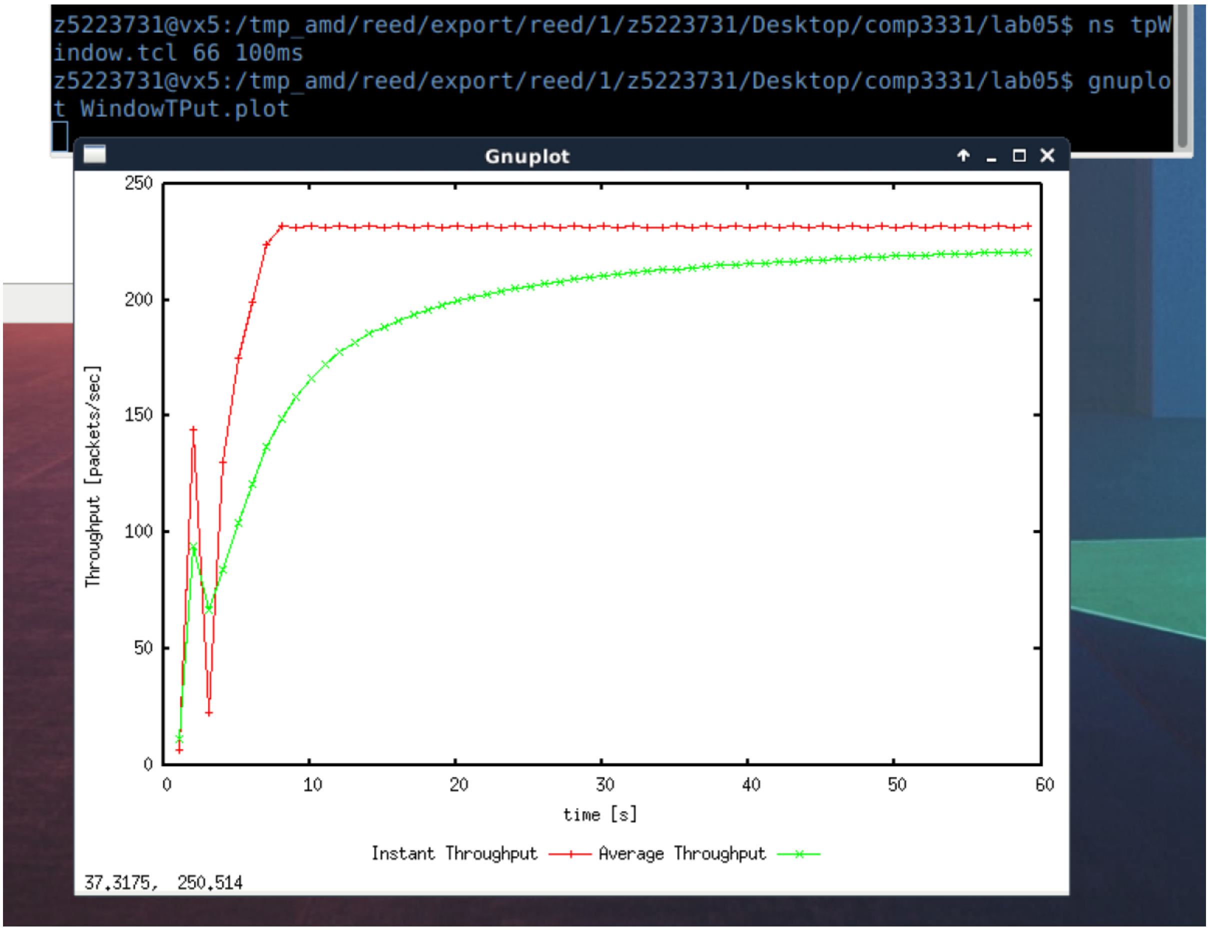
Window size 66:



Window size 67:



Hence the max congestion window size is 66.



The average packets per second is approximately 225 from the graph.

Average throughput:
225 * 500 = 112.5kb/s = 900 kbps = 0.9 Mbps.
The average throughput is almost equal to the link capacity.

TCP Tahoe vs TCP Reno

Recall that, so far we have observed the behaviour of TCP Tahoe. Let us now observe the difference with TCP Reno. As you may recall, in TCP Reno, the sender will cut the window size to 1/2 its current size if it receives three duplicate ACKs. The default version of TCP in ns-2 is TCP Tahoe. To change to TCP Reno, modify the Window.tcl OTcl script. Look for the following line:

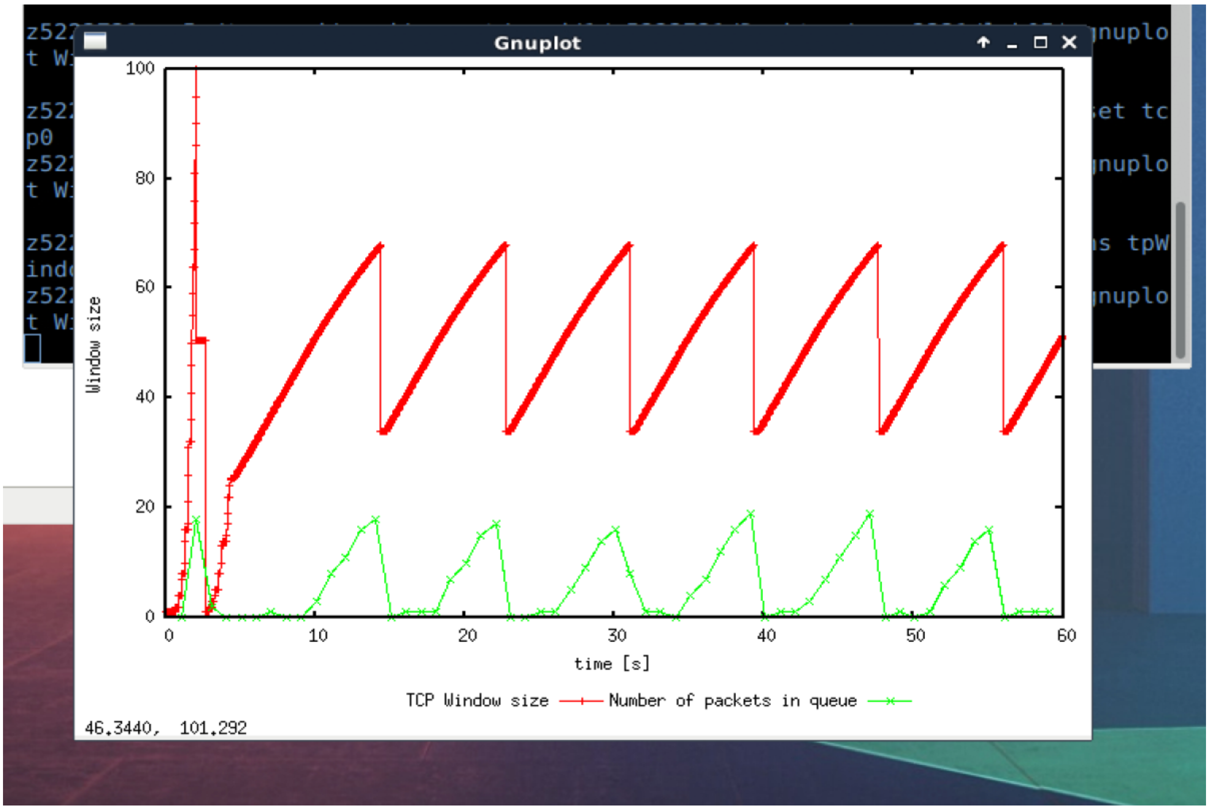
```
set tcp0 [new Agent/TCP]
```

and replace it with:

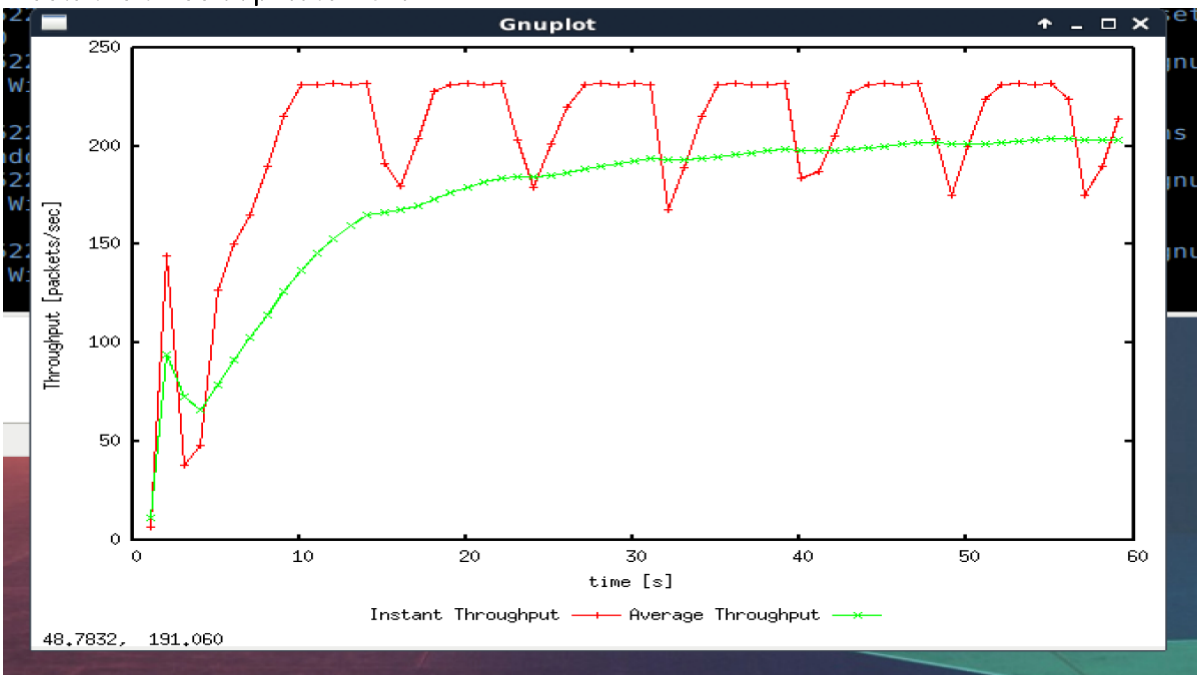
```
set tcp0 [new Agent/TCP/Reno]
```

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?

Answer:



Q1: Reno only go back to zero once and doesn't have the slow-start. Reno only reduce the current window size to half when meet the congestion event, and increase linearly once it meets the three duplicate ACKs.

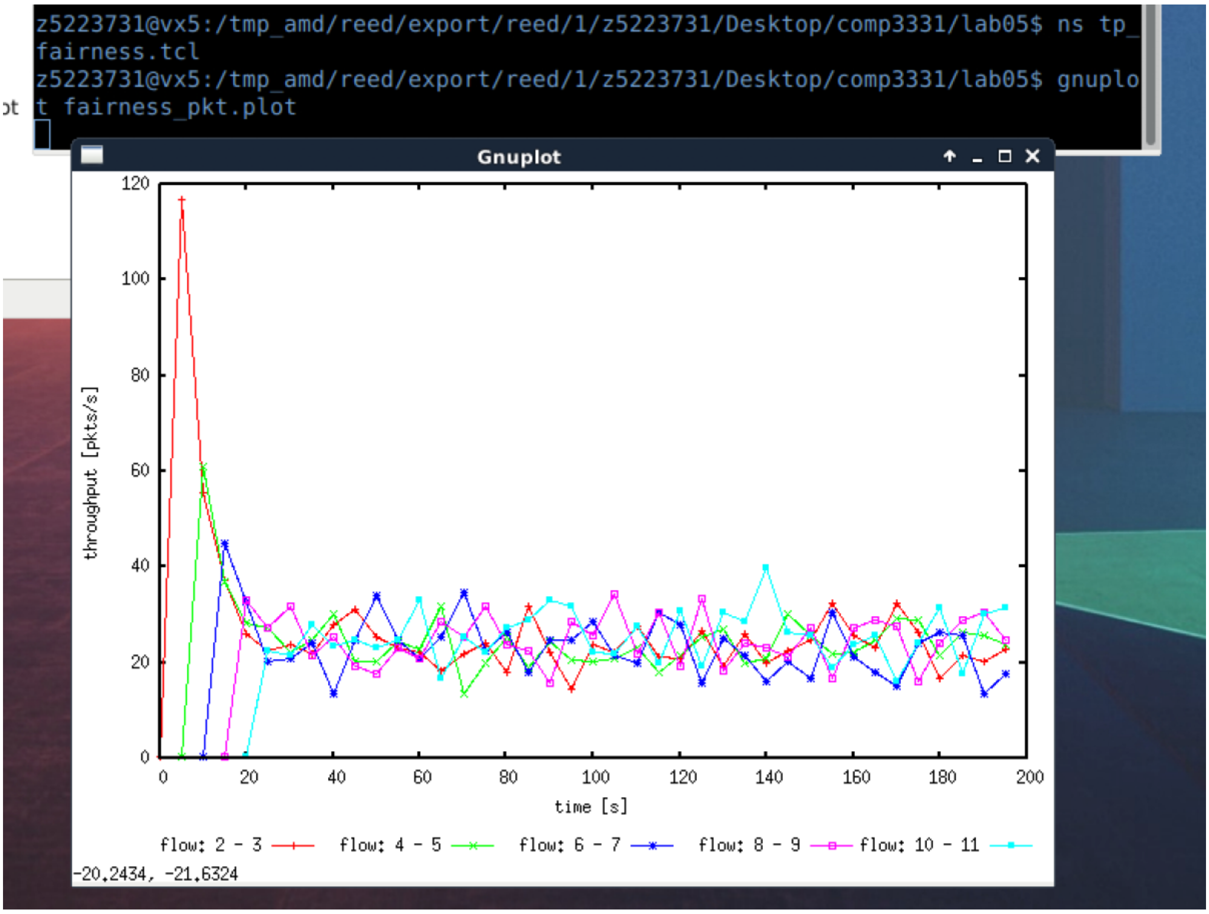


Q2: Reno average throughput is approximately 200 packets/s which is higher than Tahoe because the Tahoe have to set window size back to 1 and do the slow-start once it meets the three ACKs but Reno doesn't.

Exercise 2 Answer:

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.

Answer:



Yes, it does.
At the first 20s, each flow share the capacity of the common link differently but after 20s the throughout of each flow is 20 – 40 and they are roughly similar when each of them change their window size to achieve long term fairness.

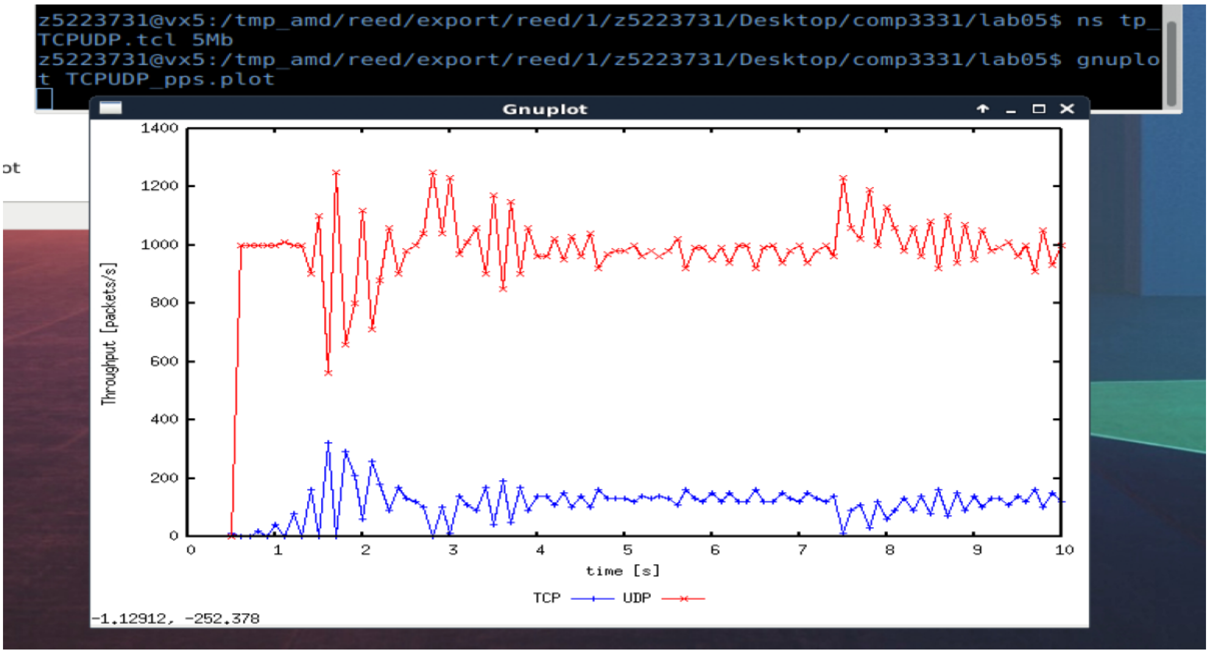
Question 2. What happens to the throughput of the pre-existing TCP flows when a new flow is created? Explain the mechanisms of TCP which contribute to this behaviour. Argue about whether you consider this behaviour to be fair or unfair.
Answer:
The throughput of pre-existing TCP flows will reduce when there is a new flow created. The reason is the new flow will do slow-start and create a congestion, the rest of flows will

detect the congestion and reduce the window size to avoid overwhelm. It is fair when a new flow is create and the rest of them will have less fair share capacity of the common link.

Exercise 3 Answer:

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

Answer:
UDP will transmit at scheduled rate due to it doesn't have congestion control, the TCP will use the rest of capacity.

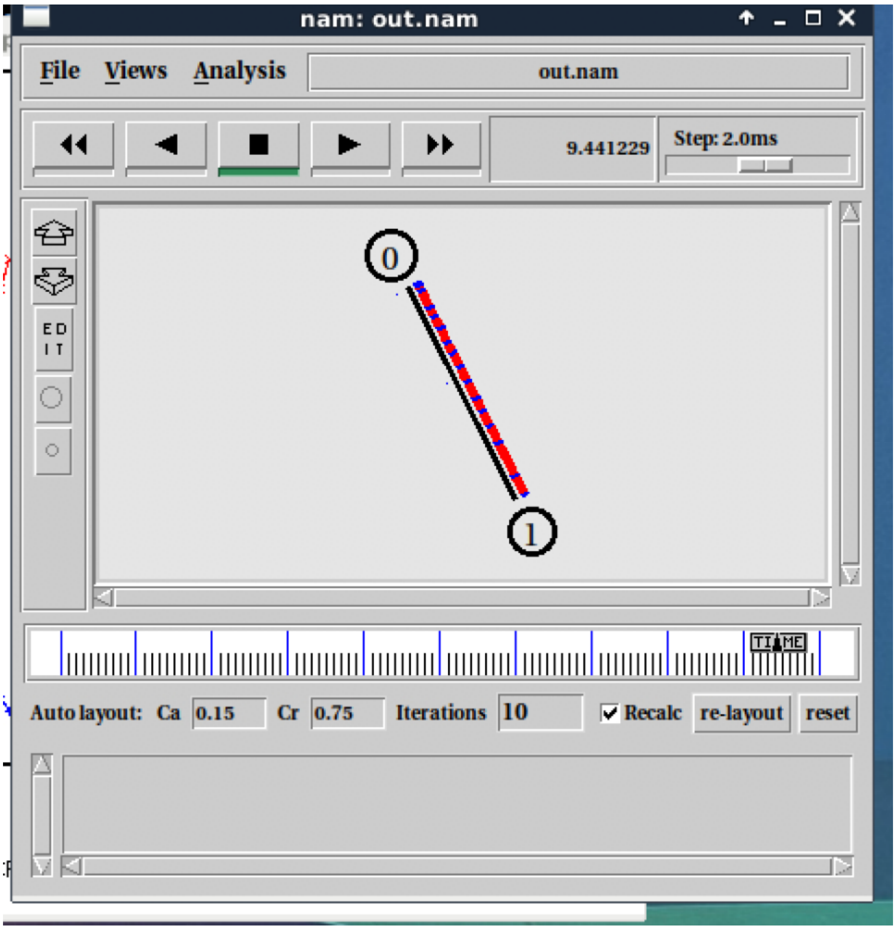


Now, you can use the simulation to test your hypothesis. Run the above script as follows:

```
$ns tp_TCPUDP.tcl 5Mb
```

The script will open the NAM window. Play the simulation. You can speed up the simulation by increasing the step size in the right corner. You will observe packets with two different colours depicting the UDP and TCP flow. Can you guess which colour represents the UDP flow and the TCP flow respectively ?

Answer:
Blue is TCP another one is UDP.



You may disable the NAM visualiser by commenting the "exec nam out.nam &' line in the 'finish' procedure.
Plot the throughput of the two flows using the above script (TCPUDP_pps.plot) and answer the following questions:

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.

Answer:
The UDP flow achieve higher throughput than TCP because the UDP doesn't have congestion control and doesn't care about the packets loss, hence it transmits at constant rate. However, the TCP have congestion control and does care about packets loss so that it will change window size based on network condition, as a result of this, it will reduce the rate of transmission.

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?
Answer:
Advantages: high transmission rate, less hearer, less packets size.
Disadvantages: packets loss, no congestion control, out of order.
If everyone use UDP instead, it will cause network congestion and as a result of this, everyone will loss many packets, the network will be overwhelm, the file could be broken.