COMP3331

Lab 4

Dean So (z5204873)

Exercise 1: Understanding TCP Congestion Control using ns-2

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:*

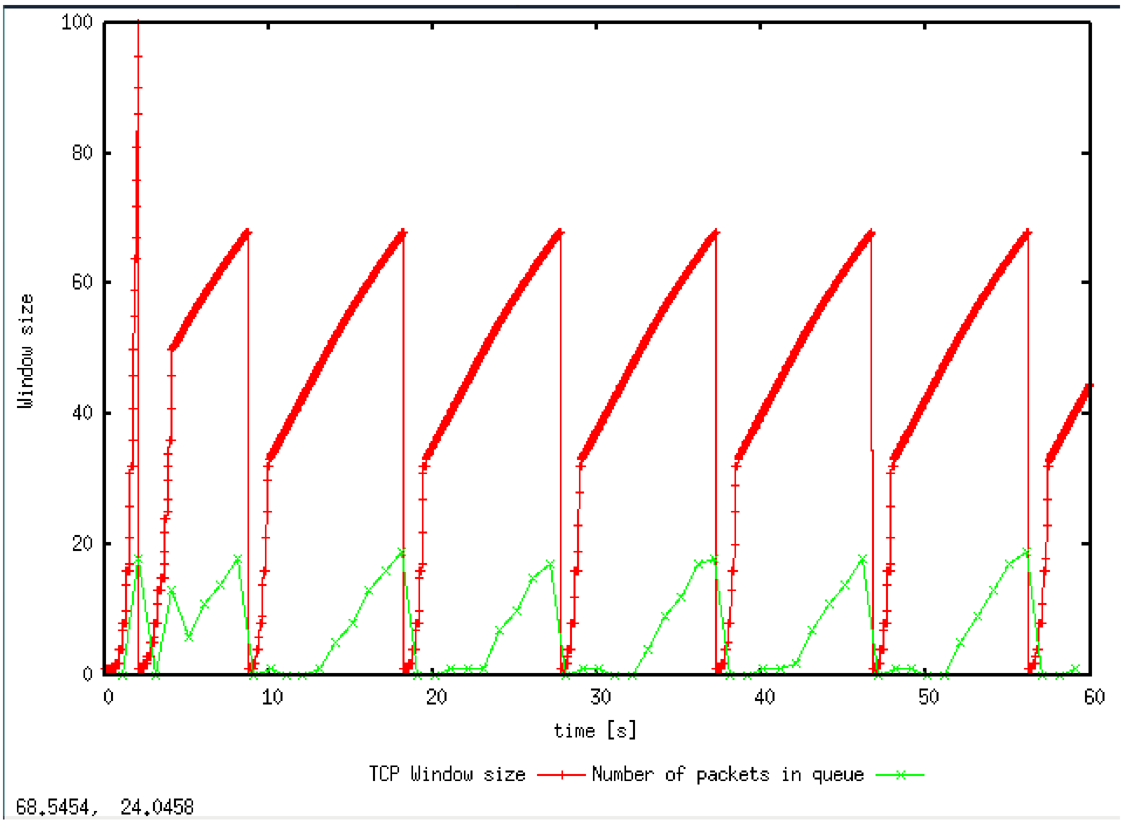
   ```
   $ns tpWindow.tcl 150 100ms
   ```

   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.

Maximum size window is 100 packets. The queue became full (the green graph) due to the increased window size and the sender dropped packets. The sender reduced the congestion window to 1. Then the slow start threshold became CWND/2 after the first loss event and increases the slow start threshold linearly.



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:

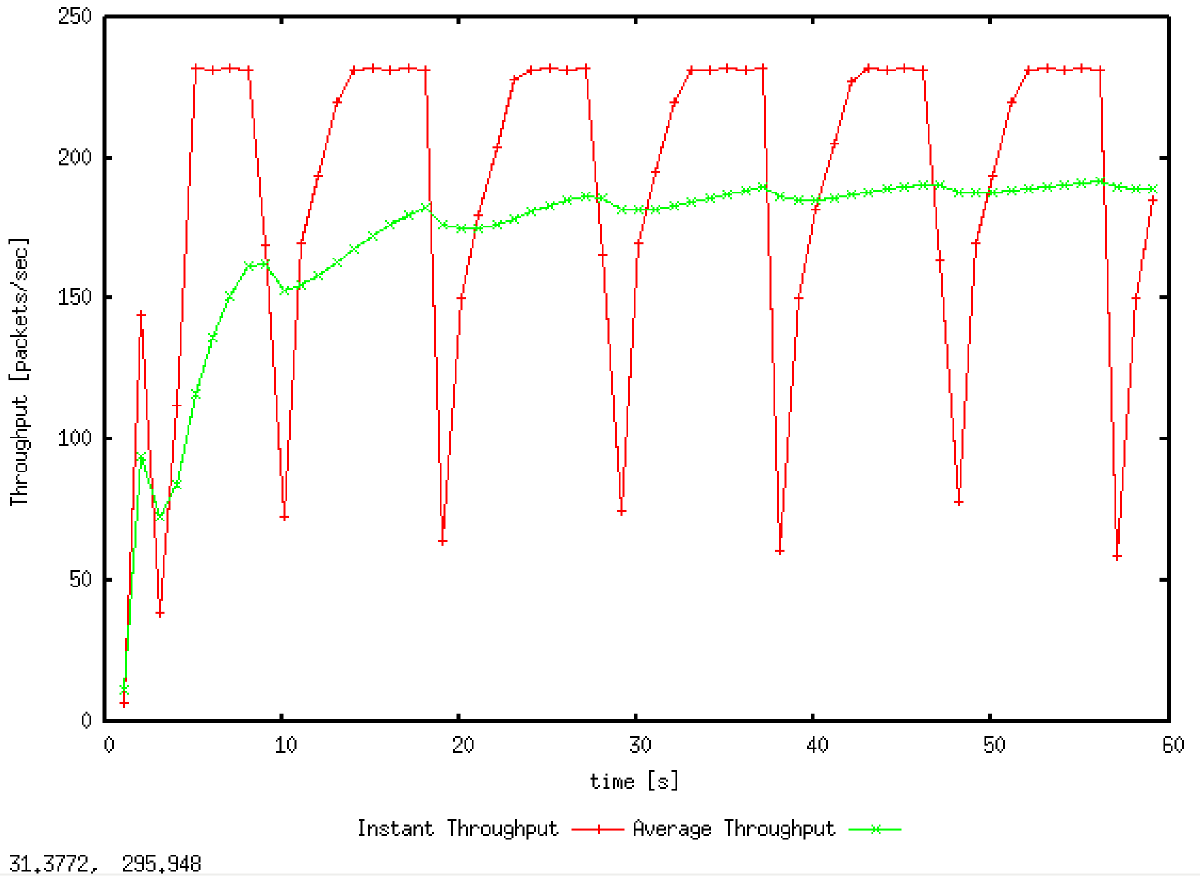
   ```
   $gnuplot WindowTPut.plot
   ```

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

From the graph, the average throughput is approximately 190 packets/sec thus,

average throughput (with header) = 190*(500+20+20)*8 = 820800 bits/sec

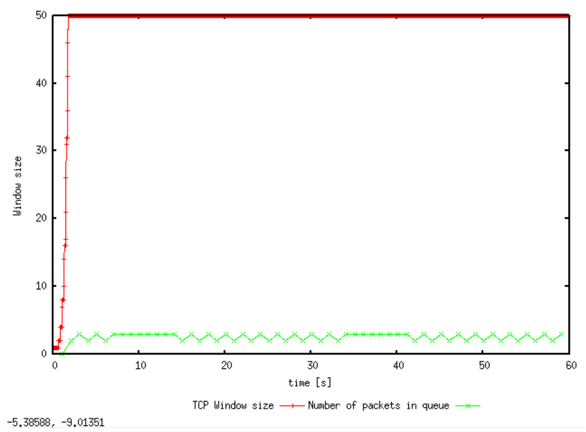
average throughput (without header) = 190*(500-20-20)*8 = 699200 bits/sec

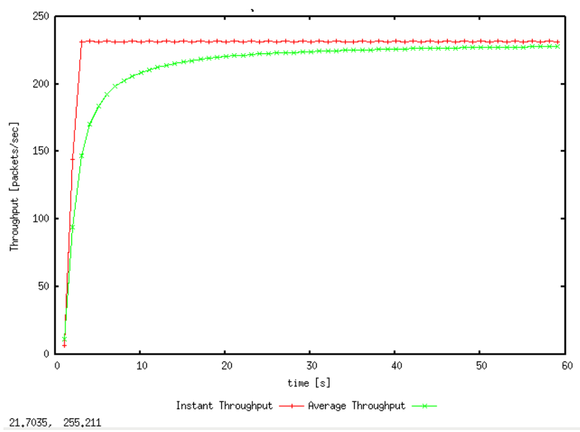


3. *Rerun the above script, each time with different values for the max congestion window size but the same RTT (i.e. 100ms). How does TCP respond to the variation of this parameter? Find the value of the maximum congestion window at which TCP stops oscillating (i.e., does not move up and down again)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

To avoid oscillation it was found that a maximum window size of 50 or lower was ideal and any window size above 66 yielded oscillation with essentially identical behaviour to the initial 150 window size.





Average throughput was 227 packets/sec which is (with header) 225*540 = 122580 bytes

The link capacity is 125000 bytes, therefore the average throughput is 98.064% of the link capacity.

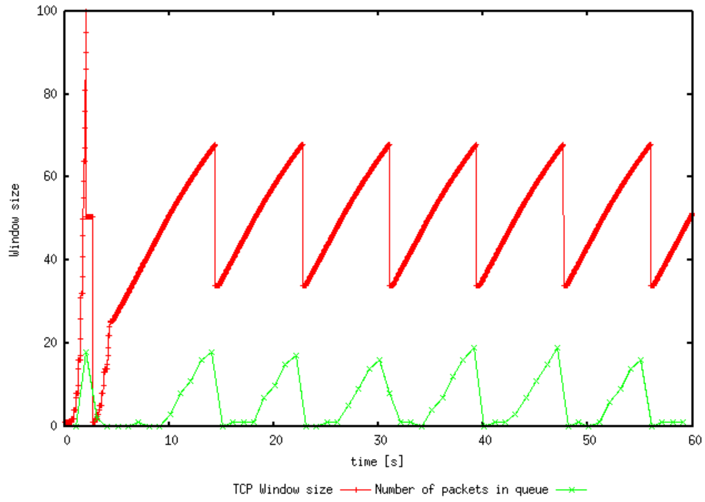
4. *Repeat the steps outlined in Question 1 and 2 (NOT Question 3) but for TCP Reno. Compare the graphs for the two implementations and explain the differences. (Hint: compare the number of times the congestion window goes back to zero in each case). How does the average throughput differ in both implementation*

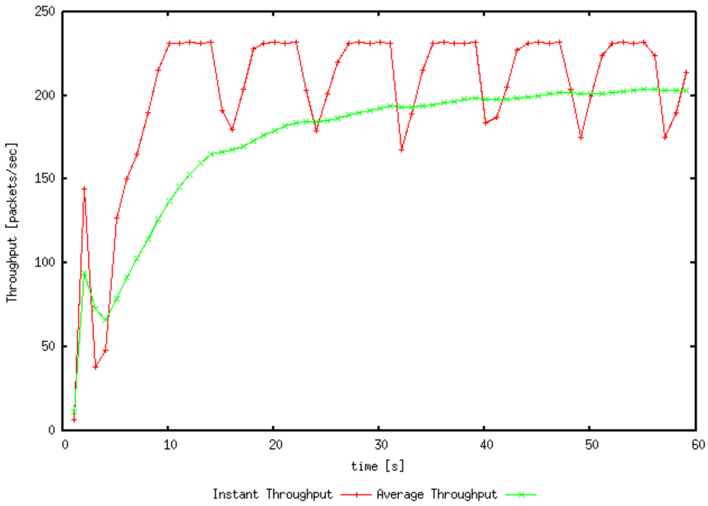
With TCP Reno, it does not re-enter a slow-start phase when loss occurs and the window doesn't reach 1. Also, average throughput is higher than TCP Tahoe.

TCP Reno throughput = 200 packets/sec

With header: 540*8*200 = 864000 bits/sec

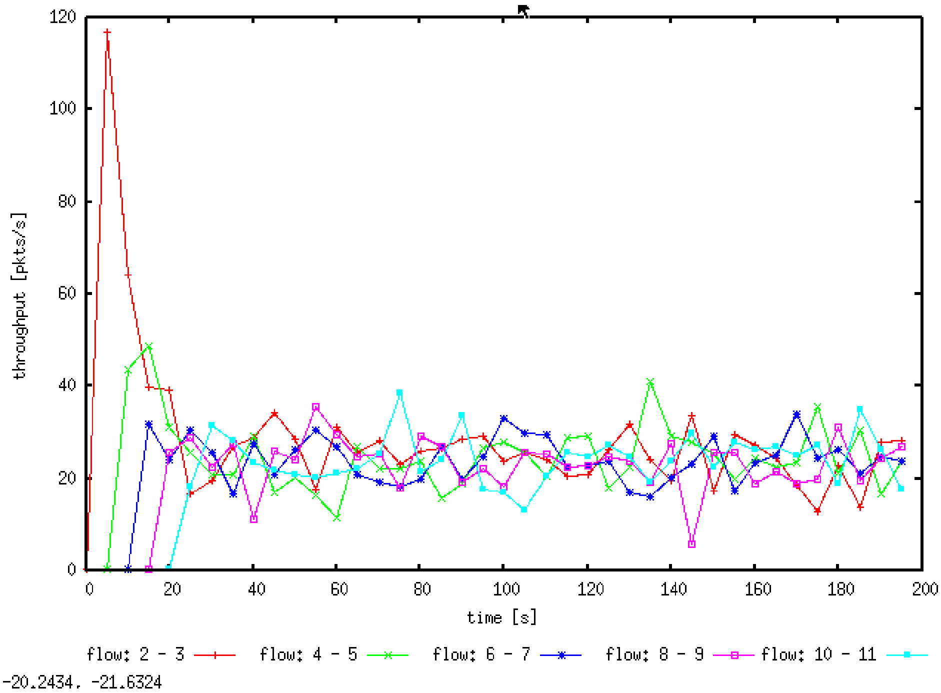
Therefore TCP Reno performs better compared with TCP Tahoe throughput.





10.9845. 262.551

Exercise 2: Flow Fairness with TCP



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.*

Although initially, some connections are higher after some time, the links clearly show that they get an equal share of the capacity of the common link.

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.*

Each time a new flow is created, the pre-existing TCP flows decrease to equalise between the other flows. Each connection adjusts the size of the connection window when a new connection comes in to allow for equal sharing of the common link. This is fair behaviour.

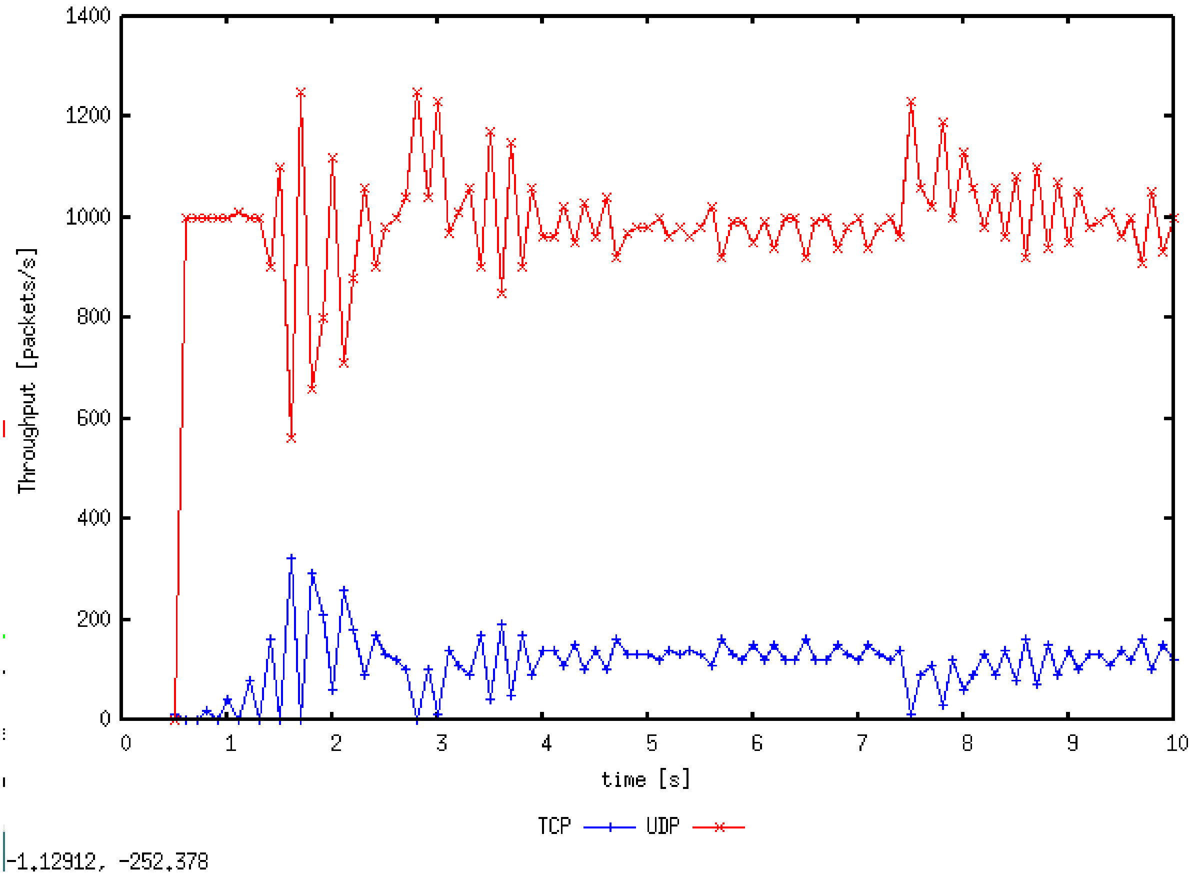
Exercise 3: Flow Fairness with TCP

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

It is expected that the UDP flow is higher than the TCP flow

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.*

UDP has achieve more throughput than TCP because UDP does not have an congestion control mechanisms, it has taken all of the bandwidth and TCP does have congestion control and thus TCP has adjusted as such.



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	Disadvantages
• Fast • Basic checksum for package corruption • No need to setup a connection	• Lacks congestion control • Doesn't check for package loss

If everyone started using UDP over TCP, the network would collapse as severe congestion would occur due to no congestion control mechanisms.