COMP9331 LAB5 –TCP Congestion Control

Changfeng LI(z5137858)

Exercise 1: Understanding TCP Congestion Control using ns-2

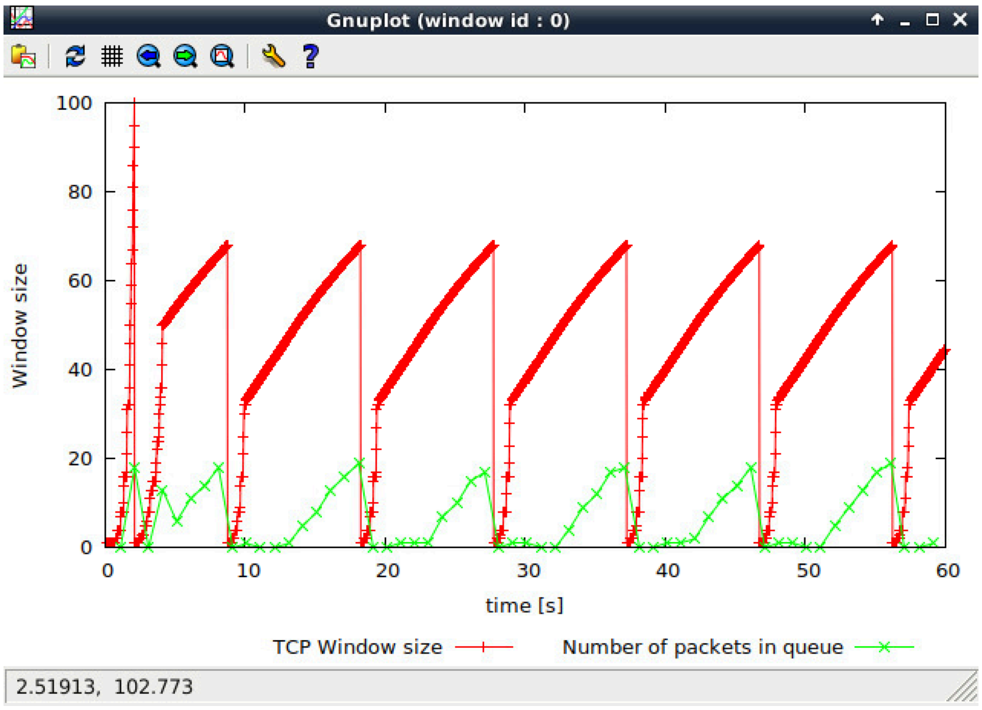
Qs1-1: What is the maximum size of the congestion window that the TCP flow reaches in this case? What does the TCP flow do when the congestion window reaches this value? Why? What happens next? I

Qs1-2: 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?

Qs1-3: 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 behavior. What is the average throughput (in packets and bps) at this point? How does the actual average throughput compare to the link capacity (1Mbps)?

Qs1-4: 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?

As1-1:



During the slow start phase, max congestion is approximately 100 packets, although we set the congestion window to 150 packets.

When the congestion size reaches this value, some packets are dropped. The reason is that buffer queue size is only 20 packets < 100.

It maintaining alternating back to the slow start phase from packet congestion.

As1-2:

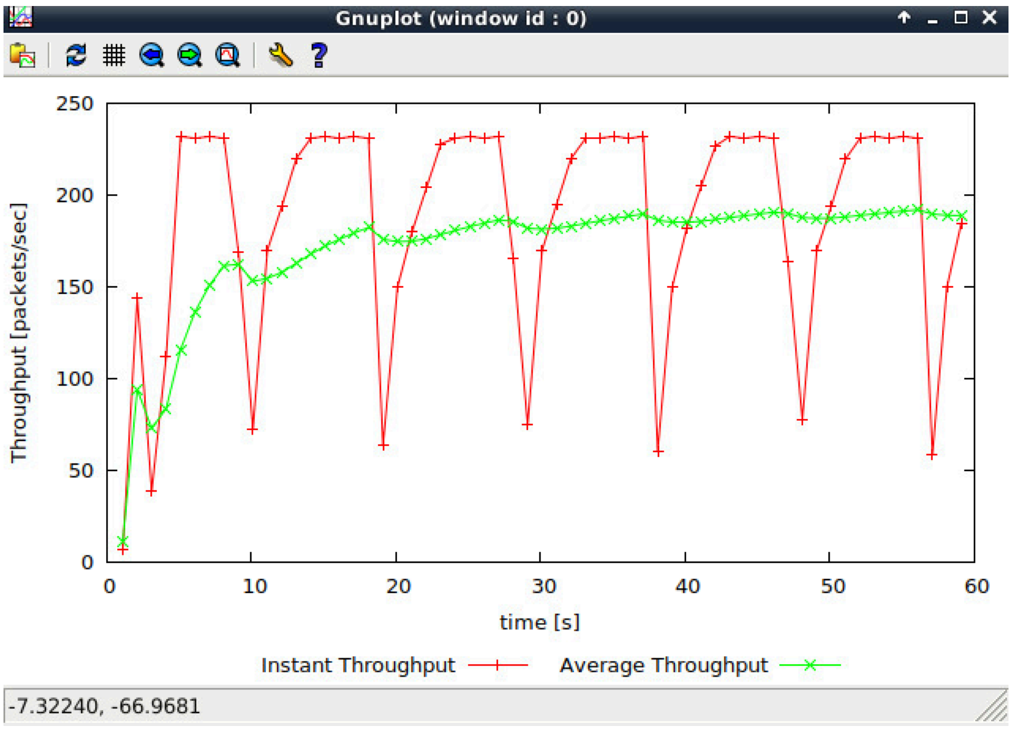
Payloads of packets = 500 Bytes

Size of Header = 20 + 20 = 40Bytes

1 Bytes = 8 bits

According to the graph, throughput of packets approximately equal to 180ms.

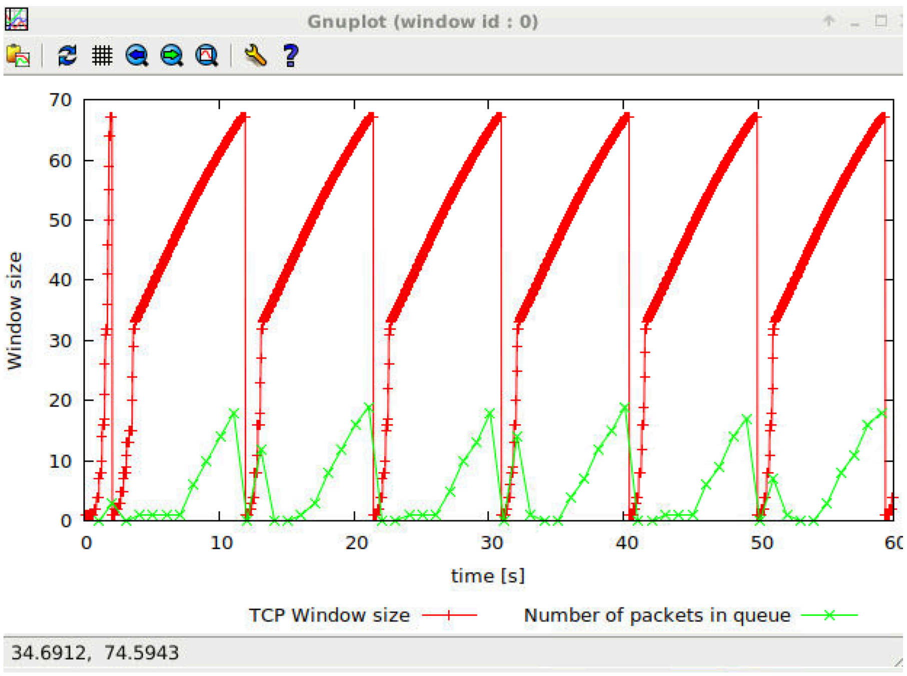
Throughput = (500 + 40) * 8 * 180 = 77.76 kbps



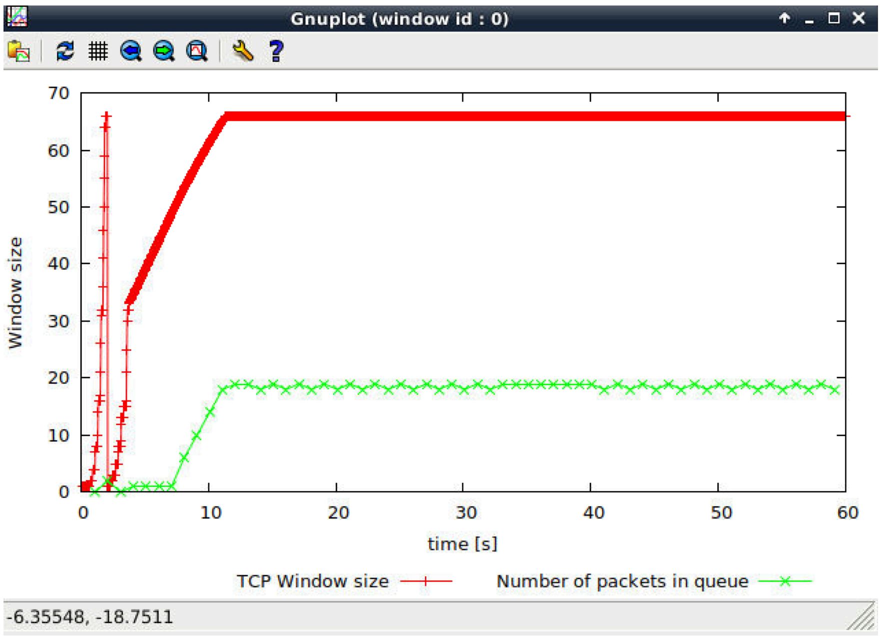
As1-3:

TCP responses by a general rule, by dropping package and return to slow start when maximum congestion window size grows greater than the maximum queue buffer size,

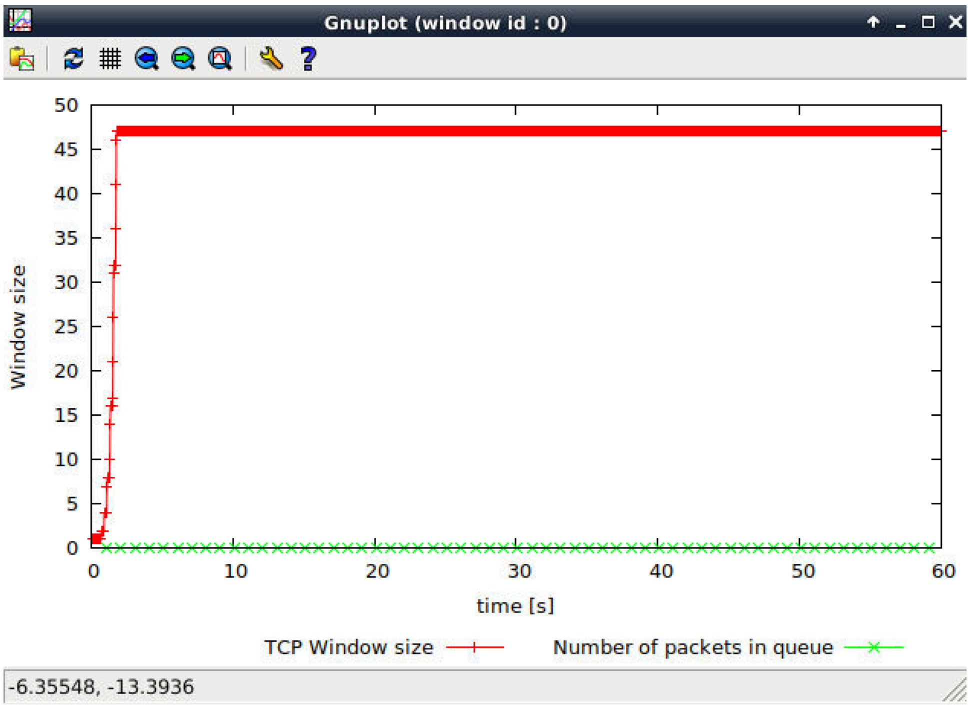
When window_size = 67



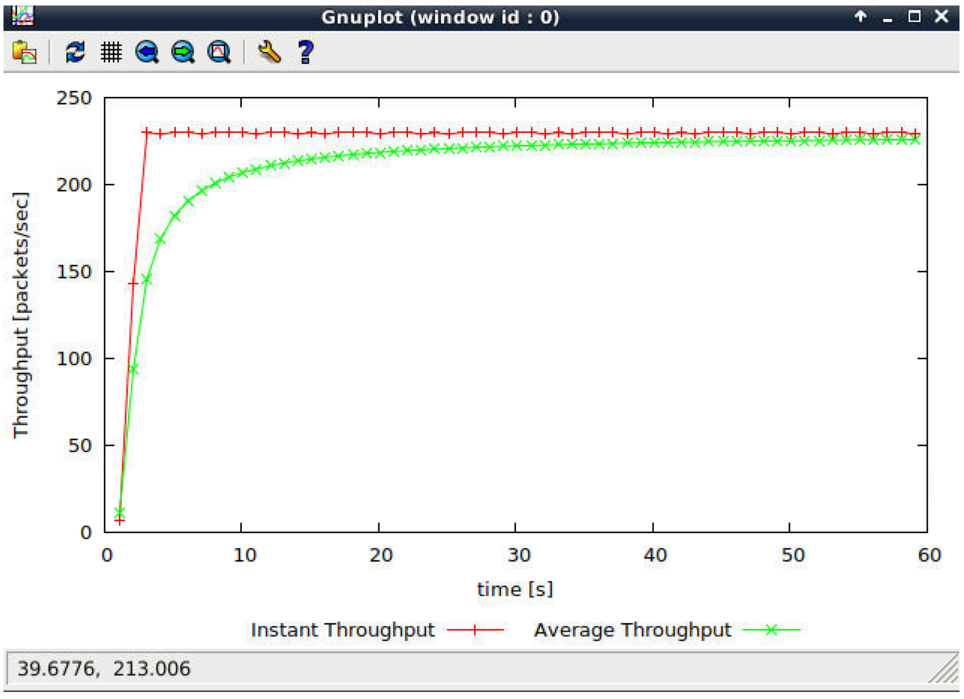
When window_size = 66, we can find that there exists no more returning to slow start anymore, which means buffer queue never becomes full. We keep reducing window_size



When window_size = 47



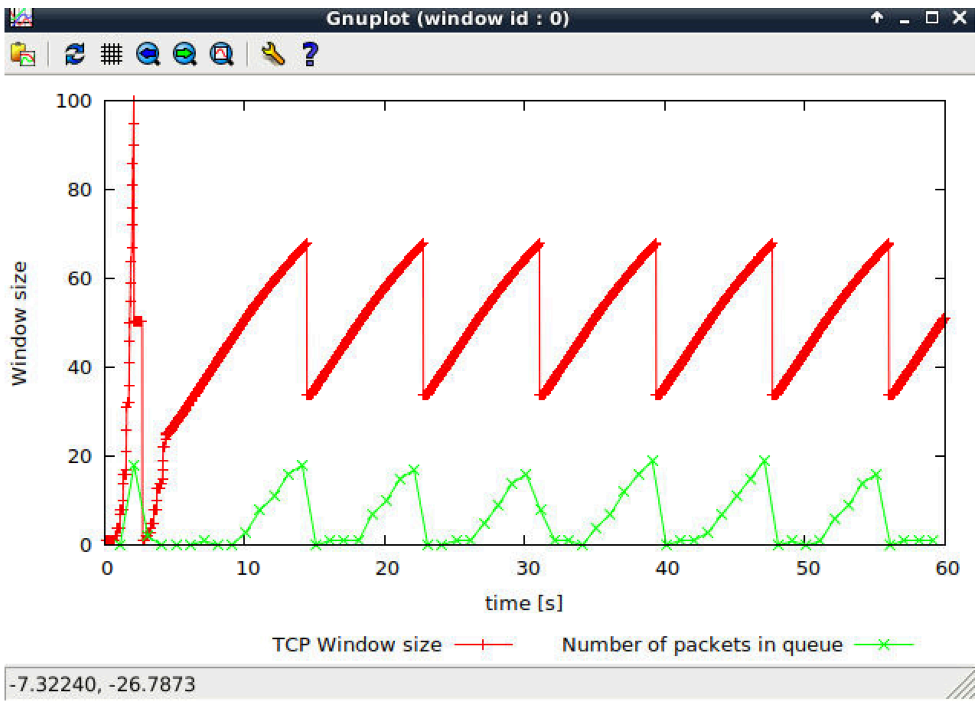
The moment file in queue stabilizes as 0, TCP tends to be stable as well, packets get maximum throughput without queuing delay.
At this point, avg_packet_throughput = 232 pps, avg_throughput = (232 * 500 * 8) = 928kbps, which is very near to 1mbps

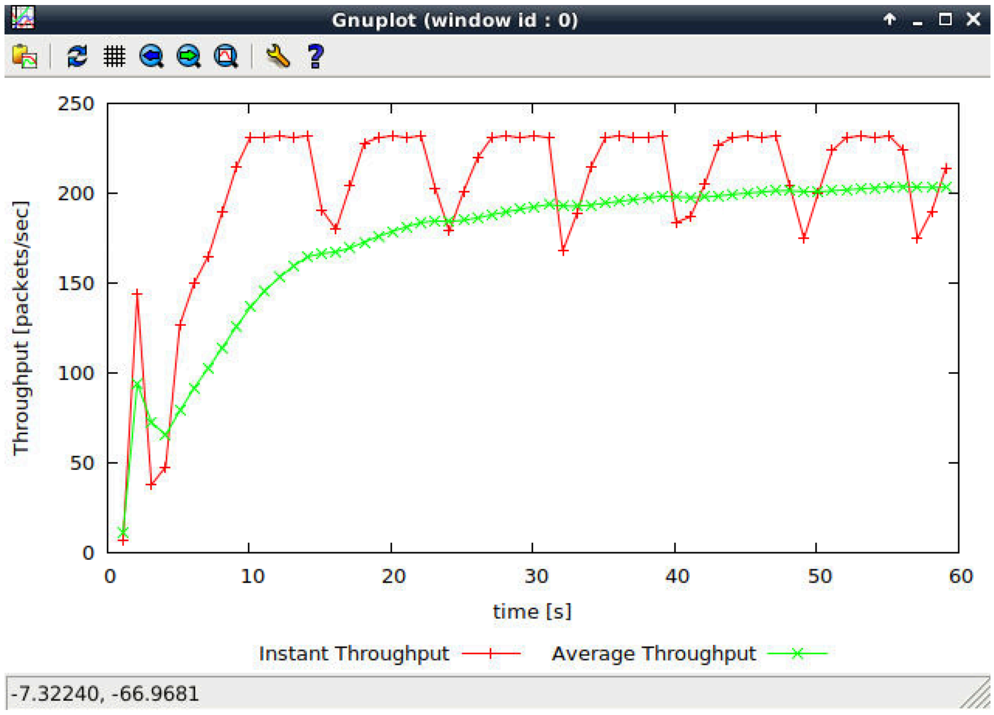


As1-4:

This question is just a comparing between TCP Tahoe and TCP Reno

With TCP Reno, it does not re-enter a slow-start phase when loss occurs





TCP Reno throughput = 204 pps

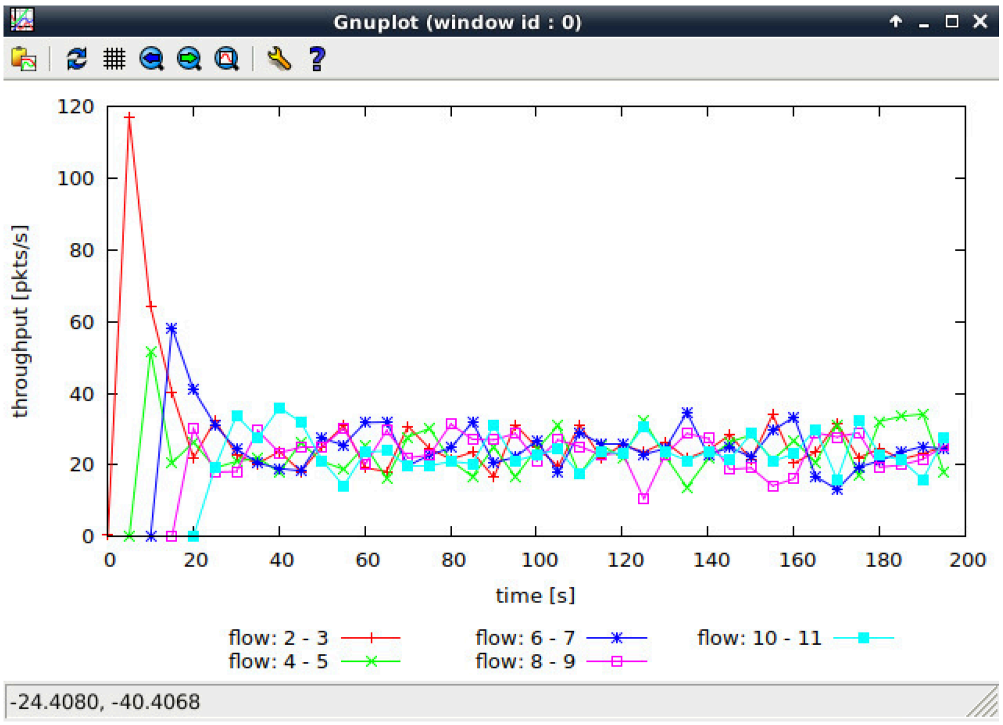
- (500 + 40) * 8 bits * 204 = 88.13kbps

Throughput previously with TCP Tahoe was 881280 bps, because TCP Reno do not need to return back to slow start phase ervery time, therefore TCP Reno performs better.

Exercise 2: Flow Fairness with TCP

Qs2-1: 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.

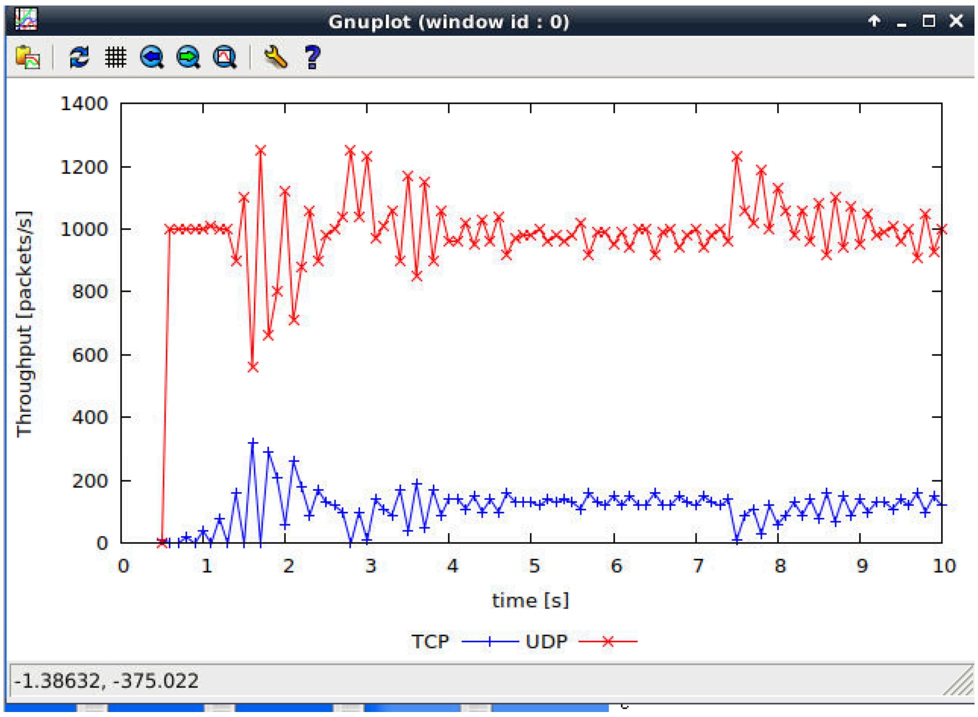
Qs2-2: What happens to the throughput of the pre-existing TCP flows when a new flow is created? Explain the mechanisms of TCP which contribute to this behaviour. Argue about whether you consider this behaviour to be fair or unfair.



As2-1. Yes, each flow get an equal share of the capacity of the common link. We can see from the graph that after 20s, although throughput for the first few connections get higher value than others, their averages are fairly similar.
As2-2.
Their throughputs significantly decreases for each new flow that is created, then they are all averaged out over time. Due to congestion control mechanisms, In first slow start, too many packages lost and congestion windows get fully ramped to ceil value. But this is a fair behavior, as each connection adjusts the size of the connection window when a new flow comes, which helps avoid congestion and make it possible of sharing of the common link.

Exercise 3: TCP competing with UDP

Qs3-1. How do you expect the TCP flow and the UDP flow to behave if the capacity of the link is 5 Mbps?
Qs3-2. 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.
Qs3-3. 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?



As3-1:
Because UDP lacks congestion control, so transmission rates can not be reduced. so UDP throughput will be higher than TCP throughput.

As3-2:
The transmission rate of UDP do not change, so UDP throughput is higher than TCP, and produce higher packet loss due to congestion..

As3-3:
Advantage: Sending speed is fast, not caring whether file has been dropped.
Disadvantage: File transfer highly relied on reliable data transfering protocol, UDP do not has this feature.
If all users applying UDP instead of TCP, the network will definately break down.