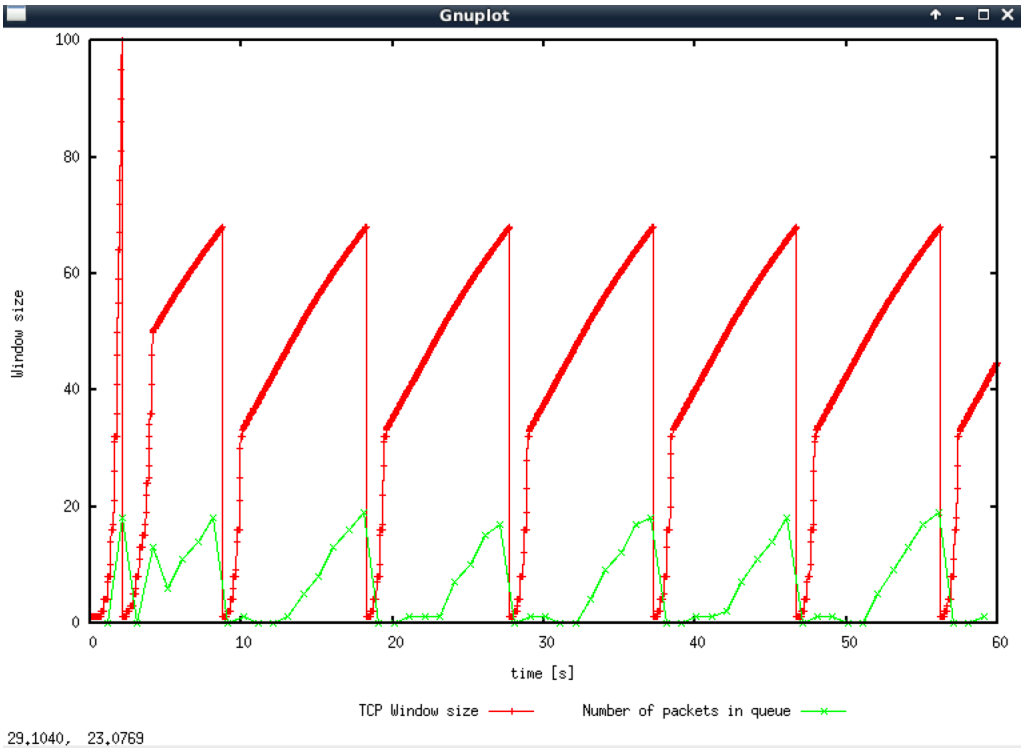
COMP9331 Lab 5

Z5185842 Luo Kaisen

Exercise 1: Understanding TCP Congestion Control using ns-2

Question 1

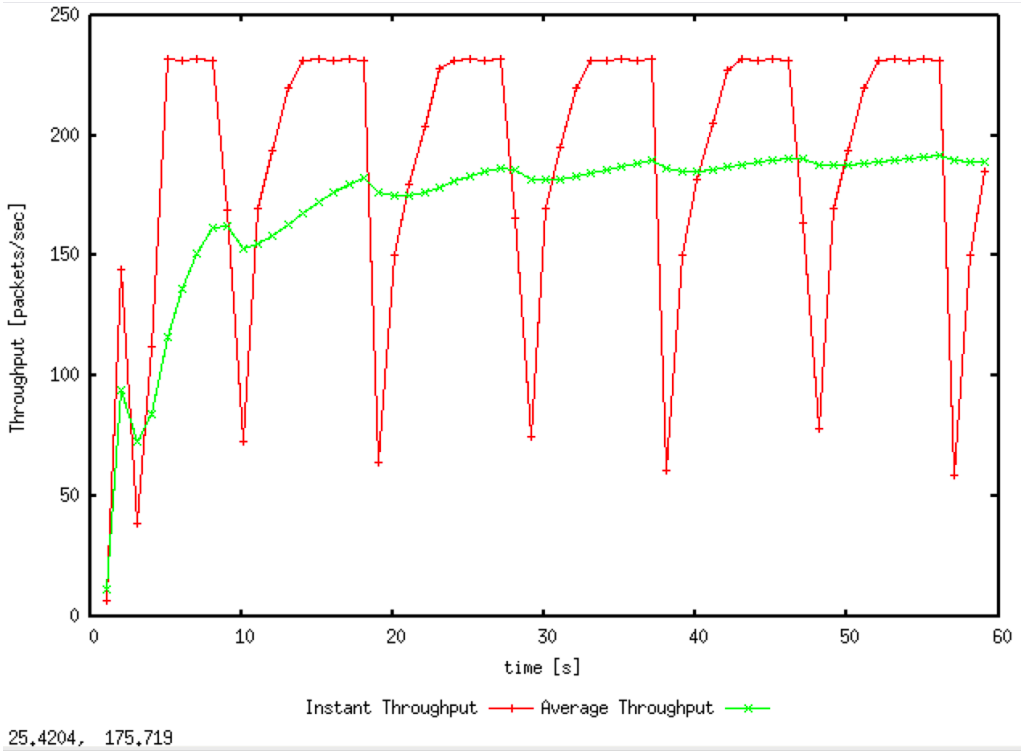
What is the maximum size of the congestion window that the TCP flow reaches in this case? What does the TCP flow do when the congestion window reaches this value? Why? What happens next? Include the graph in your submission report.



- The maximum size of congestion window is 100. Because the maximum queue size is 20, with the increase of the window size, the queue is generally getting full. When the queue has no more space for additional packets(when queue has 20 packets), the coming packets are dropped, that is why it stopped at 100 though we have 150 MSS in advance. And this lead to the congestion event at sender, then the sender sharply decreased the window size to 1 and set the threshold to 50. Next, the it turn to the slow start, the window size increase again and then the queue is full again, the connection repeat the previous process, decrease to 1 and go back to the early start phase.

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)

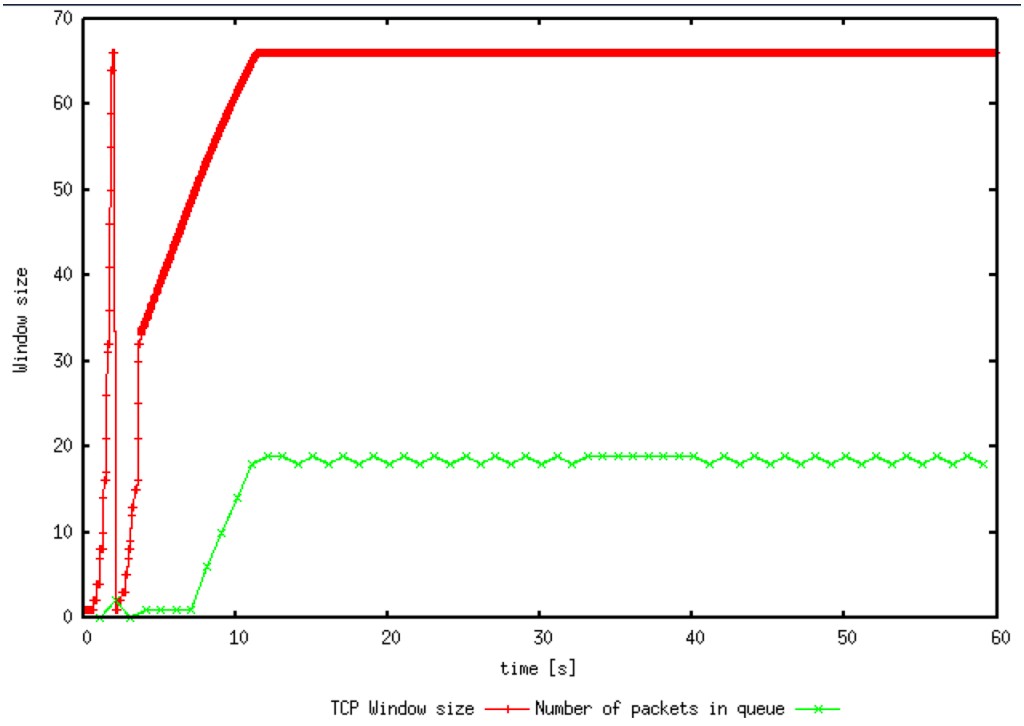


- As the graph shows, the average throughput is around 190 packets per second.
- There's two condition on bps.
  1. if we consider the payload only: $(500 + 20 + 20) \times 190 \times 8 = 802.8\ kbps$
  2. if we consider the header and the payload :$500 \times 190 \times 8 = 760\ kbps$:

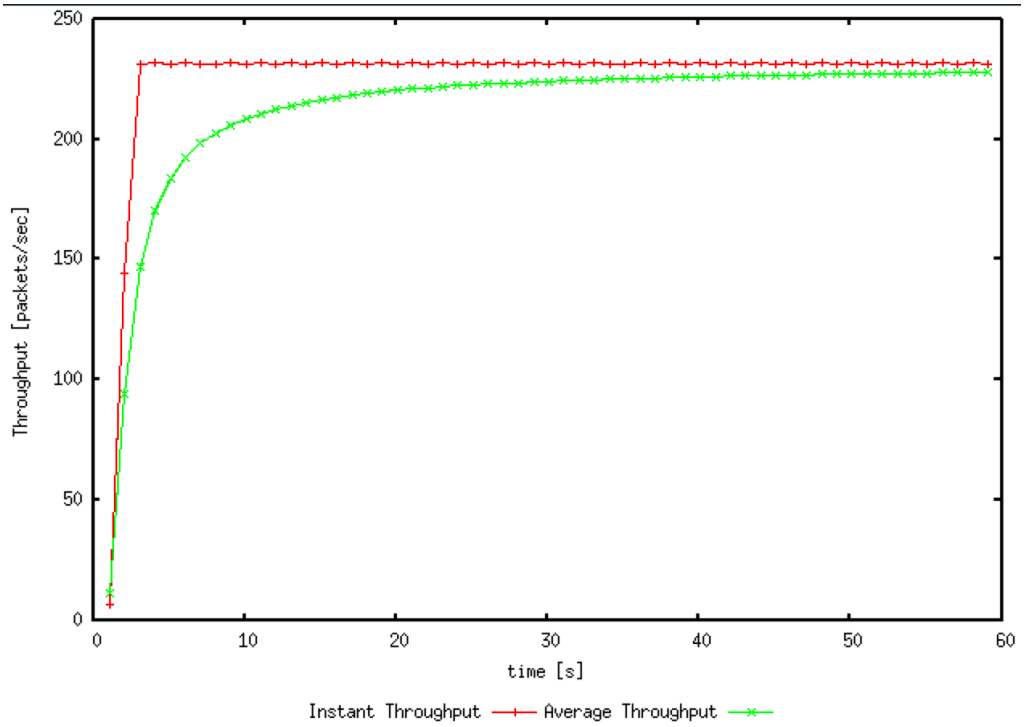
Question 3

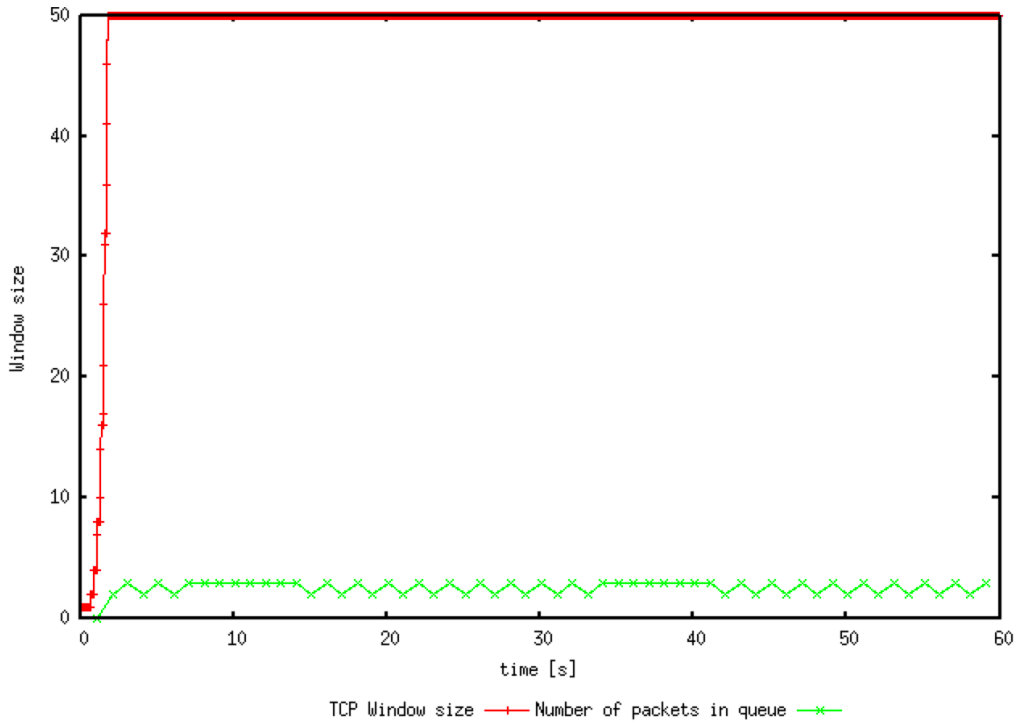
Rerun the above script, each time with different values for the max congestion window size but the same RTT (i.e. 100ms). How does TCP respond to the variation of this parameter? Find the value of the maximum congestion window at which TCP stops oscillating (i.e., does not move up and down again) to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

- With several practice on different max congestion window size, we can see when the window size exceed the maximum queue size which will lead to the congestion,then TCP will go back to the slow start phase and repeat the processes.



The maximum congestion window size that TCP stops oscillating is 50. According to the observation on the picture above and the picture in q1, when the congestion window size is 66, after the first decreased and the half the window size, the congetion window size is enough to hold the packets in sending queue, the packets number remain stable and the congestion won't occur beacuse the queue never full and there won't have any packet loss.

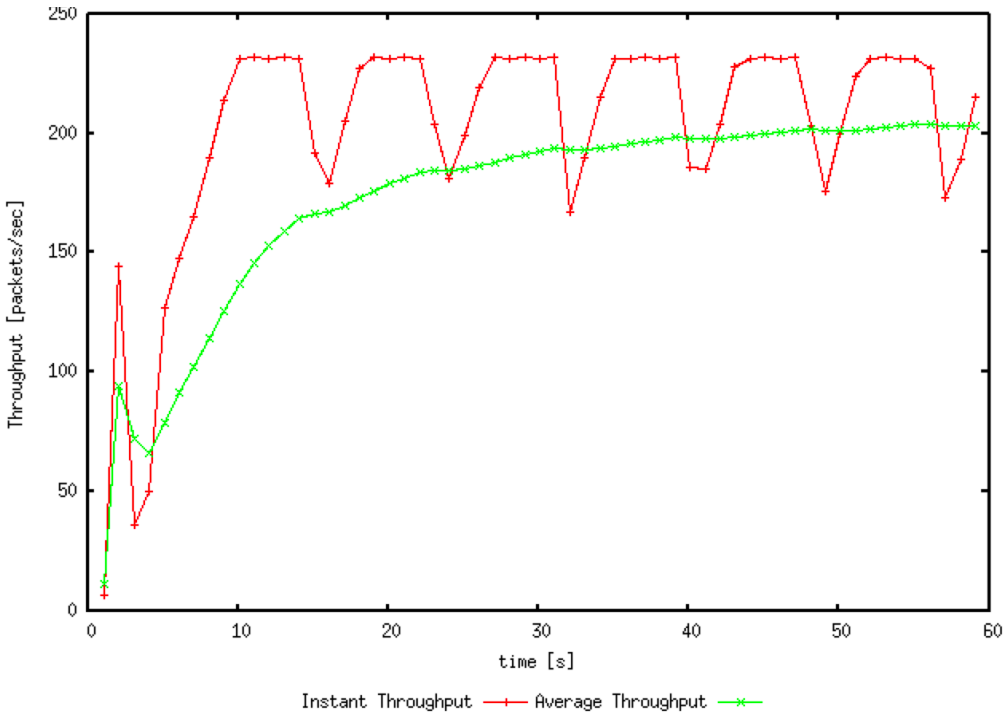




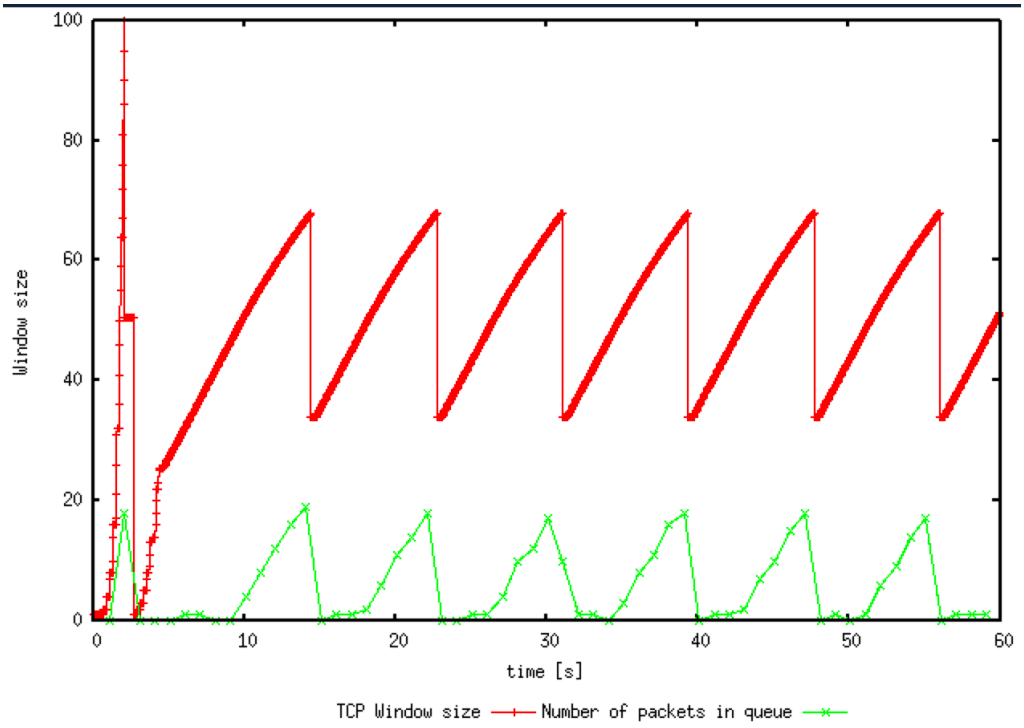
According to the pictures above, the maximum congestion window size is 50 which ensure that there won't have congestion and packet loss. At this condition, the average packet throughput is around 225 packets per second, and the average throughput is $225 \times 500 \times 8 = 900$ $Kbps$ and this is very close to the link capacity(1 Mbps).

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?



- This picture shows the average and instant throughput of the TCP. The average throughput of TCP Reno is about 200 packet/second which is higher than TCP Tahoe. This is mainly beacuse Reno dosen't initiate slow start after each congestion but Tahoe do.

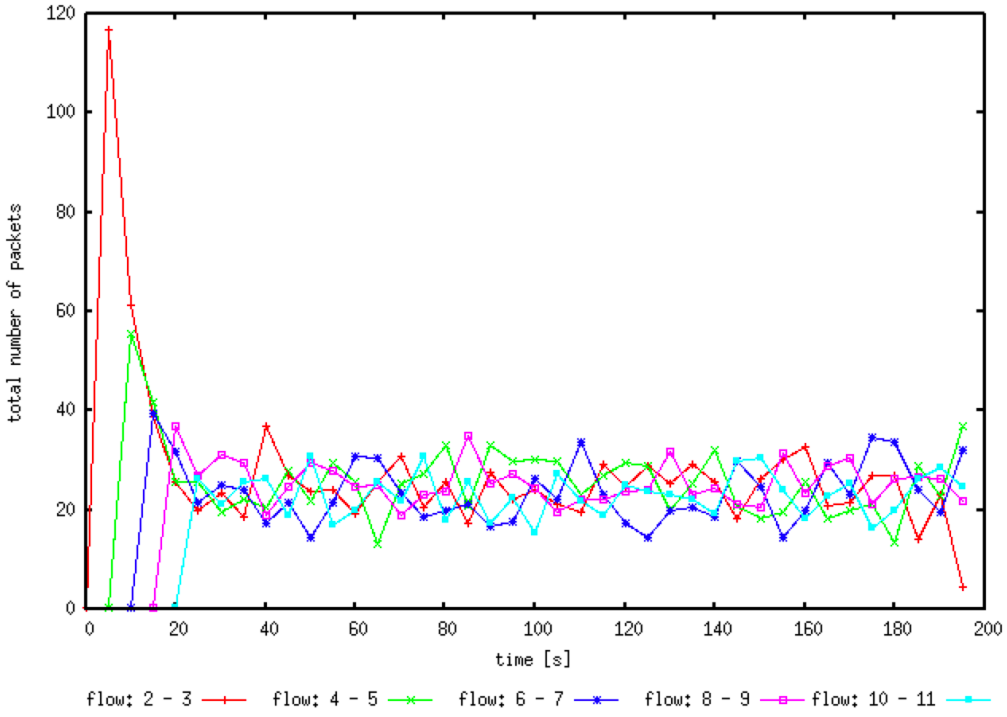


- This graph shows the relation between window size, queue and time. Different to Tahoe, Reno half the window size rather than reduce the window size to 1 when it meet a loss. After that, TCP Reno increase stablely, and repeat the process when the loss happen again.

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.



- From the graph, it shows that the five flow shared almost the equal capacity of the link. This is beacuse that this flow is controled based on the AIMD algorithm which ensures a dynamic balance. Since these flow are under same network condition and react the same.

Question 2

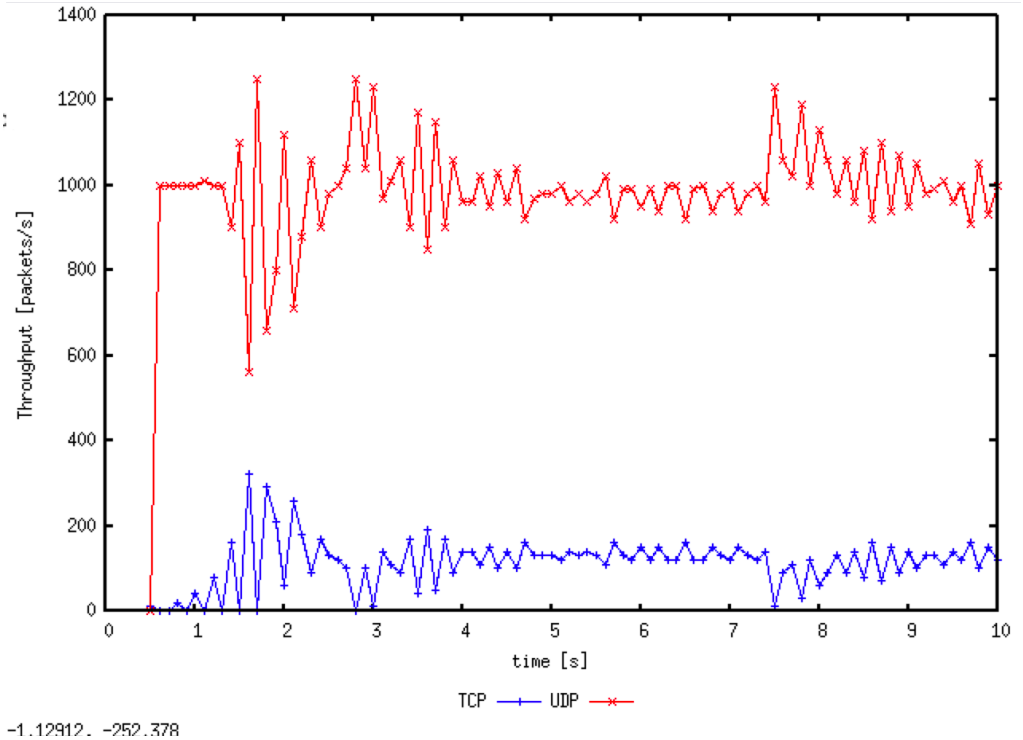
What happens to the throughput of the pre-existing TCP flows when a new flow is created? Explain the mechanisms of TCP which contribute to this behaviour. Argue about whether you consider this behaviour to be fair or unfair.

- From the graph above, we can see that when comes a new flow, the link resource for each existing flow will reduce which is mainly beacause the new flow's slow start phase will will lead the congestion. According to this, the existing TCP connections detect losses and through ACKs and timeout, adapt the size of their congestion to relief the load of the network. After that, due to the increase of the number of the flow, the average window size for each flow will also decreased in order to ensure the fairness of TCP.

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 ?



- As the UDP will transmit without any congestion control, it will behave much faster than TCP(as the picture shows). So the UDP will continue transmit at it's rate while the TCP will consider the congetion and capacity of the network.

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.

- As UDP doesn't employ any congestion control, so it won't change the transmit rate no matter the condition of the transmission, it will keep a roughly stable transmit rate. And UDP won't care about the loss which is quite different from TCP. When TCP meet a the UDP flows, it will considerly slow the rate while UDP won't. So TCP will perform much slower(lower throughput) in this condition.

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:
  UDP is mucher faster as it does not constrained by the congestion control, it will remain a high rate to transmit the file. This will lead to the low delay and faster transmission, maybe save time.
- Disadvantages:
  UDP is a unreliable protocal. It won't care about the loss which may lead to the incompletement of the file which need reliable data transfer to avoid this.
- If everyone use UDP instead of TCP, the network is very likely to face the heavy congestion as no one will reduce the speed when they transfer the file simultaneously and even lead to the collapse of the network.