Lab 5

Exercise 1

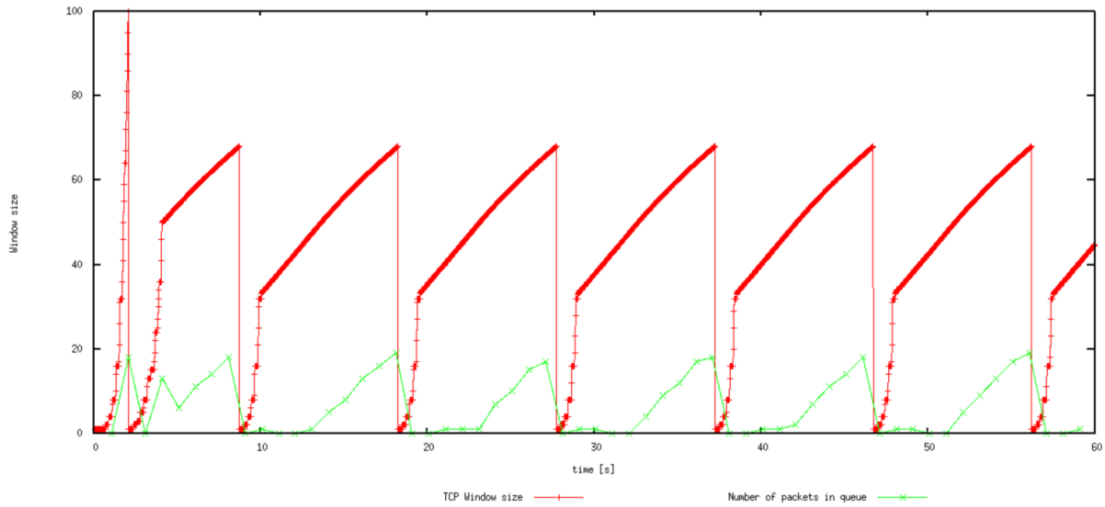
Question 1

What is the maximum size of the congestion window? What does the TCP flow do at this time? Why? What happens next?

The maximum windows size is **100** MSS.

The TCP flow is then experiencing package drop due to triple dup ACKs and timeout at this time. Then the flow will reduce the congestion window to 1 MSS (initial window size) and then set the slow start threshold to half of CWND (50). Because Reno algorithm will terminate slow start and AIMD phases when loss is detected (triple dup ACK and timeout) and restart to slow start.

Then, a new slow start phase will begin with new CWND and ssthresh. After CWND reaches ssthresh, it will enter AIMD phase until loss detected.

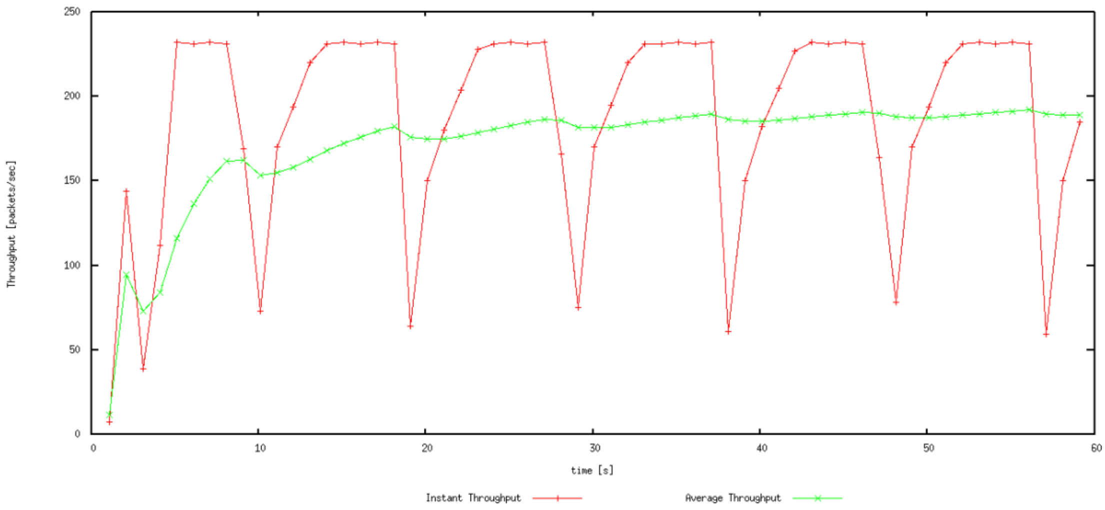


Question 2

What is the average throughput of TCP in this case?

According to *WindowMon.tr*, the average throughput is around **188.976** packets/sec. Since the sizes of IP and TCP headers are both 20 bytes, the size of each packet is $500 + 20 + 20 = 540\ bytes$.

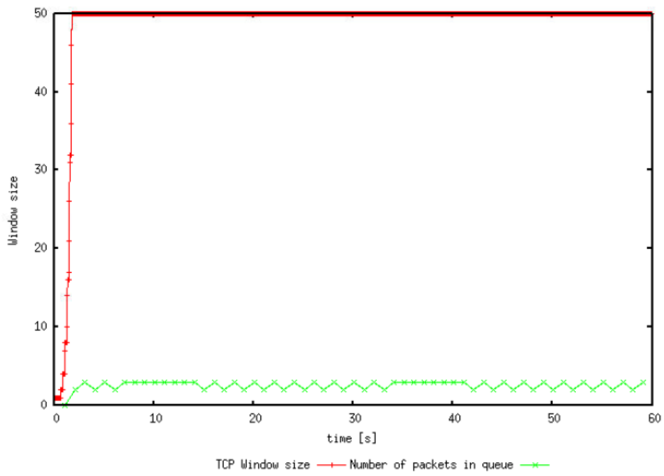
The average throughput is then $540 * 188.976 = 102047.04\ bytes/\sec = 816376.32\ bps$

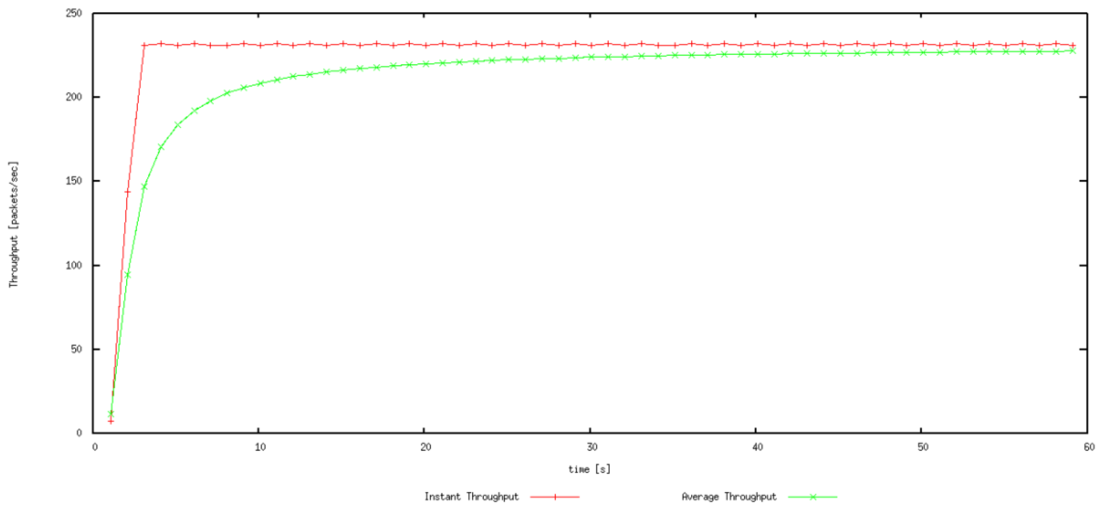


Question 3

Find the value of the maximum congestion window at which TCP stops oscillating to reach a stable behaviour. What is the average throughput, how does the actual average throughput compare to the link capacity (1Mbps)?

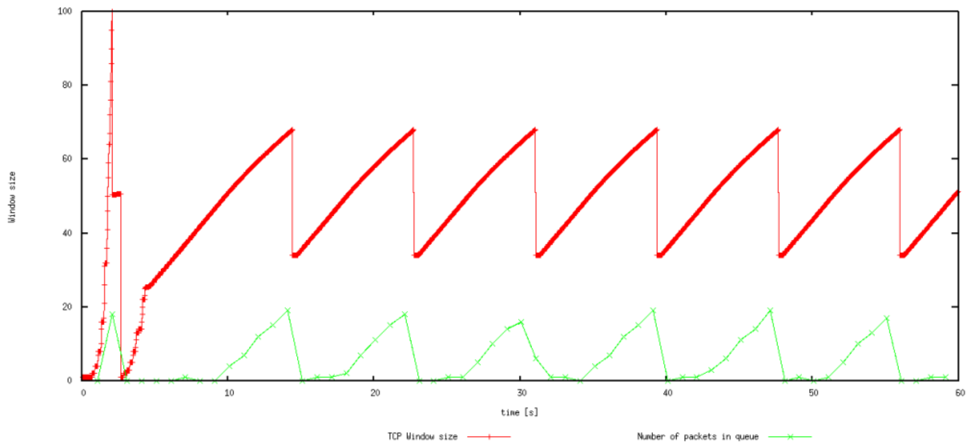
The max value is **50**. The average throughput is **227.73** packet/sec, and it will ultimately reach nearly 231.5 because the instant throughput is keeping at that level. The theoretical average throughput derived from link capacity is $1000{,}000 \div 8 \div 540 \approx 231.48\ packet/sec$ which is close to the actual value.



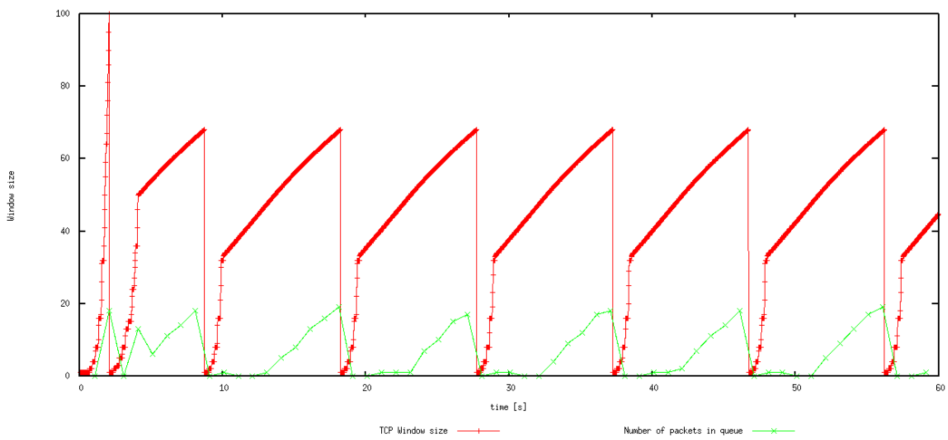


Question 4

Compare the graphs for the two implementations and explain the differences. How does the average throughput differ in both implementations?



Window plot of Reno

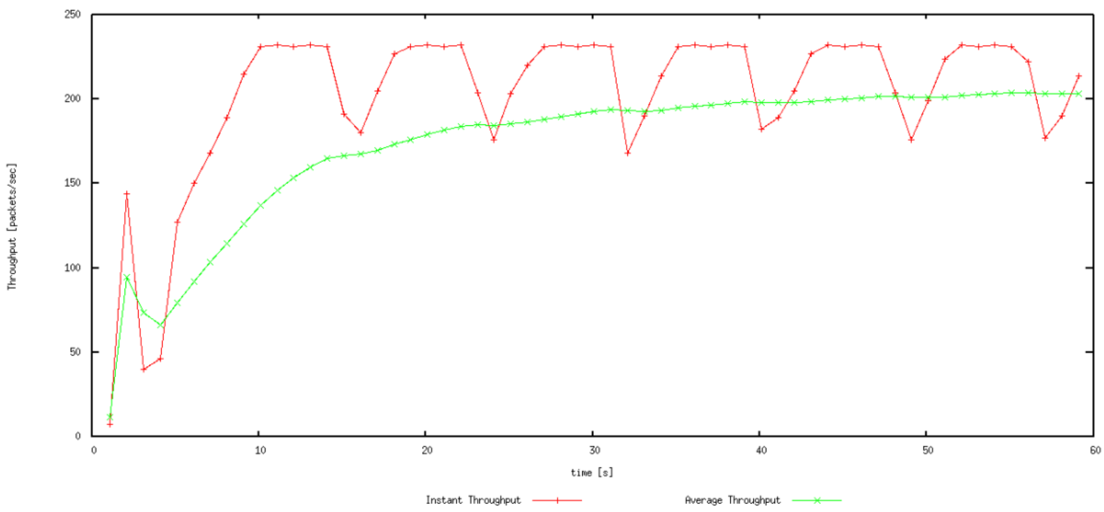


Window plot of Tahoe

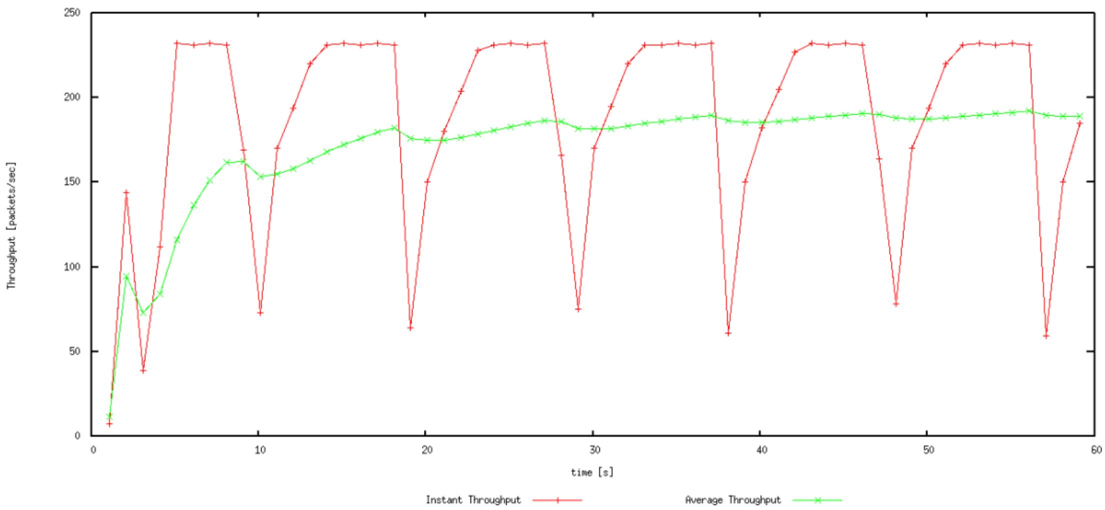
The first difference is that Tahoe will set CWND to initial state if loss detected while Reno only halves the value. In the graph of Reno, CWND only recover to the initial state once but that happens each time for Tahoe in in each congestion avoidance phase. This is because they differ in the loss handling algorithm. Reno will first enter a fast recovery phase by halving CWND upon receiving triple dup ACKs and only reset to initial state when further loss is detected.

The Instant throughput of Reno is relatively more stable comparing with that of Tahoe due to the difference in congestion recover mechanism.

The average throughput of Reno is about **203.447** while the average of Tahoe is only **188.976** packets per second.



Throughput of Reno

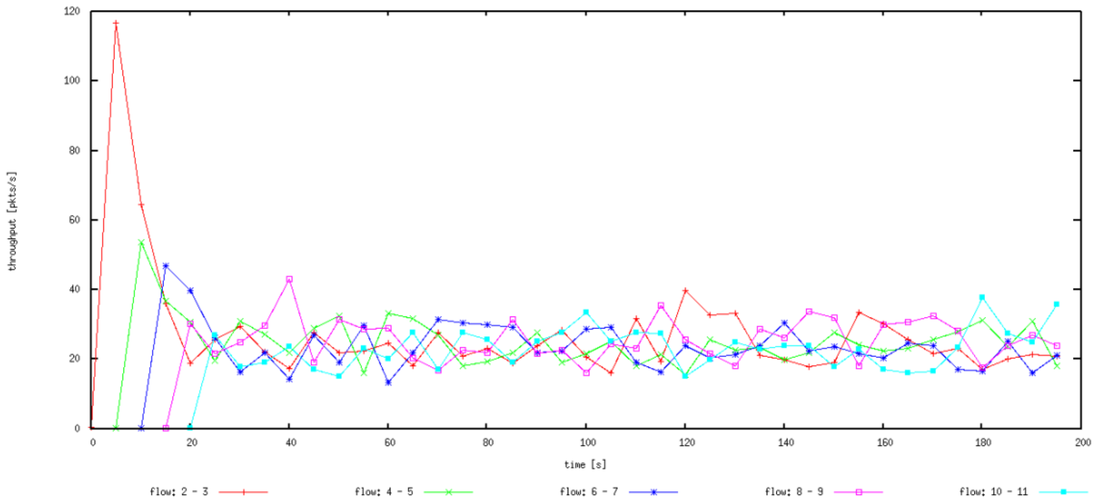


Throughput of Tahoe

Exercise 2

Question 1

Does each flow get an equal share of the capacity of the common link?



Although the throughput graph shows that they don't share the resource equally, they can still be regarded sharing approximately equal share of the capacity. Because each time a new flow join, all participating flows will start to oscillate at the new level around the equal share. And finally, all shares oscillate around the same level irrespective to the sequence of joining.

Question 2

What happens to the throughput of the pre-existing TCP flows when a new flow is created? Explain the mechanism and argue.

All flows will rebalance the shares when a new flow is created so that they can share the capacity of the link equally. That may cause the throughput of pre-existing TCP flows to decrease together by same proportion. This is achieved by the congestion avoidance mechanism of TCP protocol. A new flow joining the new work can generate a lot of traffic and may overload the link. And the AIMD style applied by each node ensure that they can dynamically rebalance the rate of sending. Flows with high throughput decreases rapidly and the nodes with low throughput increase steadily. Finally, they can reach a fair equilibrium.

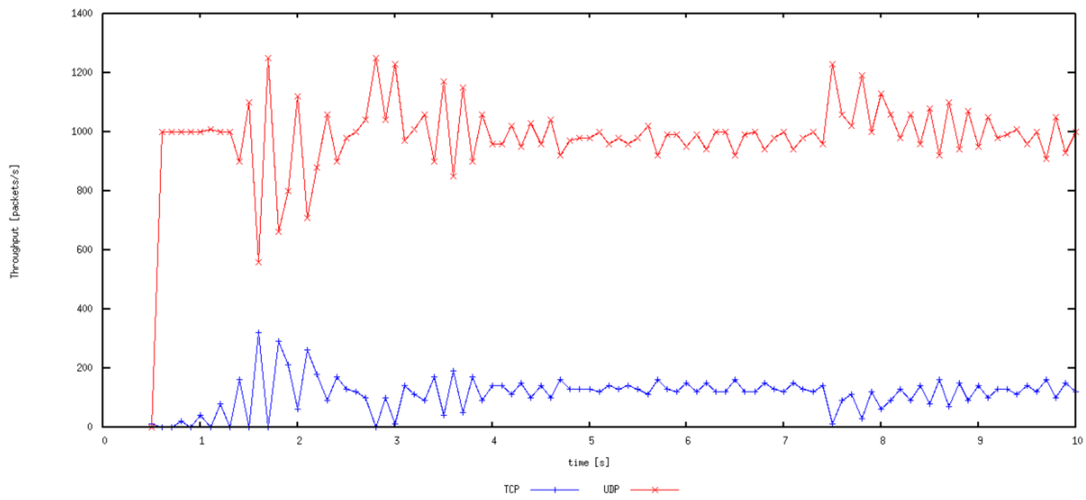
Exercise 3

Question 1

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

UPD flow is expected to occupy most part of the channel because UDP doesn't have a congestion control mechanism. As the traffic generator creates new data at a rate of 4Mbps, the UDP flow may occupy 80% (4/5) of the whole throughput.

As shown in the diagram below, the throughput of UDP flow is about four times of that of TCP flow.



Throughput with link capacity of 5Mbps

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.

It is because UDP doesn't implement a congestion control mechanism, when loss is detected, TCP will decrease it's sent rate but UDP will not. As the link capacity is not enough for the two flows, UDP will try to send at full rate and only leave the remaining space to TCP. Because UDP is always send at full rate, TCP may try to restrict its sending rate according to the network traffic and not to compete with UDP.

Question 3

Advantages of UDP are:

Constant sending rate irrespective to network condition,

smaller packet size,

potentially higher throughput and lower delay.

Disadvantages:

No congestion control, may cause serious congestion,

no guarantee on reliable data transfer,

the packages may arrive out of order.

If everyone starts using UDP instead of TCP, the network may be totally congested. Because everyone is managing to send at highest rate. If the total required throughput is much higher than the capacity, most packages will be lost.