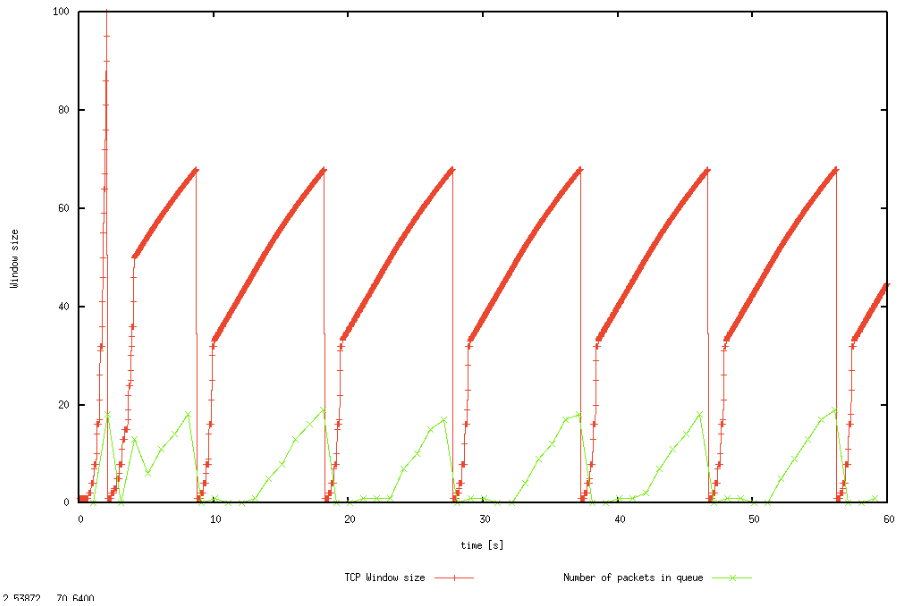
z5147986

Lab5

1.1

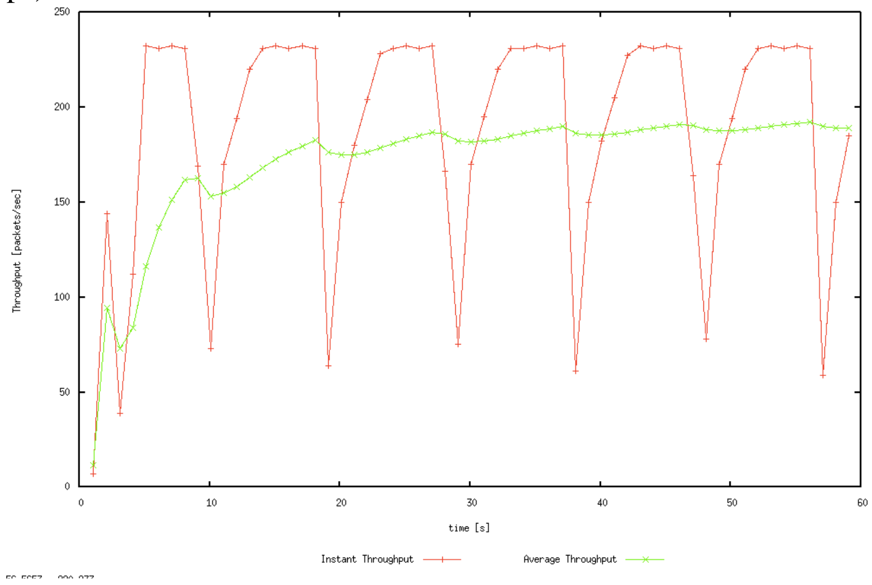
What is the maximum size of the congestion window that the TCP flow reaches in this case? What does the TCP flow do when the congestion window reaches this value? Why? What happens next?

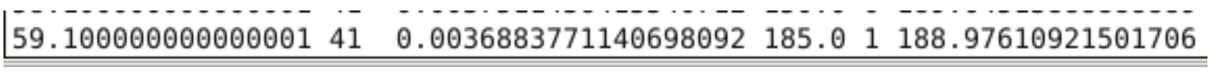


The max window size is 100. After timeout or triple dup ACK, the congestion window size dropped to 1 and the ssthresh is set to half of 100, 50. Slow start phase will start again with window size 1.

1.2

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)





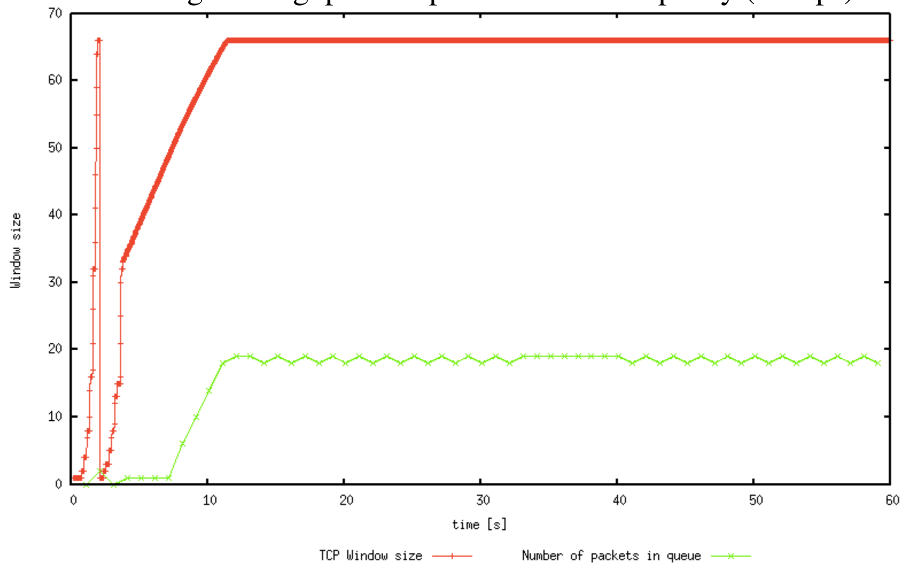
```
59.100000000000001 41  0.0036883771140698092 185.0 1 188.97610921501706
```

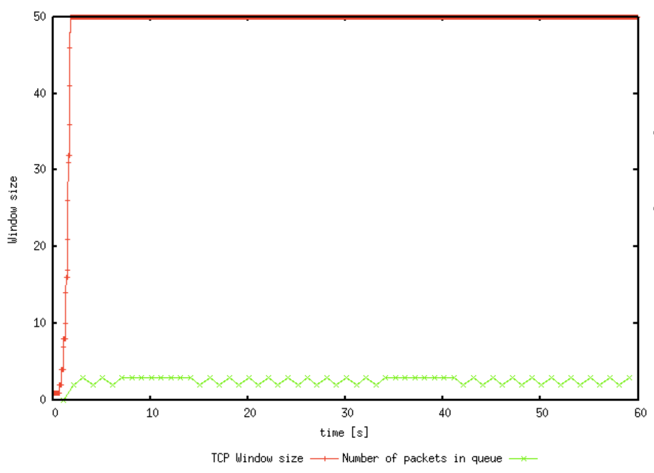
The average throughput is 188.9761092150176 packets per second *from WindowMon.tr

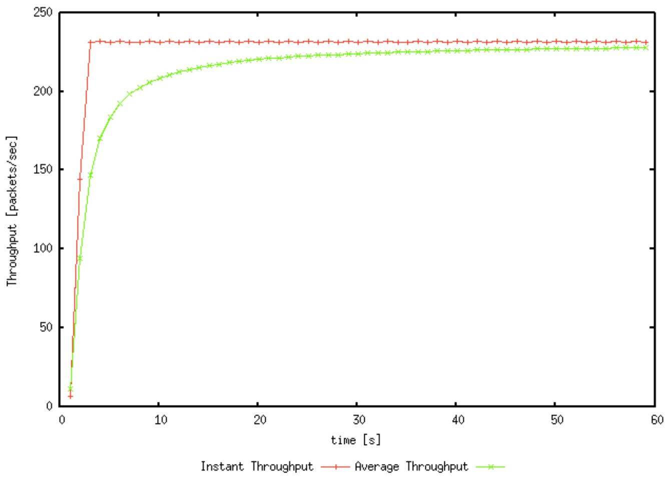
Average throughput = (500 + 20 + 20) * 188.9761092150176 = 102047.098976 bytes per second

z5147986
Lab5

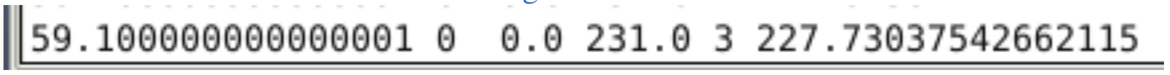
1.3
How does TCP respond to the variation of this parameter? Find the value of the maximum congestion window at which TCP stops oscillating (i.e., does not move up and down again) to reach a stable behaviour. What is the average throughput (in packets and bps) at this point? How does the actual average throughput compare to the link capacity (1Mbps)?





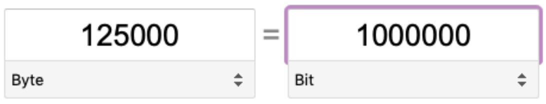


As the window size decreases, the number of oscillating decreases. Once the max window size is down till 66, then there is only one oscillating and if the max window size is lower than 51, there will not be no oscillating.
The max window size to avoid oscillating is 50



```
59.100000000000001 0  0.0 231.0 3 227.73037542662115
```

The average throughput is 227.73037542662115 packets per second *from WindowMon.tr
Average throughput = (500 + 20 + 20) * 227.73037542662115 = 122974.40273 bytes per second

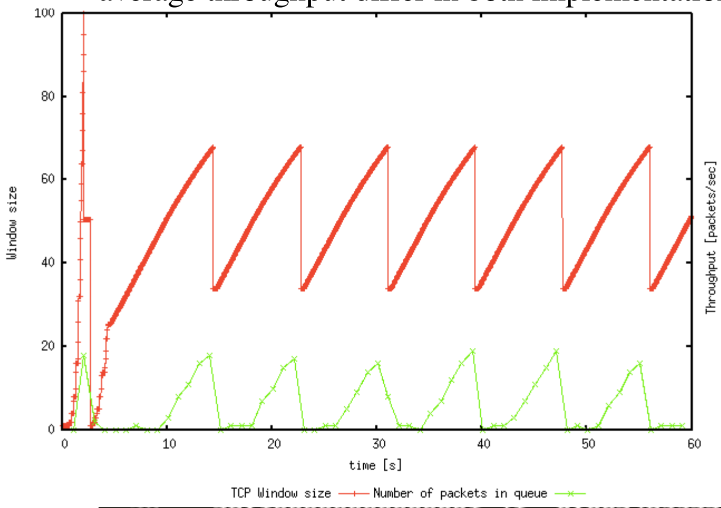


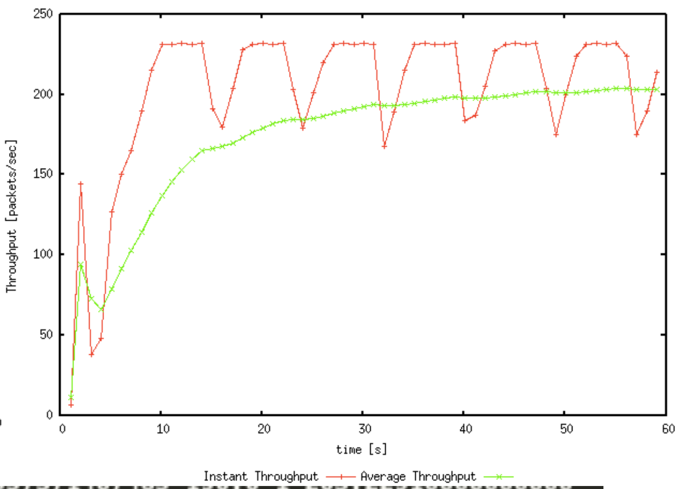
125000 = 1000000
Byte Bit

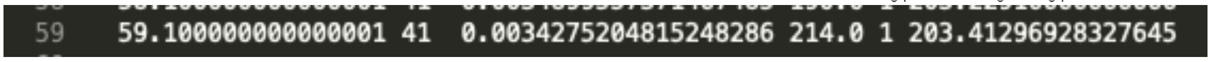
Link capacity = 100 * (122974.40273 / 125000) = 98.38%

z5147986
Lab5

1.4
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?





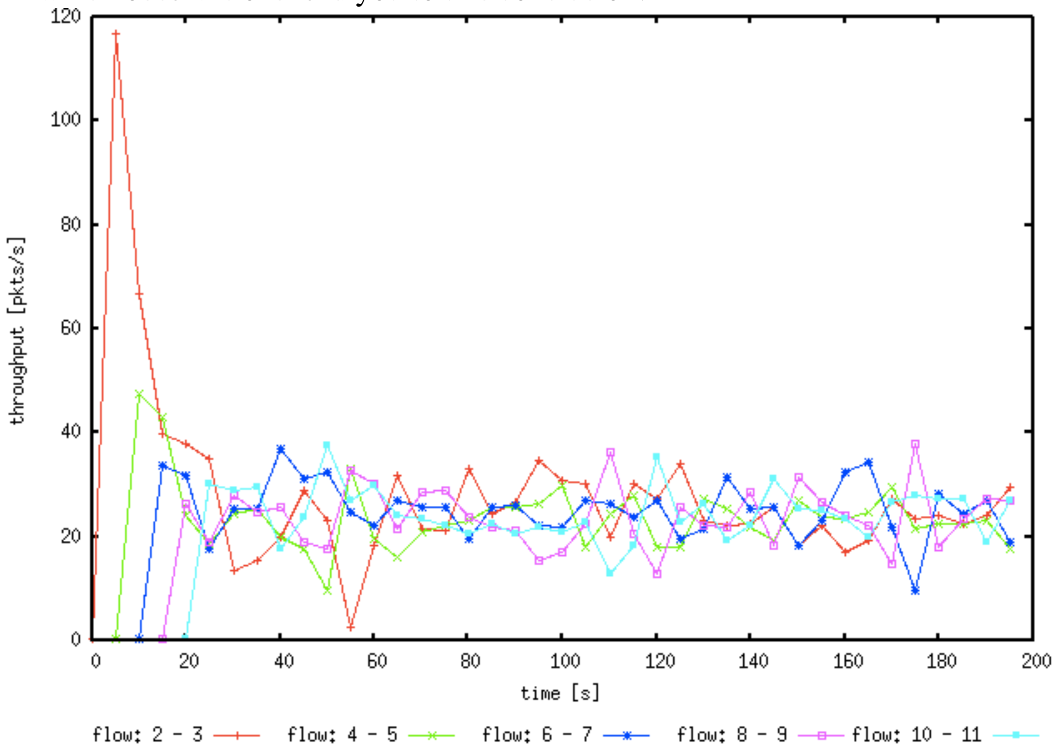


```
59    59.100000000000001 41  0.0034275204815248286 214.0 1 203.41296928327645
```

The window size only hit zero once after slow start phase for Reno. The average throughput is higher for Reno (203.41296928327645) than Tahoe (188.9761092150176).

z5147986
Lab5

2.1
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.

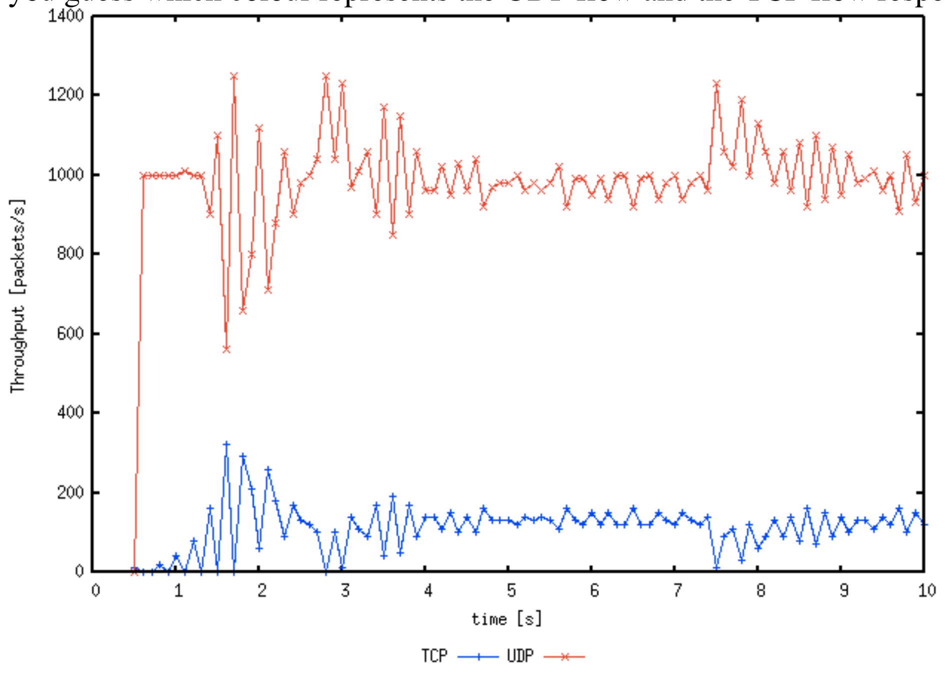


Yes, each flow get an equal share of the capacity of the common link. Although at the start each flow has different share of common link, as time increases, each flow increase and decrease till each flow average within 20 – 40 packets per second.

2.2
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.
Throughput of pre-existing TCP flows will decrease when a new flow is created to provide equal share. The congestion window size increases rapidly during slow start phase which causes congestion. Thus, all flows will adjust to adapt the network. The behaviour is fair.

z5147986
Lab5

3.1
How do you expect the TCP flow and the UDP flow to behave if the capacity of the link is 5 Mbps ? 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 ?



TCP have congestion control, UDP do not. Thus, TCP will have oscillating while UDP can use can utilise the capacity which shows that UDP will have a higher throughput compared to TCP. UDP is red, TCP is blue.

3.2
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 do not have congestion control mechanism, it provides best-effort datagram where application provide own reliability and flow control.
TCP have congestion control mechanism, it provides stable connection and not overloading the link by changing the window size based on conditions.

3.3
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 using UDP over TCP
- Higher average throughput
- Transfer rate based off link bandwidth
- Smaller packet size

Disadvantages using UDP over TCP
- No congestion control
- Packets arrive out of order
- Unaware of package loss and corrupted package

If everyone started using UDP over TCP, tons of package loss, performance suffer and network congested, hard to detect corrupted packages