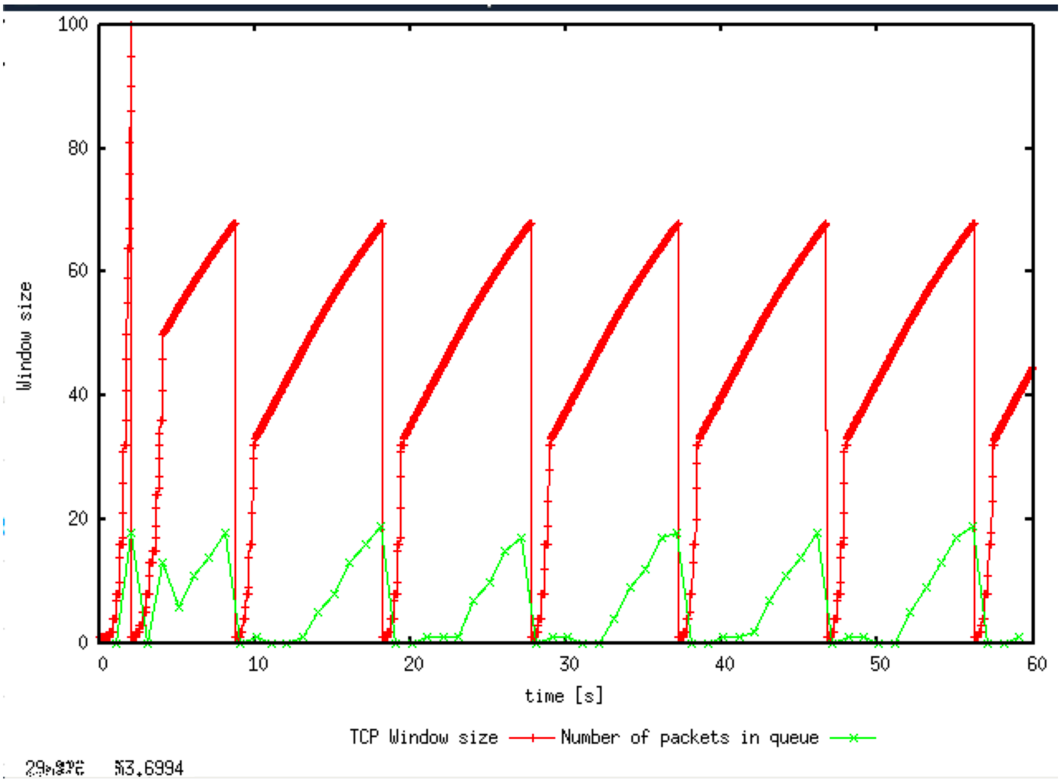
COMP3331 LAB05 REPORT

XIAO XIAOSHAN z5191546

Exercise1:

Question1:

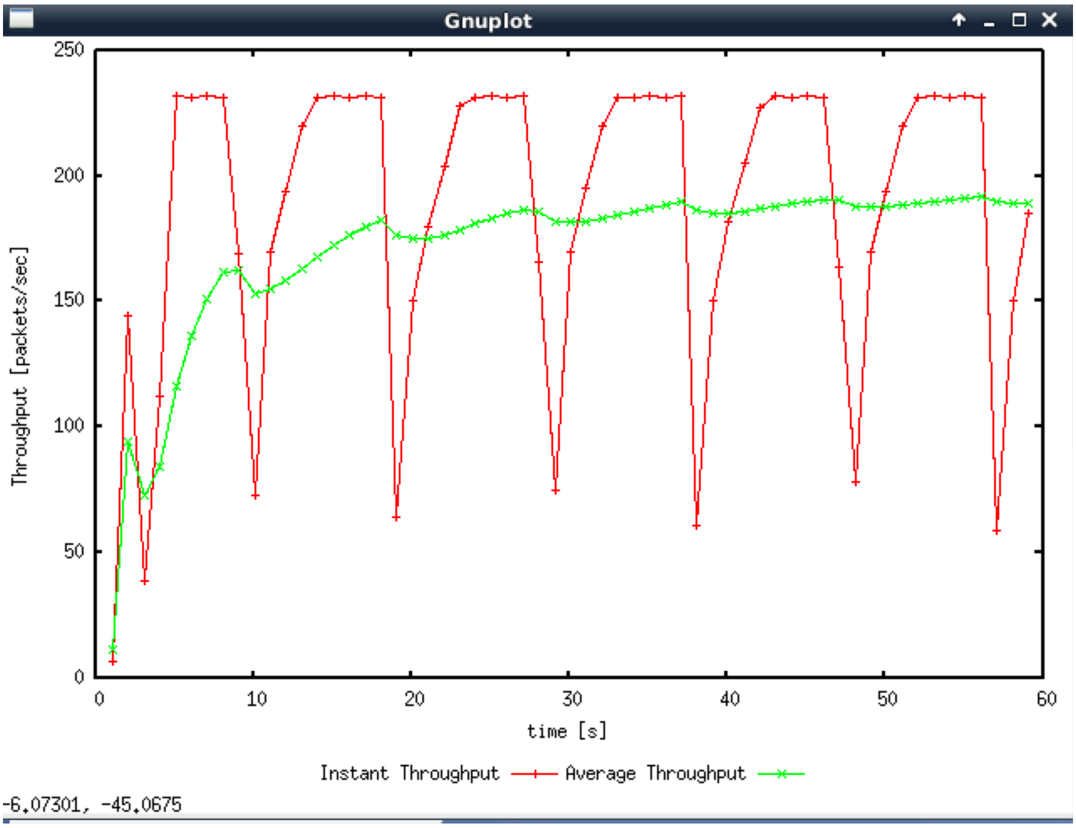


1) The maximum size of the congestion window that the TCP flow reaches in this case is 100 packets.

2) When congestion window reaches this size, it reset the window size, this may because it received 3 duplicate ACK's or a timeout.

3) It will back into the slow-start phase, and the window size will increase exponentially up to around 50 (half of the maximum size), at window size 50, it linearly increases until facing loss.

Question2:




```
58.100000000000001 41  0.0037511436413540712 150.0 0 189.045138888889
59.100000000000001 41  0.0036883771140698092 185.0 1 188.97610921501706
~
```

Throughput =

Packets per second throughput = ~190pps

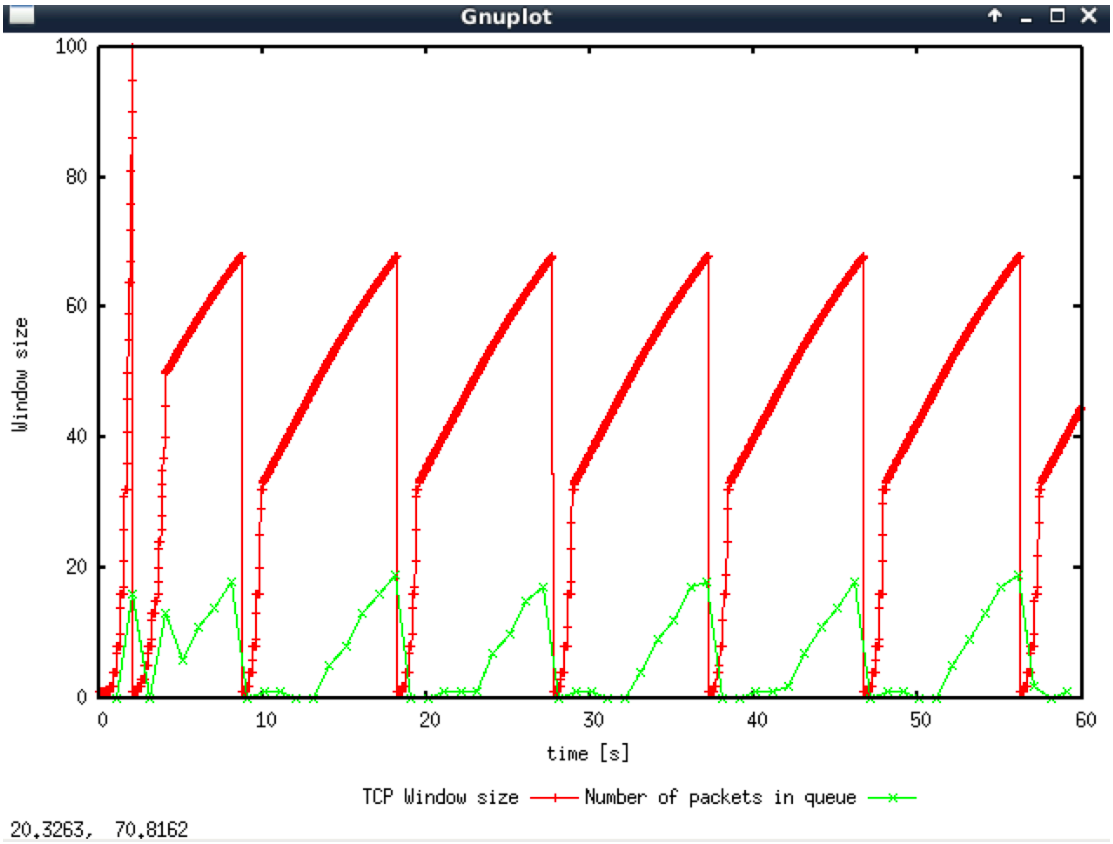
Bytes per second throughout:

IP + TCP Headers = 20 + 20 = 40 bytes

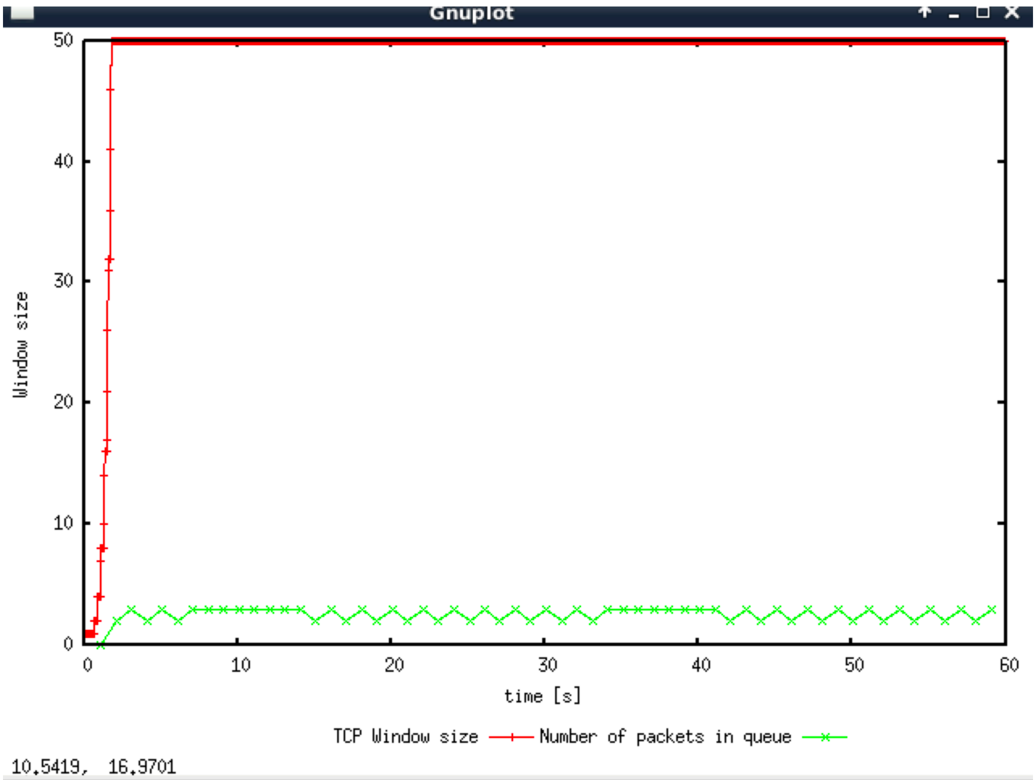
Payload = 500bytes

= (500+40) * 8 * 190 = 820,880 bps throughput

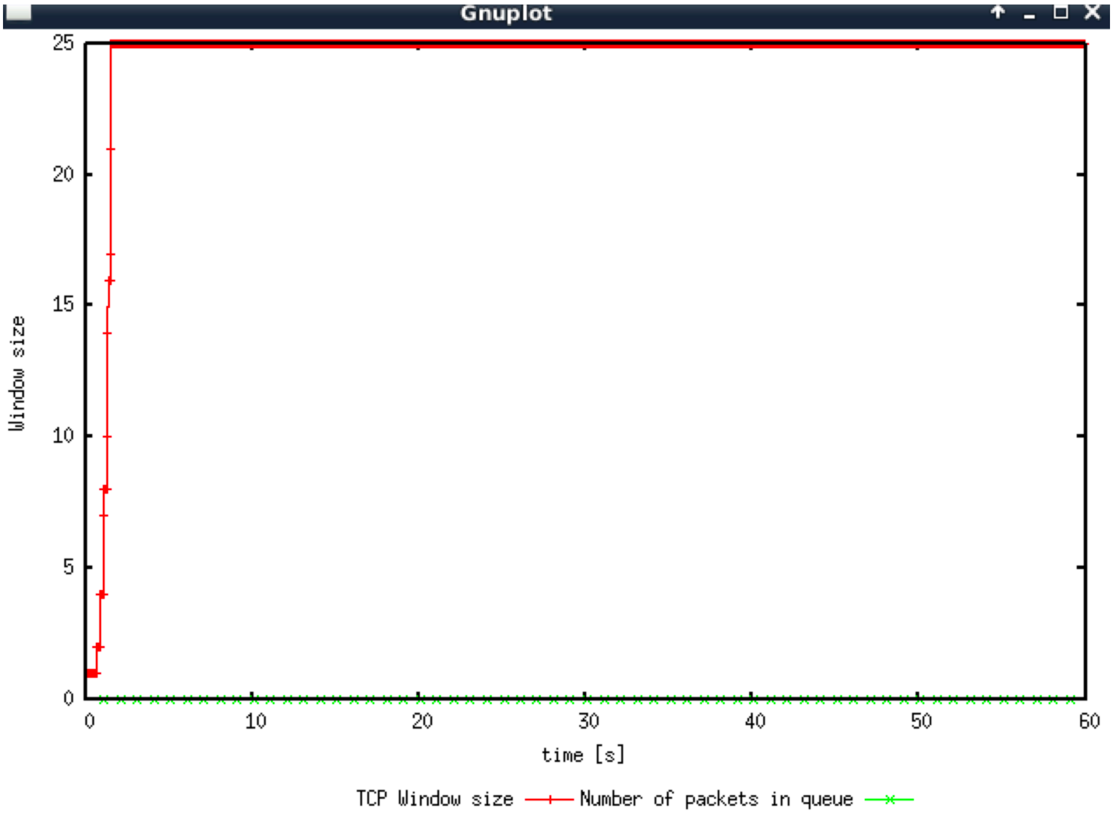
Question3:



(max size = 100)

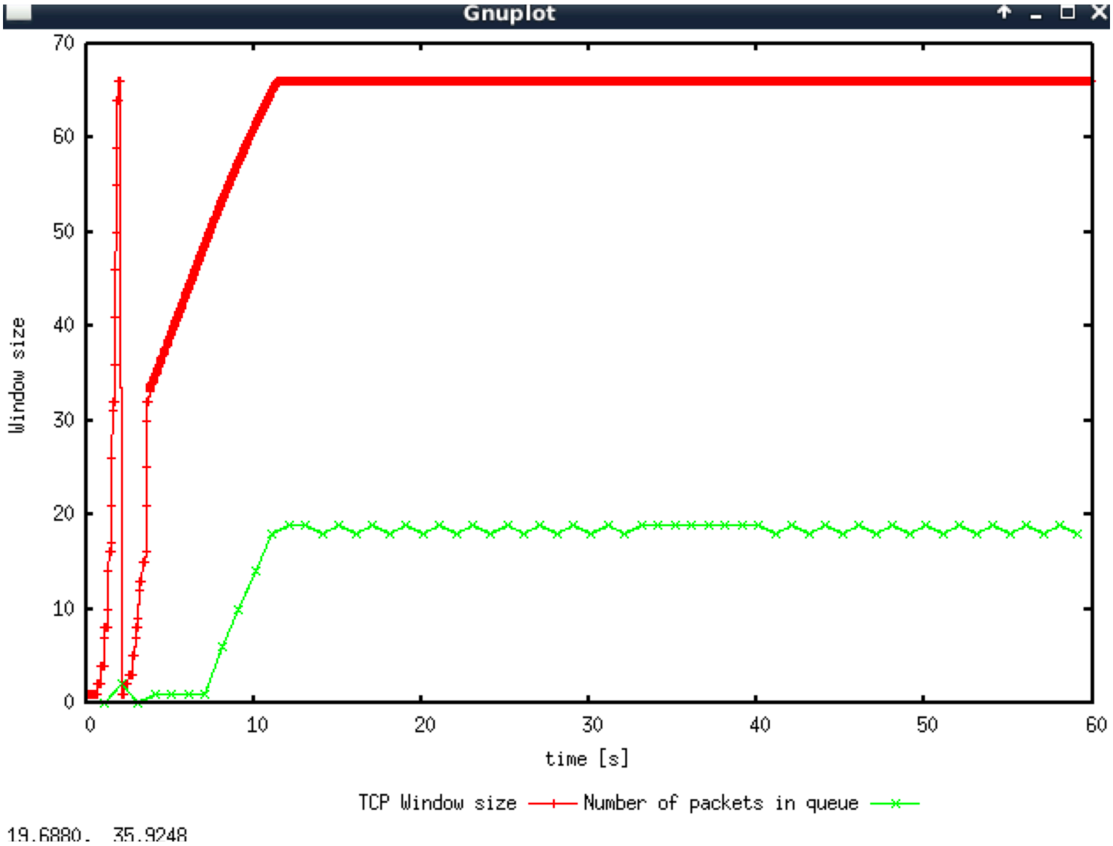


(max size = 50)

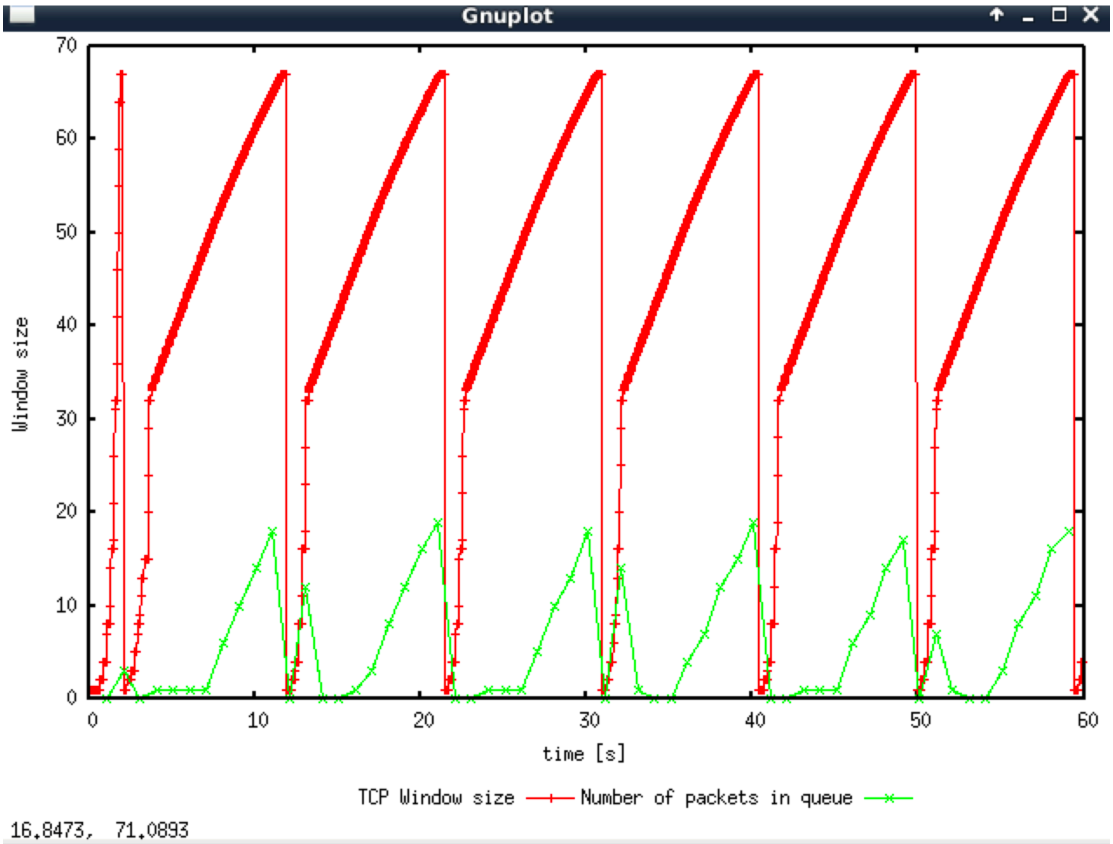


(max size = 25)

1) TCP will increase to reach that the value of that parameter, and also make sure the window size would not go over that value.
2) The value of the maximum congestion window at which TCP stops oscillating is 66, the initial drop is due to the slow-start of TCP. Also, we can check that we still get oscillation when the max value is 67 (from the graph below).



(max value = 66)



(max value = 67)

3) The average throughput =

Packets per second throughput = ~220pps

```
58.100000000000001 14  0.0010988148496978258 231.0 19 220.625
59.100000000000001 14  0.0010792476102374346 232.0 18 220.81911262798636
~
```

Bytes per second throughout:

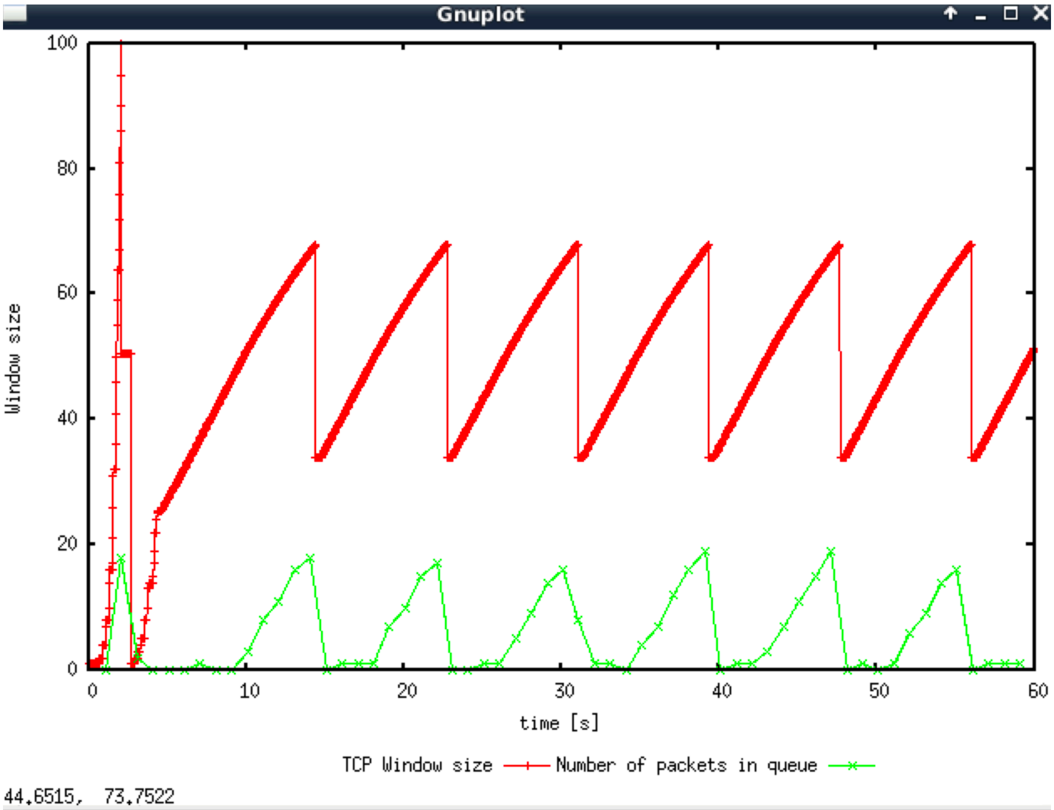
IP + TCP Headers = 20 + 20 = 40 bytes

Payload = 500bytes

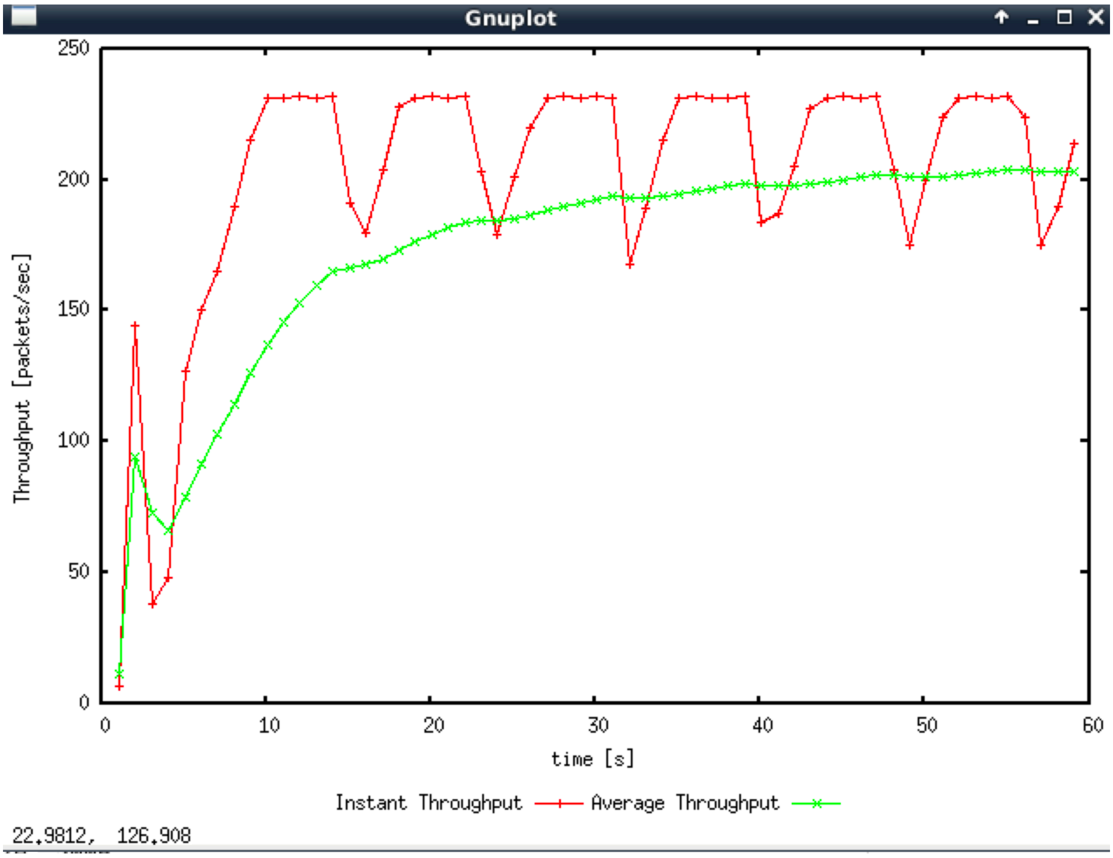
= (500+40) * 8 * 220 = 950,400 bps throughput

4) 100 – 100 * (1000000 – 950,400) / (1000000) = 95.400% utilisation of the link. The average throughput is almost at link capacity.

Question4:



($gnuplot Window.plot for TCP Reno)



($gnuplot WindowTPut.plot for TCP Reno)

From the graph above, we can find that TCP Reno doesn't enter into the slow-start phase when loss occurs.

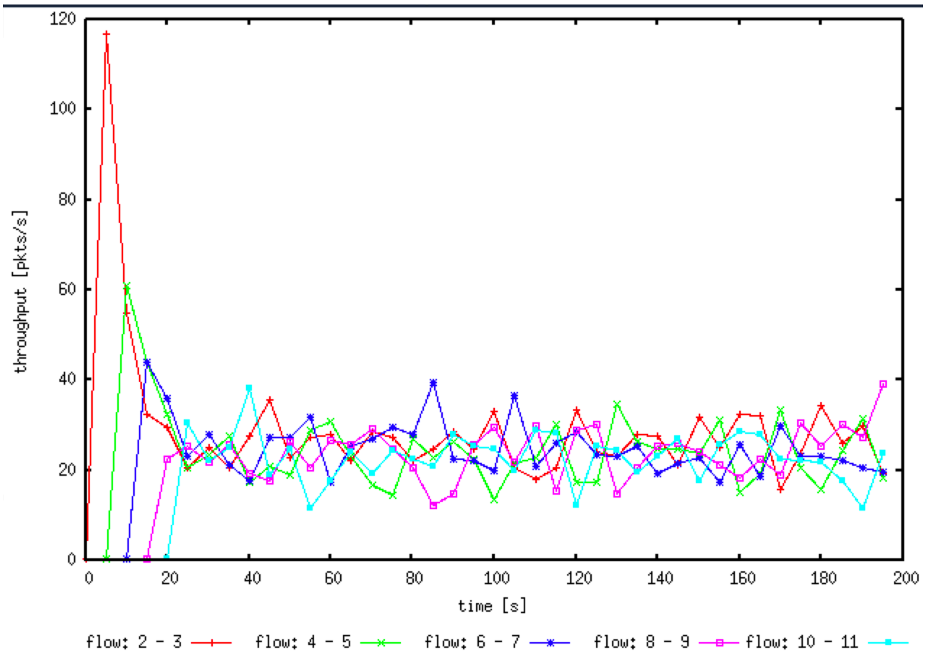
TCP Reno throughput = 200 pps

- (500 + 40) * 8 bits * 200 = 864,000 bps

Throughput previously with TCP Tahoe was 820,800 bps, therefore TCP Reno performs better.

Exercise 2:

Question1:



Each flow gets an equal share of the capacity of the common link, this is shown on the graph above, initially, each flow gets a different capacity, but as time increase, each flow is trying to converge.

Question2:

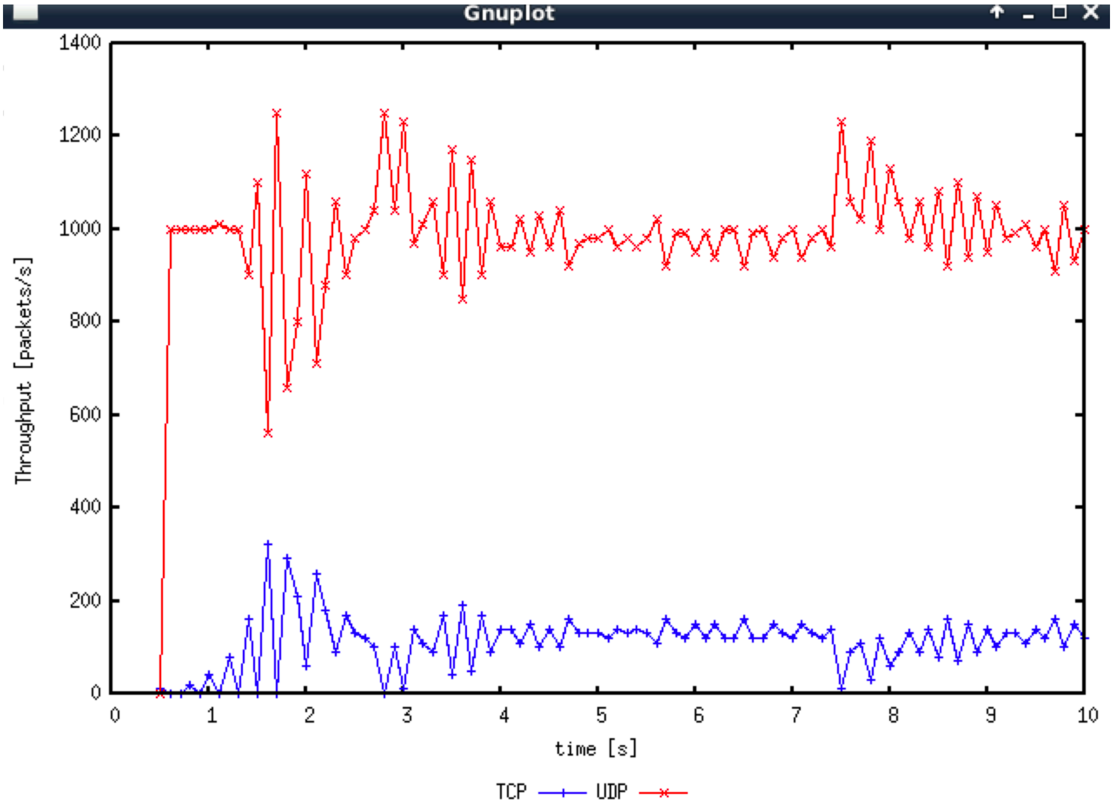
When a new flow is created, the throughput of pre-existing TCP flows will immediately decrease. The reason for this is the additive increase/multiple decrease (AIMD) mechanism of TCP. This mechanism is fair because after multiple congestions, the window size of each flow is nearly equal, and they will receive an equal share of the link.

Exercise 3:

Question1:

I expect when the capacity of the link is 5Mbps, the throughput of both TCP and UDP flow will increase, but UDP will have higher throughput than TCP, since UDP doesn't have congestion control.

Quesiton2:



The reason for UDP flow has higher throughput than TCP flow is that UDP doesn't have congestion control.

Question3:

Advantages:

1. UDP is faster because it doesn't need to handshake before transferring data.
2. UDP is more efficient and has lower latency than TCP, because it doesn't need to do the error checking and retransmit while transferring data.

Disadvantages:

1. UDP is an unreliable and unstable protocol since it would not check the data sent by client is received by server or not.
2. There do not have a congestion control mechanism for UDP, so a link may collapse due to the packets being sent to a blocked link.

If everybody started using UDP instead of TCP, the network would likely to be blocked since it lacks congestion control, and the new packets being sent would easily lost.