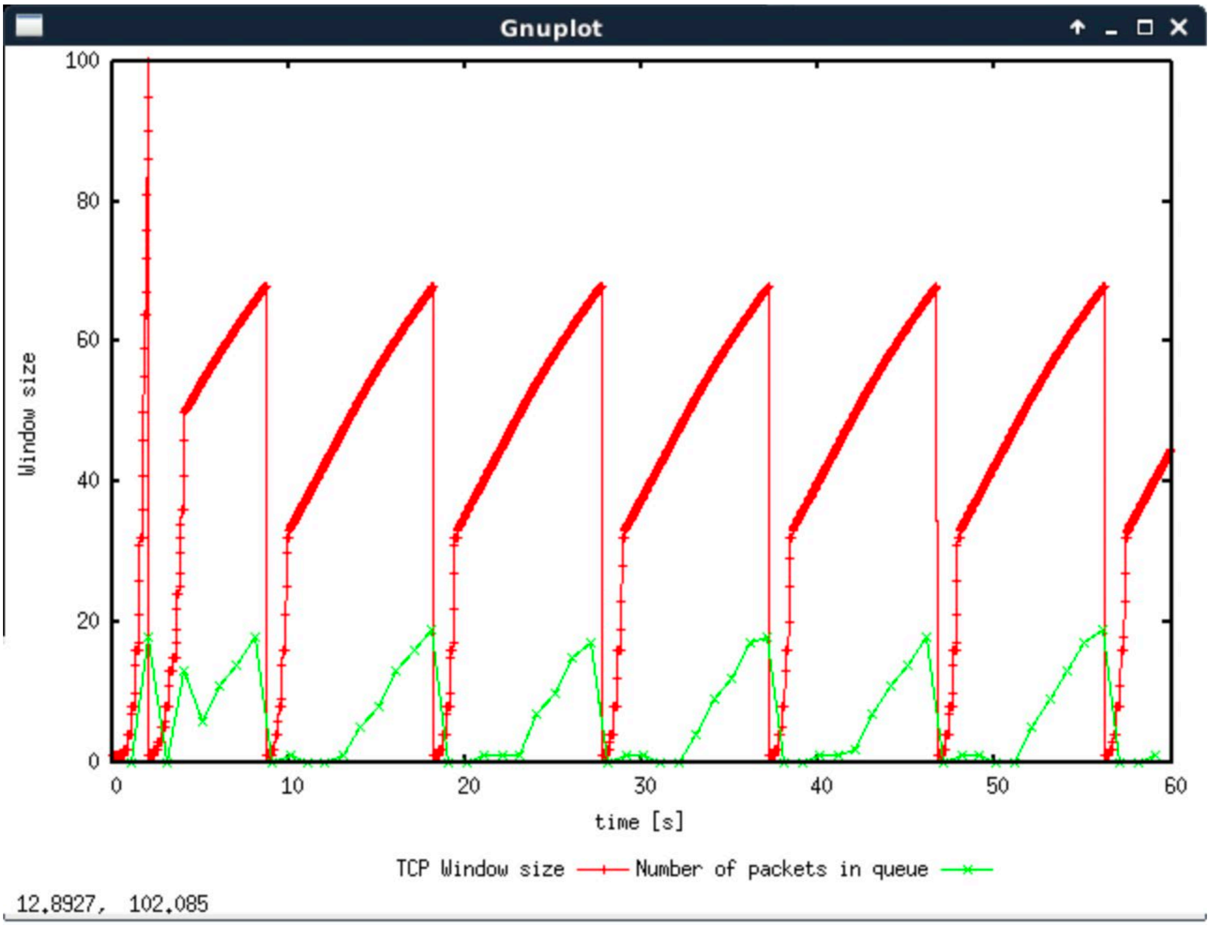
# COMP3331 Lab05 Report

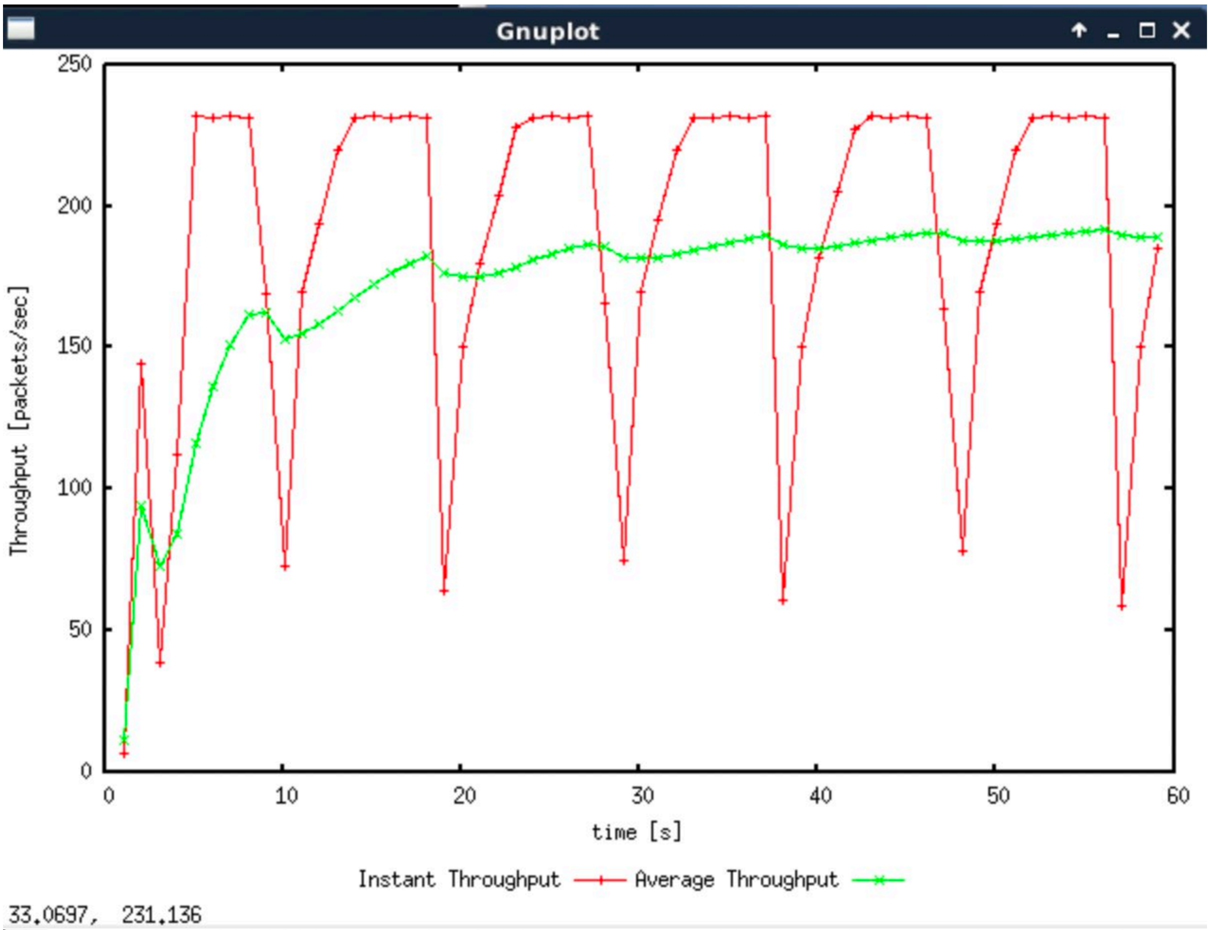
## Exercise 1.

### Question 1.




The maximum window size in this case is 100.
When the congestion window size reaches to 100, either triple duplicate ACK or a timeout have occurred, which cause the congestion window size to reset.
The slow start phase will initiate with congestion window size 1, and the congestion window size will exponentially increase until threshold is 50 (half of maximum window size), then it will start congestion avoidance which linearly increase again until loss.

Question 2.




```
59 59.100000000000001 41  0.0036883771140698092 185.0 1 188.97610921501706
```

The average throughput is approximately 188.97610921501706 packets per second.

Calculate throughput (including header):

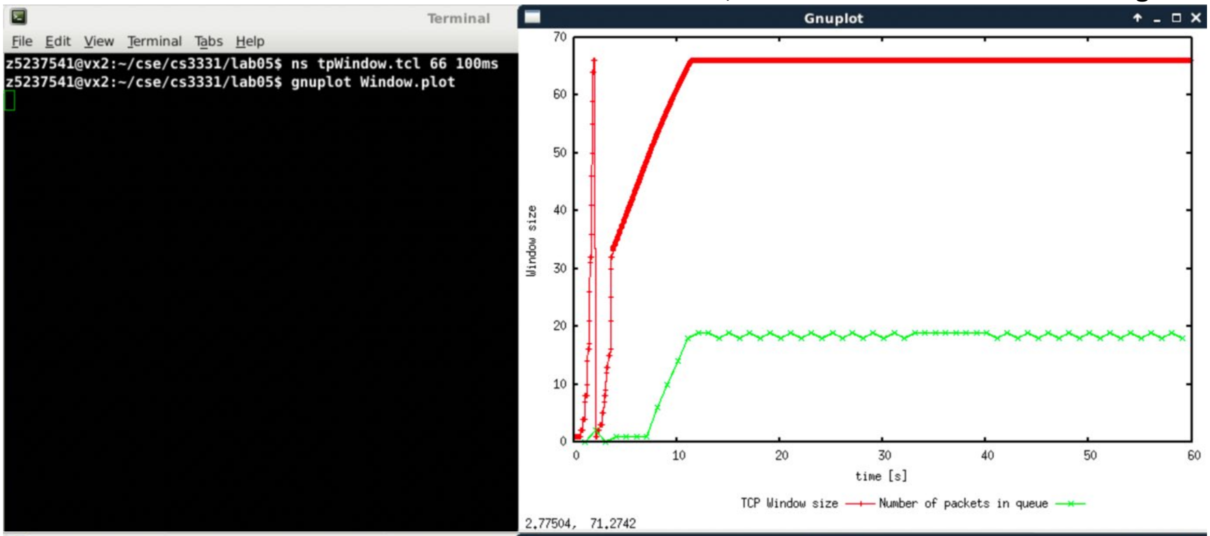
$$188.97610921501706 \times (500 + 20 + 20) = 102047.099\ Bytes/s \cong 816.377\ kbps$$

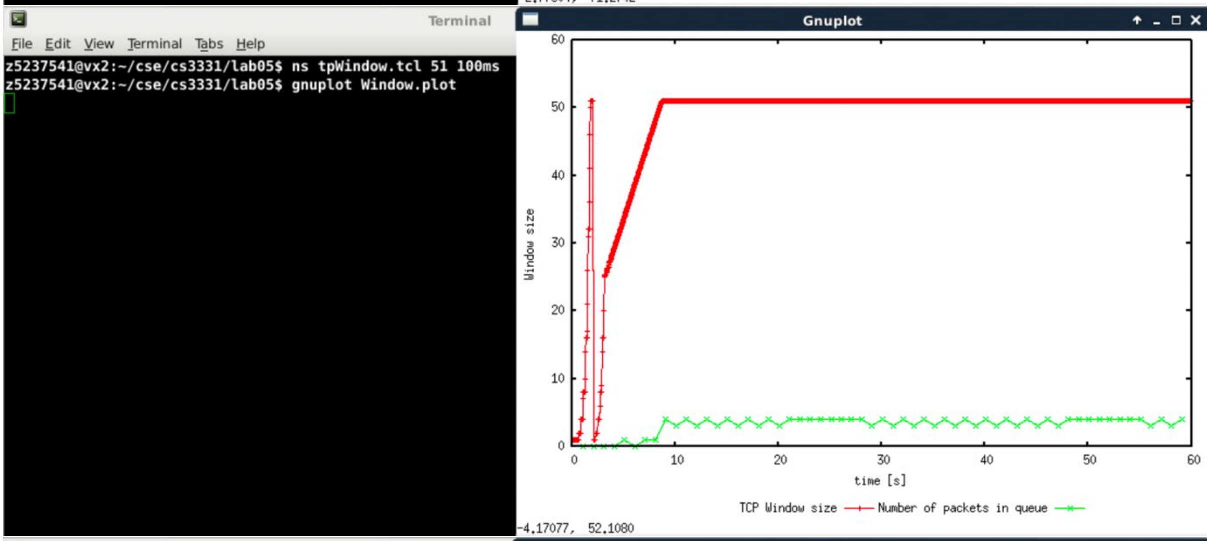
Calculate throughput (without header):

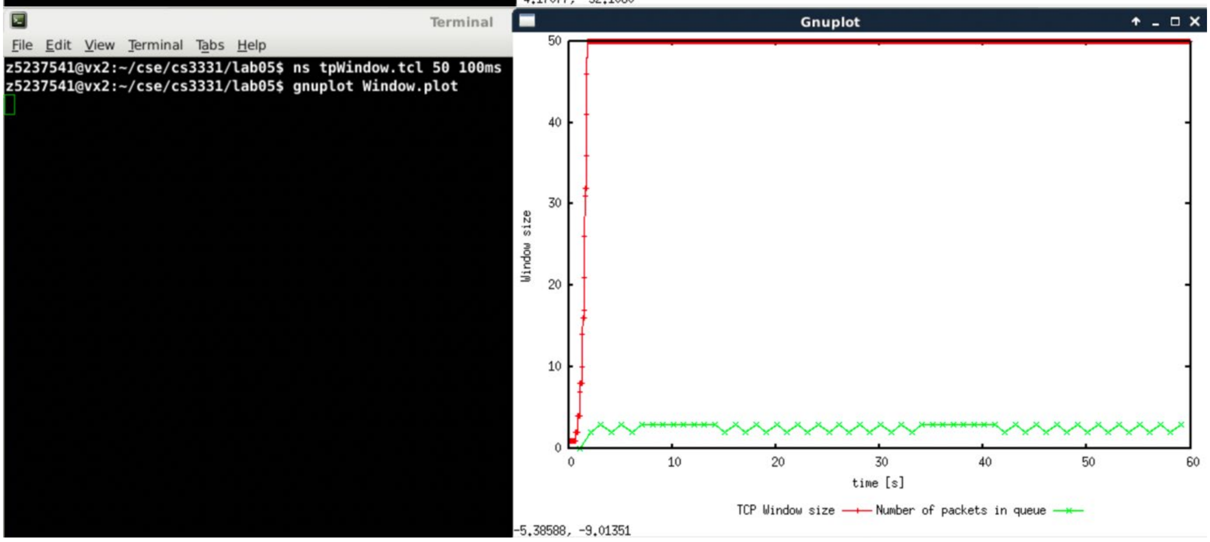
$$188.97610921501706 \times 500 = 94488.055\ Bytes/s \cong 755.904\ kbps$$

## Question 3.

When the max initial window size decreases, the number of oscillating decreases. Once the max initial window size decreases to 66, we will only get one oscillating.
Note that if the max initial window size is lower than 51, then there will be no oscillating.




```
59 59.100000000000001 14  0.0010792476102374346 232.0 18 220.81911262798636
```

The average throughput is approximately 220.81911262798636 packet per seconds.

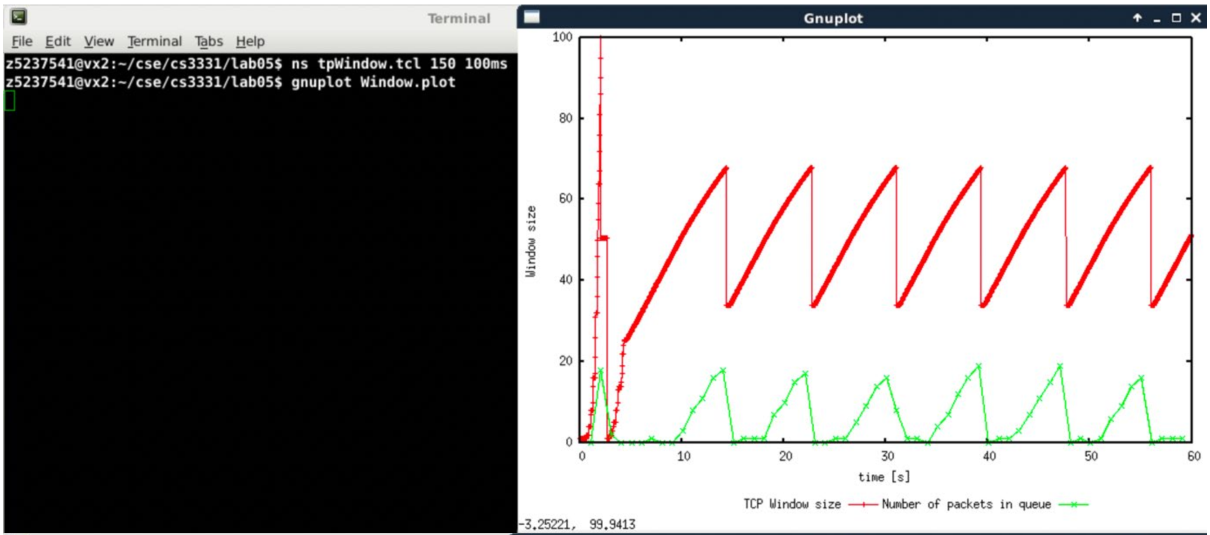
Calculate throughput (including header):

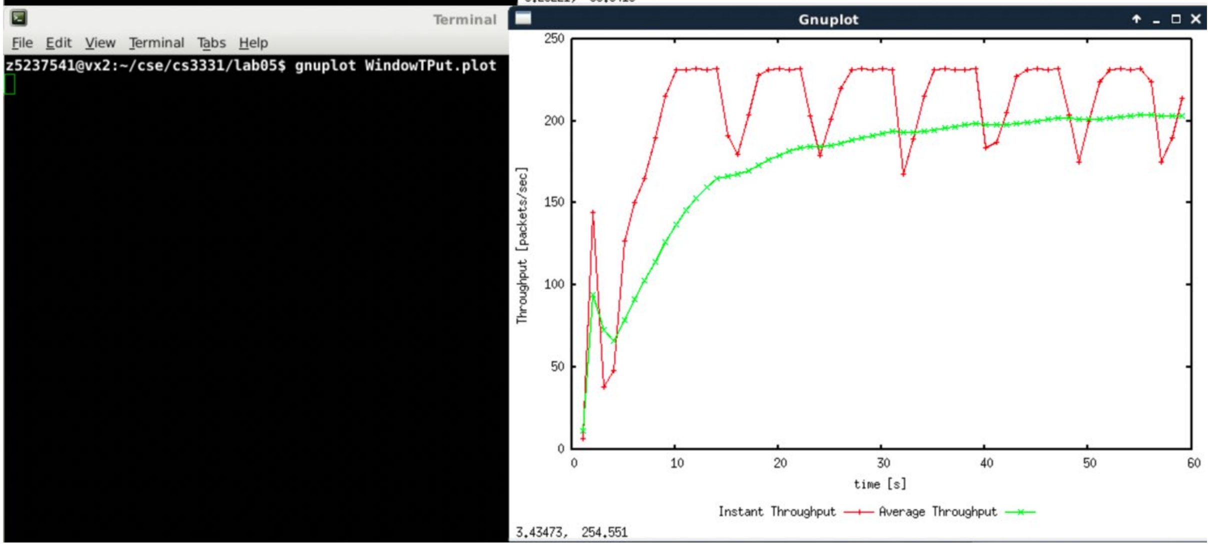
$$220.81911262798636 \times (500 + 20 + 20) = 119242.3208\ Bytes/s = 953.939\ kbps$$

Calculate link capacity:

$$\frac{9539385666}{1000000} \times 100 = 0.9539\%$$

## Question 4.



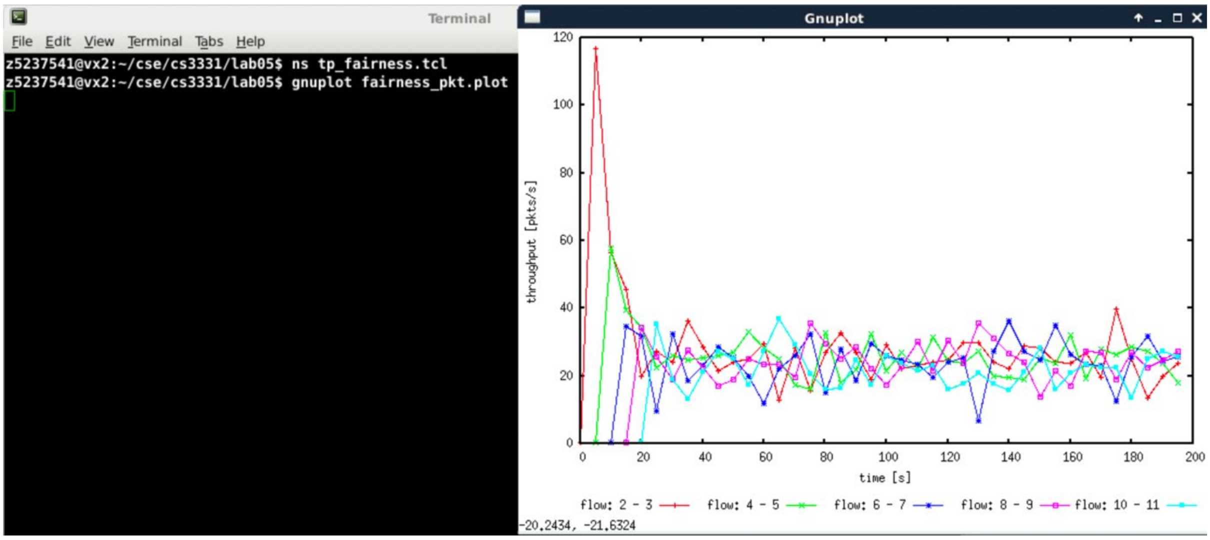

59 59.100000000000001 41 0.0034275204815248286 214.0 1 203.41296928327645

The window size only decreases to zero after slow start phase for TCP Reno.

The average throughput of TCP Reno (203.41296928327645) is higher than the TCP Tahoe (188.9761092150176).

## Exercise 2.

### Question 1.




Each flow gets an equal share of capacity of the common link. Although each flow has different common link at the beginning, when the time increases, each flow increases and decreases throughput until each of them has an average throughput between 20 and 40 packets per second.

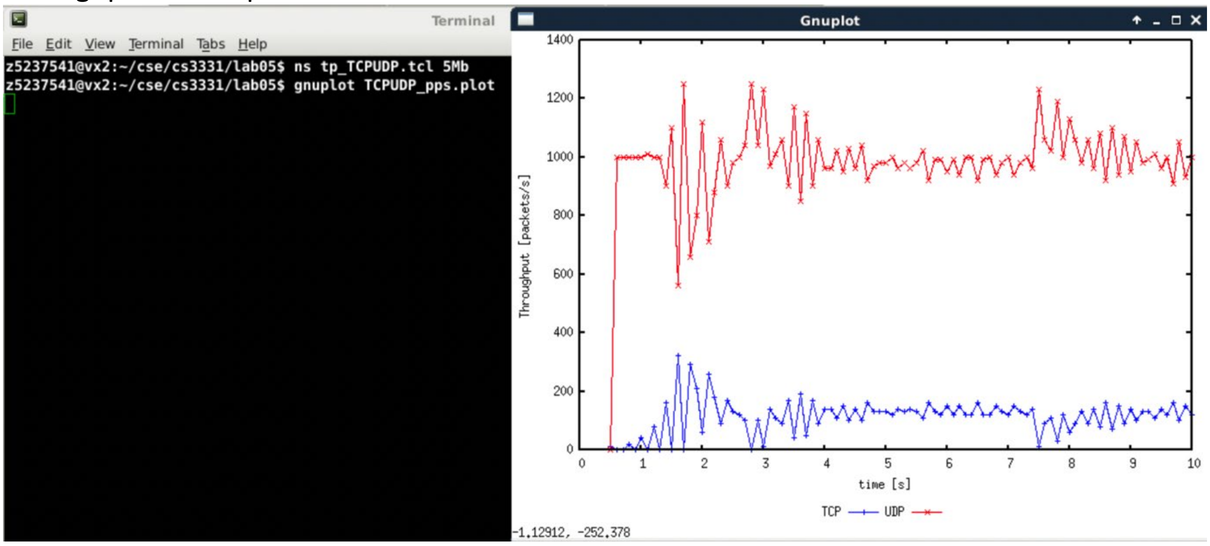
### Question 2.

The throughput of the pre-existing TCP flows will decrease when a flow is created to provide an equal share of common link. The congestion window size increases rapidly at the slow start phase which leads to congestion. Therefore, all flows will adjust to adapt the network, which the behaviour is **fair**.

## Exercise 3.

### Question 1.

TCP has the congestion control while UDP does not. Therefore, UDP will have higher throughput in comparison to TCP.




UDP is red and TCP is blue.

### Question 2.

UDP does not have congestion control mechanism. It provides "best-effort" datagram where applications provide their own reliability and flow control.

TCP has congestion control mechanism. It provides a stable connection and will not overload the link capacity by changing the window size based on conditions.

### Question 3.

Advantage of using UDP over TCP

1. Higher average throughput
2. Transfer rate based on link bandwidth
3. Smaller packet size

Disadvantage of using UDP over TCP

1. No congestion controls
2. Packets arrival out of order
3. Unaware of packets loss and corrupt packets

If everyone started to use UDP rather than TCP, there will be many packets loss, performance suffer and network congested, and difficult to detect corrupted packets.