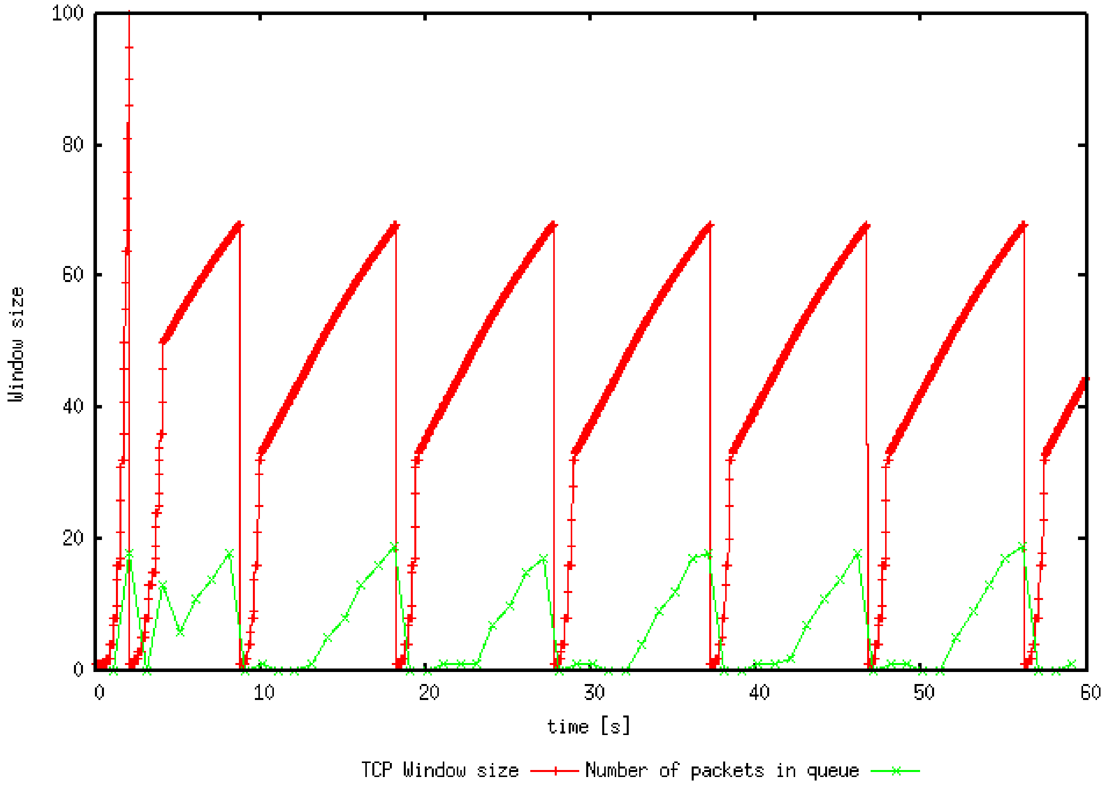
Exercise1

Question1

Here is the picture result of Question 1. From **the first picture**, we can find **the maximum size of the congestion window** which TCP flow could reach in this case is nearly **68**. We can also find this result in the second picture which is **67.9581.** Furthermore, from this picture, we can also find that when the window size goes to the max value, **the TCP stopped the process of congestion avoidance**. This is because it exists a **packet loss** situation in this process. After that, the TCP set the **ssthresh** value to the **half value of the maximum size**. Then the TCP returns to **the slow start stage**.



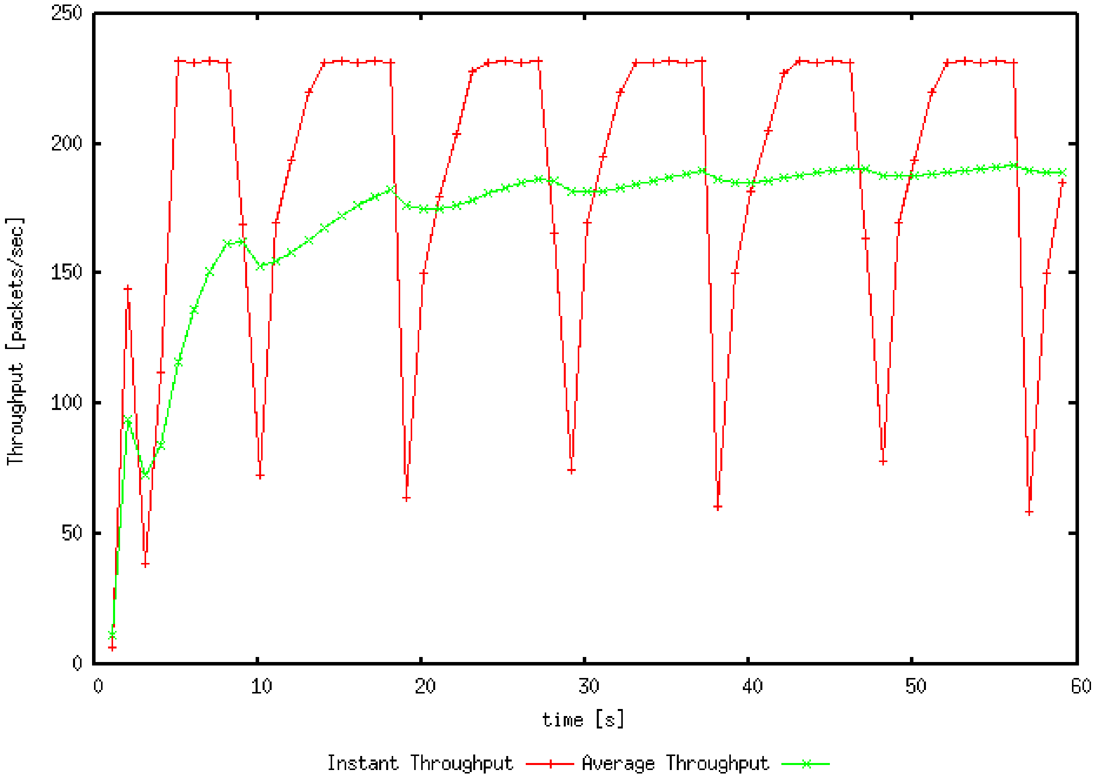
8.4599999999999085	67.0693
8.4799999999999081	67.1438
8.4999999999999076	67.2034
8.5199999999999072	67.2778
8.5399999999999068	67.352
8.5599999999999064	67.4114
8.5799999999999059	67.4856
8.5999999999999055	67.5596
8.6199999999999051	67.6188
8.6399999999999046	67.6927
8.6599999999999042	67.7518
8.6799999999999038	67.8255
8.6999999999999034	67.8992
8.7199999999999029	67.9581
8.7399999999999025	1
8.7599999999999021	1
8.7799999999999017	1

Question2

From the WindowMon.tr file, we can get the following result. From this picture, we can find that the average value of packet throughput is about **189 packets/seconds**.

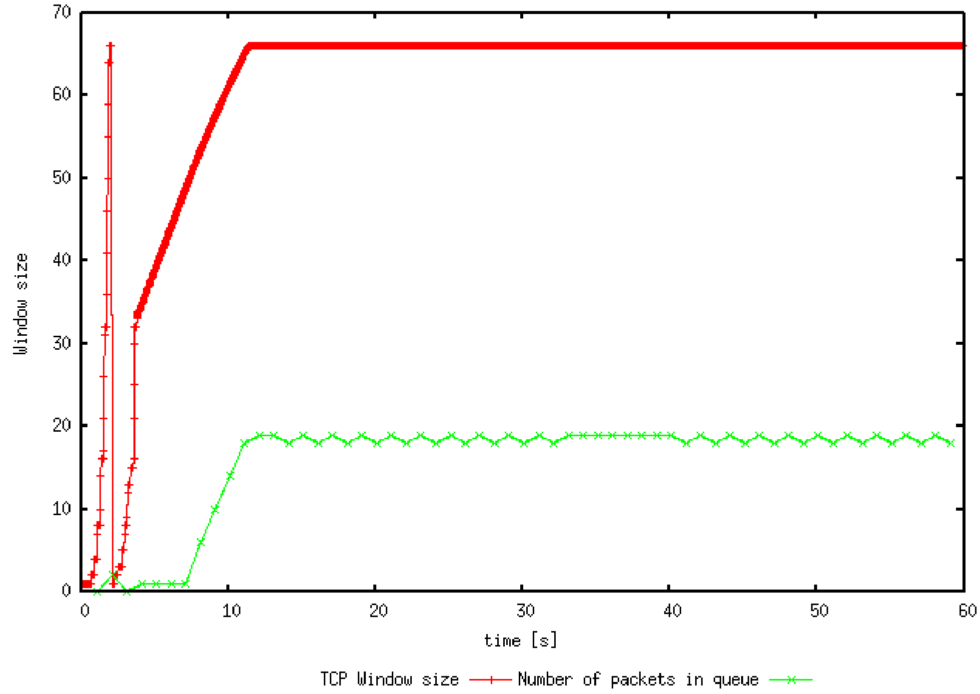
```
188.97610921501706
```

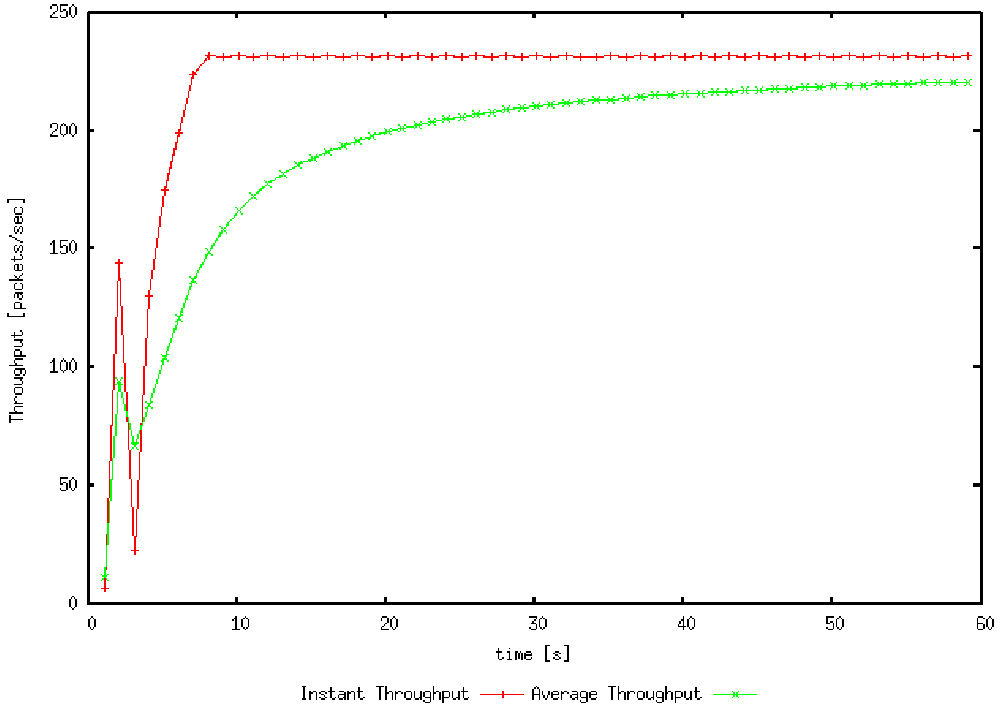
From the question, we can also get that the data load of each packet is **500 – 20 = 480 bytes**. Therefore, the final result of the average throughput is **189 * 480 * 8 = 725760 bps**.



Question3

The maximum value of the congestion window in which TCP **stopped fluctuation is 66**. If we set the value above 66 the TCP process stays the same which can find the fluctuation in the graph. If we set the value lower than 66, the TCP will stay stable. This means if we set the value lower than 66, we can find that the flow will keep stable when the value of the new congestion window size is lower than 66.

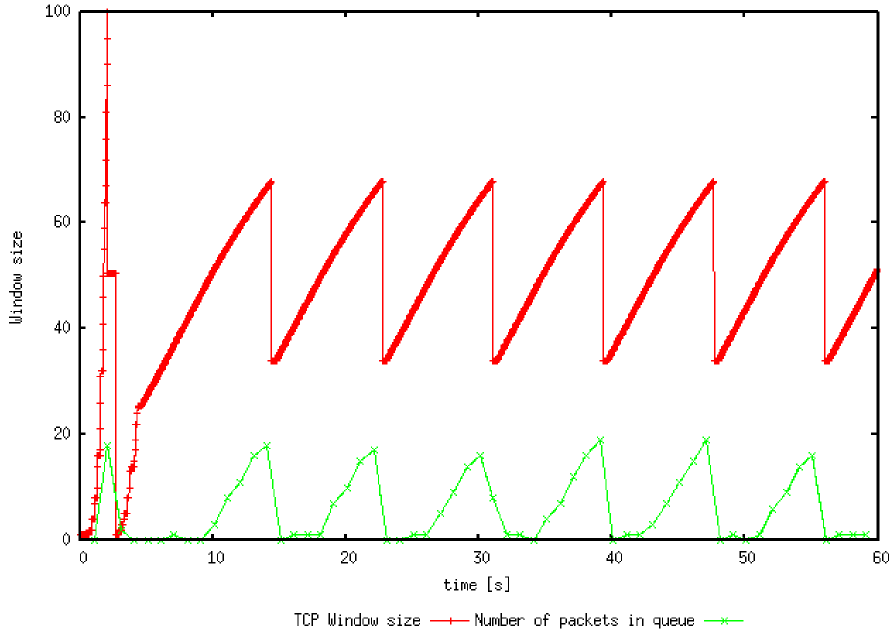


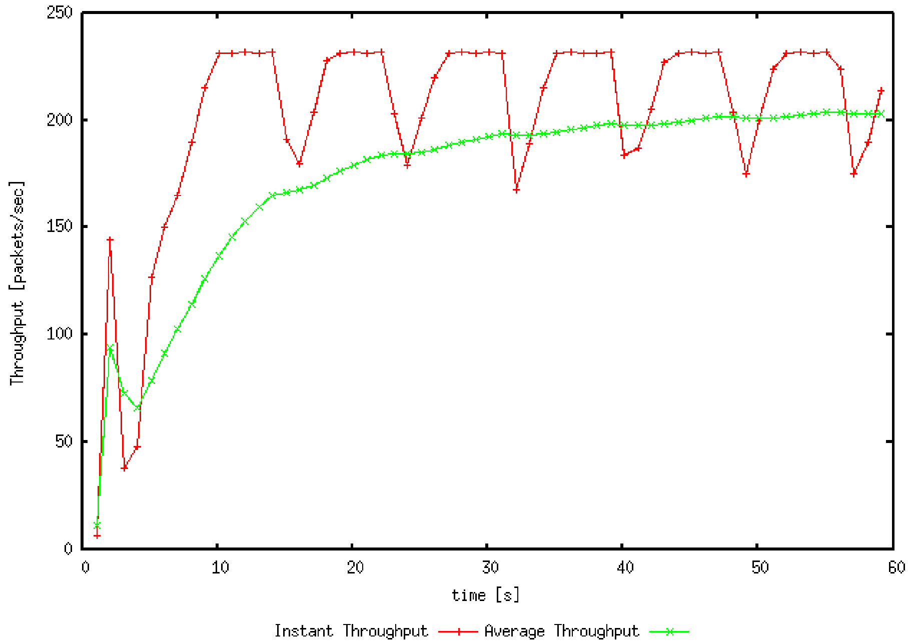


We can also find the value of packet throughput from the WindowMon.tr file, which is about **221(220.819)**. Therefore, the final throughput is **221 * 480 * 8 = 848640 bps = 0.85 Mbps**. Compared with the link capacity, 0.85mbps is **less than the link capacity**, which is 1 Mbps.

Question4

From the previous result, the congestion window drops to 0 every time, if there is a packet loss. This is the situation of the TCP Tahoe. If we changed Tahoe to Reno, we can get the following result. In the following picture, the congestion window only drops to 0 once which is the slow start, after that, if there exist a packet loss, it turns to the fast recovery stage. Then, the size of the congestion window becomes half.
From the WindowMon.tr file, we can also find the packet throughput for the Reno is about **203 packets per second**. This value is **larger than the result of the Tahoe which is 189**.

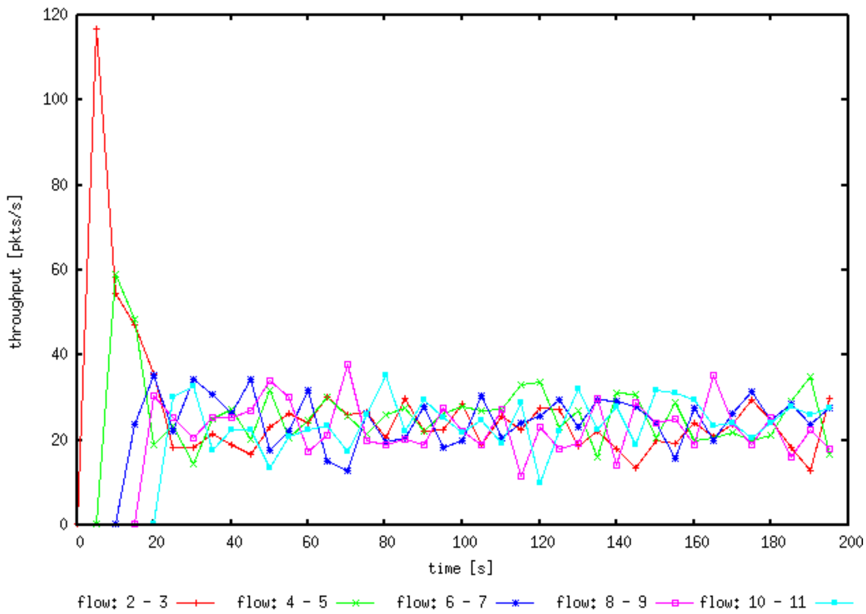




Exercise2

Question1

Here is the plot result. From this plot, we can find that the first flow (which is the red line) reaches a packet throughput at about **120 packets per second**. Furthermore, assume that the maximum throughput is 120 packets per second. This is because from the plot we can easily find that when the second flow starts the first flow goes down as soon as possible. When all these five flows become stable, we can find every flows' packet throughput is **between 20 and 30 packets per second**. The throughput value of 20-30 packets per second is **just 1/5 of the maximum throughput**. Therefore, we can conclude that all these **five flows would share the capacity of the common link in an equal way**.



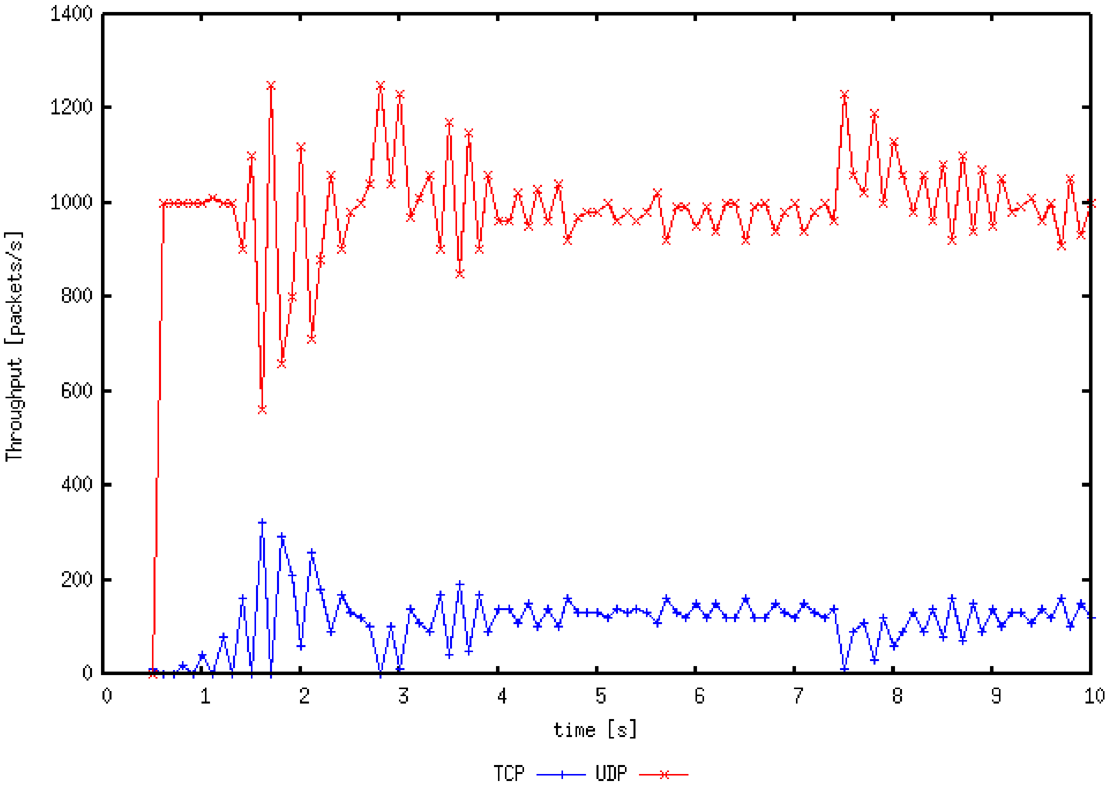
Question2

From the previous picture, we can find that when the new flow starts the previous flow will drop. This is because the previous flow reaches the maximum throughput, when the new flow starts, the previous flow will be packet loss. In this situation, the packet loss will cause the previous flow turns to slow start if the packet was lost. Otherwise, if we can get 3 duplicate ACKs during this process, it turns to congestion avoidance. In this way, every flow will start from some low throughputs. This means if choose the slow start the throughput will set as 1 packet per second. Otherwise, if choose congestion avoidance, the current throughput will be half of the previous throughput. In other words, in the final situation, every flow will have the same packet throughput. Therefore, it is fair for each flow.

Exercise3

Question1

The **UDP flow will have a throughput at 4 Mbps**. Furthermore, the **TCP flow only has a throughput at 1 Mbps**, which is 5 Mbps – 4 Mbps = 1 Mbps. Here is the plot result of this question. The **blue flow** in the plot is **TCP**. The **red flow** in the plot is **UDP**.



Question2

This is because there is **no congestion control function in UDP**. This means UDP can start at the full speed, 4 Mbps in this case. **UDP occupied 4 Mbps in this case, there are only 1 Mbps left in the link**. The TCP will start from 0 throughputs. This is because of the slow start function. After the TCP reaches 1 Mbps, there will be packets loss for both the UDP and the TCP. **Based on the function of TCP, TCP will turn to a slow start or congestion avoidance**. In this way, UDP can keep the current speed which is 4 Mbps. This means the speed fluctuation (caused by packet loss) of the TCP flow will not affect the speed of the UDP flow.

After the packet loss, the TCP set its ssthresh to 0.5 Mbps (half of 1 Mbps). Therefore, the TCP can also reach the max speed of 1 Mbps. After the **TCP flow becomes stable**, we can find that **2 flows are stabilized with each other**.

Question3

Advantages:
1. The UDP protocol has no connections established, so there will be no handshakes as TCP.
2. Considered the congestion control mechanism, UDP protocol has nod congestion control function, so using the UDP protocol to transfer files is faster than using the TCP.
3. As for the UDP protocol, there is no need to maintain the connection states.

Disadvantages:
1. The UDP protocol cannot provide a reliable data transfer service. This means the UDP protocol cannot make sure whether the data packet is transferred successfully.
2. If some applications cannot accept the packet loss, they will not choose UDP protocol to finish the data transfer.

If everyone changes to use the UDP protocol to transfer data, due to the UDP protocol has no congestion control mechanism, too many UDP connections can jam the link. This will also cause a high rate of packet loss of both TCP and UDP. Therefore, everyone wants to resend the data, this will make this situation worse in the network. This kind of situation forms a vicious circle in the network. Finally, the network is down and not functional.