

## Exercise 1

### Question 1

The maximum size of the congestion window that the TCP flow reaches is 100. When the congestion window reaches this value, there is a triple duplicate ACK/timeout and the congestion window size drops to 1. The slow start threshold (ssthresh) is set to half of 100 which is 50. This is done to avoid congestion. The slow start phase will start again after this with the window size being 1.

### Question 2

From the WindowTPut graph, the average throughput is around 188 packets/sec which is confirmed in WindowMon.tr where the last data point (line 59) shows the average is 188.97610921501706 packets/second. The average throughput in bytes/second is:  $[500 \text{ (packet payload)} + 20 \text{ (IP header)} + 20] \text{ (TCP header)} * 188.97610921501706 \text{ (avg tput)} = 102047.098976 \text{ bps}$ .

### Question 3

As the window size decreases, the amount of oscillation in TCP decreases. 50 is the value of the maximum congestion window at which TCP stops oscillating. The average throughput at this point is 227.73037542662115 packets/sec which is 122974.40273 bps. Average throughput compare to link capacity of 1Mbps (125000 bytes):  $122974.40273 / 125000 * 100 = 98.38\%$

### Question 4

The window size hits zero once after the slow start phase for Reno while in Tahoe, it hits zero multiple times. Reno has a higher average throughput than Tahoe ( $\sim 203$  vs 188).

## Exercise 2

### Question 1

Each flow gets an equal share of the capacity of the common link because each flow eventually fluctuates within the 20 to 40 packets per second range. The flows only have a different share of the common link at the start, but as time goes on, the share gets evenly spread out, so TCP is fair.

### Question 2

The throughput of pre-existing TCP flows decreases when a new flow is created in order to provide an equal share. The mechanisms of TCP that contribute to this behaviour are that the congestion window size increases rapidly during the

slow start phase, causing congestion. As a result, all flows will adjust to adapt to the network so this behaviour is fair.

## **Exercise 3**

### **Question 1**

TCP has congestion control, so it provides a stable connection while not overloading the link by varying the window size based on the conditions. UDP, on the other hand, does not have congestion control so it provides best-effort datagram, leaving it to the application to provide its own reliability and flow control. Thus, UDP will have more throughput due to not having a congestion control mechanism.

### **Question 2**

The advantages of using UDP instead of TCP for a file transfer when the connection is competing with other flows for the same link are:

- Higher average throughput
- Smaller packet size
- Transfer rate is based off link bandwidth.

The disadvantages are: - No congestion control - Unaware of package loss and corrupted packages - Packets can arrive out of order

If everybody started using UDP instead of TCP, there will be a lot of package loss, more network congestion leading to lower performance and more difficulty detecting corrupted packages.