Lab Report

Participants

Name	E-mail	Social Security Number
Dhiru Kholia	dhiru.kholia@gmail.com	850227-8255
Jack Shabo	jshabo@kth.se	930618-3717
Max Lindblad	maxlindb@kth.se	931116-4611
William Schröder	wisc@kth.se	931022-6791

WireShark Data

The WireShark data is available in the ‘output_data’ and ‘graphs’ folders found in the same zip-file as this report.

Answers to Questions

Question 10.1.7

a. 20 packets total
b. They take values of [0, 10002].
c. 12 packets do not carry a data payload.
d. The amount of user data transferred is 10000 bytes.
e. From WireShark (see output_10_summary): TCP total: 11336 bytes, UDP total: 10780 bytes. UDP is more efficient in terms of overhead with the difference: 11336 - 10780 = 556 bytes. Efficiency calculation (sent data / user data) for TCP: 10000/11336 = 88%, Efficiency for UDP: 10000/10780 = 92%.

Question 11.1.6

a. The first three packets (1, 2, 3) make up the three way handshake. The flags are: [SYN], then [SYN, ACK] and finally [ACK].
b. The initial sequence numbers are 391360401 for the client and 3874018891 for the server.

c. The first packet containing application data is the fourth packet.
d. The initial window size is 5840 for the client and 28960 for the server.
e. It takes roughly 0.3 milliseconds to open the connection.

Question 11.1.7

a. The last three packets (19, 20, 21) are for closing the connection.
b. The flags are: [FIN, ACK], then [FIN, ACK] and finally [ACK]

Question 11.2.5

a. It closes the connection by sending back a packet with the flags [RST, ACK].
b. It takes roughly 0.08 milliseconds to close the connection.

(Answer to the NOTE):
The sender will retransmit the SYN packet until it received a response. The most likely outcome is however that the sending application has a timer/set of attempts that cancel out the request after a while.

Question 12.1.4

a. There was in total of 167 packets sent and received, so our answer is that a lot of segments were sent.
b. The data consists of the different characters.
c. Nagle's Algorithm and/or Delayed Acknowledgement causes this. Both of these can cause segments to be grouped together, thus leading to less segments being sent.
d. This is because of Delayed Acknowledgement. The delay is roughly 150 milliseconds.
e. Client windows size appears to be steadily at 181 and server window size appears to vary around 229 - 245.

Question 12.1.5

a. Yes, multiple characters are in a single packet.

Question 12.2.5

a. On average, the sender sends two packets before receiving an ACK. TCP seems to delay acknowledgement for a certain amount of time before sending the ACK for multiple packet.
b. For a single packet, it will acknowledge 1 500 packets, but sometimes it ACKs 3 000 when it ACKs two packets at the same time.

c. The receiver's window size start around 5 792 and increases up to 130 000 towards the end.
d. It takes around 2,651 ms for packet no 522 to get acknowledgement (packet 555).
e. No, the sender does not have time to send that much data before it receives an ACK.

Question 13.1.4

a. One unique packet is sent 6 times during retransmission timeout; 1 initially and then 5 retries.
b. Retransmission does not end until the sender has received an ACK from the receiver for the retransmitted packet.

Question 14.1.4 (Slow Link)

There are some packet losses, where the receiver sends duplicate ACKs to indicate that it is missing a packet. We can see this happening for packet 49-65 for example.

Question 14.1.5 (Slow Link)

a. The congestion window does follow the rules of the slow-start phase at the start of the transmission. The initial window size is very low but it rapidly increases in size for each acknowledgement sent.
b. Yes, after the packet losses there is a quick increase in sequence numbers.

Question 14.1.4 (Fast Link)

For the fast link there were no packet losses, everything seemed to be going smoothly.

Question 14.1.5 (Fast Link)

a. Yes, between time index 0,005 and 0.006 the sequence numbers increase very rapidly, and with this also the window sizes (as seen in WireShark on these packets).
b. No, since there were no packet losses there can not be fast recovery.