Lab 04 Report

`Lines in this font represent terminal commands and terminal output.`

Lines beginning with `$` are the terminal commands that were run.

Lines in this font are the answers to the questions.

EXERCISE 1

1. The IP address of `gaia.cs.umass.edu` is 128.119.245.12. It is sending and receiving from port 80 for this connection. The client (source) computer has IP address 192.168.1.102, and is sending and receiving from port 1161.

2. The sequence number of the segment containing the HTTP POST command is 232129013.

3. The sequence numbers of the first 6 segments in the TCP connection, their time sent and the time of their acknowledgment received are shown below. The formula used to calculate the $k+1$ EstimatedRTT is below.

$$\text{EstimatedRTT}_{k+1} = (1-\alpha)\cdot\text{EstimatedRTT}_k + \alpha\cdot\text{SampleRTT}_k$$

Take $\text{EstimatedRTT}_1 = \text{SampleRTT}_1$ as the question indicates. The SampleRTT is calculated from the difference between the Time Sent Column and the Time ACK Received Column for each Row.

Sequence Number	Time Sent	Time ACK Received	SampleRTT
232129013	0.026477	0.053937	0.02746
232129578	0.041737	0.077294	0.035557
232131038	0.054026	0.124085	0.070059
232132498	0.054690	0.169118	0.114428
232133958	0.077405	0.217299	0.139894
232135418	0.078157	0.267802	0.189645

$$\begin{aligned}
\text{EstimatedRTT}_1 &= 0.02746\\
\text{EstimatedRTT}_2 &= (1-0.125)*0.027466+0.125*0.035557\\
&= 0.028477375\\
\text{EstimatedRTT}_3 &= (1-0.125)*0.028477375+0.125*0.070059\\
&= 0.033670484375\\
\text{EstimatedRTT}_4 &= (1-0.125)*0.033670484375+0.125*0.114428\\
&= 0.043765173828125\\
\text{EstimatedRTT}_5 &= (1-0.125)*0.043765173828125+0.125*0.139894\\
&= 0.05578127709960937\\
\text{EstimatedRTT}_6 &= (1-0.125)*0.05578127709960937+0.125*0.189645\\
&= 0.07251424246215821
\end{aligned}$$

4. The length of segment 1 is 585 bytes, where 20 of these bytes are for the TCP header. The lengths of segments 2-6 are all 1480 bytes, including 20 bytes of header.

5. The minimum amount of available buffer space advertised by the receiver is 5840 bytes in the receiver's SYNACK message, for the entirety of the trace. No throttling occurs as the available buffer space only increases over the duration of the trace.

6. No retransmissions occur as there are no duplicate sequence numbers from the client, and they only increase for the duration of the trace. There are also no duplicate ACKs for the entirety of the trace.

7. In this case, the receiver typically ACK'd two segments at a time, which contained 2920 bytes all up. Segment 52 is where the first 'delayed' ACK occurred, and from then on, the majority of the ACKs where delayed ACKs that ACK'd two segments at a time. However, prior to segment 52, ACKs would ACK individual segments, containing 1460 bytes.

8. The throughput for the connection is the number of bytes transferred over the time taken to transfer the bytes. The number of bytes is found by subtracting the initial sequence number of the source from the last ACK number of the receiver. In this trace, the total number of bytes transferred is $232293103 - 232129012 = 164091$. The time taken to transfer this number of bytes was 5.651141 seconds. The throughput is thus 29036.7909773 bytes/sec.

EXERCISE 2

1. The sequence number used to initiate the TCP connection is 2818463618.

2. The sequence number of the server's SYNACK is 1247095790. The ACK number is 2818463619, which is calculated by adding one to the sequence number of the client's SYN sequence number, as this is the position where the server expects the next byte.

3. The sequence number of the ACK sent by the client in response to the server's SYNACK is 2818463619. The ACK number of the ACK message sent by the client is 1247095791. This ACK number indicates that the SYNACK sent by the server contained one byte. However the ACK from the client contained no bytes, as an ACK contains no data, by definition of the byte stream.

4. The client did the active close, as they were the first to send the FIN message. This is a simultaneous close, as the server also sent a FIN message, before receiving the FIN message from the client.

5. The amount of data transferred from the client to the server is given by the ACK number of the server's last ACK minus the Client ISN. In this example, that is $2818463653 - 2818463618 = 35$. The amount of data transferred from the server to the client is given by the ACK numberof the client's last ACK minus the Server ISN. In this example, that is $1247095832 - 1247095790 = 42$.