Lab4: Exploring TCP

Exercise 1

***Question 1* . What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection? What is the IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu?**

Gaia
IP: 128.119.245.12
Port: 80

Client
IP: 192.168.1.102
Port: 1161

***Question 2.* What is the sequence number of the TCP segment containing the HTTP POST command? Note that in order to find the POST command, you'll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with a "POST" within its DATA field.**

Sequence number: 232129013

***Question 3.* Consider the TCP segment containing the HTTP POST as the first segment in the TCP connection. What are the sequence numbers of the first six segments in the TCP connection (including the segment containing the HTTP POST) sent from the client to the web server (Do not consider the ACKs received from the server as part of these six segments)? At what time was each segment sent? When was the ACK for each segment received? Given the difference between when each TCP segment was sent, and when its acknowledgement was received, what is the RTT value for each of the six segments? What is the *EstimatedRTT* value (see relevant parts of Section 3.5 or lecture slides) after the receipt of each ACK? Assume that the initial value of *EstimatedRTT* is equal to the measured RTT (*SampleRTT*) for the first segment, and then is computed using the *EstimatedRTT* equation for all subsequent segments. Set alpha to 0.125.**

Sequence Numbers → Time sent → Time Received of first 6 segments:

	Seq No.	Time Sent (secs)	Time Received (secs)	RTT	Estimated RTT
Segment 1	232129012	0.026477	0.053937	0.02746	0.02746
Segment 2	232129578	0.041737	0.077294	0.035557	0.0285
Segment 3	232131038	0.054026	0.124085	0.070059	0.0337
Segment 4	232132498	0.054690	0.169118	0.11443	0.0438
Segment 5	232133958	0.077405	0.217299	0.13989	0.0558
Segment 6	232135418	0.078157	0.267802	0.18964	0.0725

Question 4. **What is the length of each of the first six TCP segments?**
The lengths for each of the segments are:

1) 565 bytes
2) 1460 bytes
3) 1460 bytes
4) 1460 bytes
5) 1460 bytes
6) 1460 bytes

Question 5. **What is the minimum amount of available buffer space advertised at the receiver for the entire trace? Does the lack of receiver buffer space ever throttle the sender?**

The minimum buffer is 5840 bytes. It does not ever throttle the sender

Question 6. Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question?

No there aren’t any retransmitted segments. This was done by checking if the sequence number was smaller than the previous segments.

Question 7. **How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment (recall the discussion about delayed acks from the lecture notes or Section 3.5 of the text).**

The receiver typically ACKs 1460 bytes. Yes, the receiver is ACKing every other received segment in segments 10 and 11. This is done for efficiency as there is wastage sending a new Ack for every single segment

Question 8. **What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.**

Total data transmitted is the Ack number of the last segment – sequence number of the first segment

Therefore, total data transmitted is: 232,293,103 – 232,129,012 = 164,091 bytes

Transmission time is 5.455830 (last transferred segment) – 0.026477 (first segment) = 5.42494 secs

Therefore, throughput is 164,091 bytes / 5.42494 secs = 30223 bps

Exercise 2

***Question 1* . What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and server?**

Sequence Number is 2,818,463,618

***Question 2.* What is the sequence number of the SYNACK segment sent by the server to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment? How did the server determine that value?**

Sequence number for SYNACK is 1,247,095,790
Ack number returned is 2,818,463,619
This was determined by adding a byte to the Sequence number received from the SYN

***Question 3* . What is the sequence number of the ACK segment sent by the client computer in response to the SYNACK? What is the value of the Acknowledgment field in this ACK segment? Does this segment contain any data?**

Sequence number is 2,818,463,619
Value of the Ack is 1,247,095,791
This segment does not contains 33 bytes of data

***Question 4* . Who has done the active close? client or the server? how you have determined this? What type of closure has been performed? 3 Segment (FIN/FINACK/ACK), 4 Segment (FIN/ACK/FIN/ACK) or Simultaneous close?**

Both the client and server did the active close. As seen in segments 304 and 305, both client and server sent a FINACK to each other as their final sending segment. This shows that it’s a simultaneous close

***Question 5* . How many data bytes have been transferred from the client to the server and from the server to the client during the whole duration of the connection? What relationship does this have with the Initial Sequence Number and the final ACK received from the other side?**

Bytes from client to server:
2,818,463,652-2,818,463,619 = 33 bytes

Bytes from server to client:
1247095831 ‐ 1247095791 = 40 bytes

The difference between the final ACK and the initial sequence number is the total data that is transferred. This does not include the SYN and FIN flags because they do not contain any data in their segments.