Exercise 1: Understanding TCP using Wireshark

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?

- Server IP Address: 128.119.245.12
- Server Port: 80
- Client IP Address: 192.168.1.102
- Client 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.

- Seq=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.

#	Seq#	Time Sent	Time Received	RTT	EstimatedRTT
1	232129013	0.026477	0.053937	0.02746	0.02746
2	232129578	0.041737	0.077294	0.035557	0.02847213
3	232131038	0.054026	0.124085	0.070059	0.03367048
4	232132498	0.054690	0.169118	0.114428	0.04376517
5	232133958	0.077405	0.217299	0.139894	0.05578128
6	232135418	0.078157	0.267802	0.189645	0.07251424

- EstimatedRTT = (1 - α)EstimatedRTT + α(SampleRTT)
  - where $\alpha = 0.125$

Question 4. What is the length of each of the first six TCP segments?

#	Seq#	Length
1	232129013	565
2	232129578	1460
3	232131038	1460
4	232132498	1460
5	232133958	1460
6	232135418	1460

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?

- Minimum buffer space (seen in first ACK) = 5840
- Maximum buffer space = 17520
- After observing the trace, we don't see the sender being throttled due to a lack of buffer space.

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 retransmitted segments have been observed.
- We check for consecutive seq# from client to server OR consecutive acks from server to client.
    - Also note that we have observed increasing sequence numbers.

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).

- Data received between each ACK is the difference between two consecutive ACKs.
    - This is usually 1460 in the trace example (with the first ACK length being 565).
- The receiver is ACKing every other received segment in No18 onwards with data of length 1460.

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

Throughput = Total Data Transmitted / Total Time Taken
= (Last Seq# – First Seq#) / (Last Seg Time – First Seq Time)
= (232293103 - 232129013) / (5.455830 - 0.023172)
= 30204.3677 bytes per second

Exercise 2: TCP Connection Management

Consider the following TCP transaction between a client (10.9.16.201) and a server (10.99.6.175).

No	Source IP	Destination IP	Protocol	Info
295	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [SYN] Seq=2818463618 win=8192 MSS=1460
296	10.99.6.175	10.9.16.201	TCP	5000 > 50045 [SYN, ACK] Seq=1247095790 Ack=2818463619 win=262144 MSS=1460
297	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [ACK] Seq=2818463619 Ack=1247095791 win=65535
298	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [PSH, ACK] Seq=2818463619 Ack=1247095791 win=65535
301	10.99.6.175	10.9.16.201	TCP	5000 > 50045 [ACK] Seq=1247095791 Ack=2818463652 win=262096
302	10.99.6.175	10.9.16.201	TCP	5000 > 50045 [PSH, ACK] Seq=1247095791 Ack=2818463652 win=262144
303	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [ACK] Seq=2818463652 Ack=1247095831 win=65535
304	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [FIN, ACK] Seq=2818463652 Ack=1247095831 win=65535
305	10.99.6.175	10.9.16.201	TCP	5000 > 50045 [FIN, ACK] Seq=1247095831 Ack=2818463652 win=262144
306	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [ACK] Seq=2818463652 Ack=1247095832 win=65535
308	10.99.6.175	10.9.16.201	TCP	5000 > 50045 [ACK] Seq=1247095831 Ack=2818463653 win=262144

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?

- Seq=2818463618

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?

- Seq=1247095790
- Ack=2818463619
- The Ack# is essentially the last successfully acknowledged sequence number + length of data

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?

- Seq=2818463619
- The last ACK in response to the SYNACK doesn't contain any 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 client and server has done the active close. This is evident as the client retransmits the same seq with server acknowledging that same sequence. We also see two consecutive FINACKs, one from the server and one from the client.

- Closure Type: 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?

The data transferred from the client and the server is essentially the difference between the final ACK received and the initial sequence number.

Total Data Transferred = Final ACK received – Initial Sequence Number
= 2818463652 – 2818463619 = 33 bytes