**1. What is the IP address and TCP port number used by the client computer (source) that is transferring the alice.txt file to gaia.cs.umass.edu? To answer this question, it’s probably easiest to select an HTTP message and explore the details of the TCP packet used to carry this HTTP message, using the “details of the selected packet header window” (refer to Figure 2 in the “Getting Started with Wireshark” Lab if you’re uncertain about the Wireshark windows).**

* IP Address: 10.125.154.7
* TCP Port: 56951

**2. What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection?**

* IP Address: 128.119.245.12
* TCP Port: 80

**3. What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu? (Note: this is the “raw” sequence number carried in the TCP segment itself; it is NOT the packet # in the “No.” column in the Wireshark window. Remember there is no such thing as a “packet number” in TCP or UDP; as you know, there are sequence numbers in TCP and that’s what we’re after here. Also note that this is not the relative sequence number with respect to the starting sequence number of this TCP session.). What is it in this TCP segment that identifies the segment as a SYN segment? Will the TCP receiver in this session be able to use Selective Acknowledgments (allowing TCP to function a bit more like a “selective repeat” receiver, see section 3.4.5 in the text)?**

* Sequence Number (raw): 1130696858
* The SYN flag in the TCP header field identifies the segment as SYN
* The TCP receiver would not beable to use selective acknowledge

**4. What is the sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN? What is it in the segment that identifies the segment as a SYNACK segment? What is the value of the Acknowledgement field in the SYNACK segment? How did gaia.cs.umass.edu determine that value?**

* Sequence Number (SYN): 0
* Sequence Number (SYN, ACK): 1
* The segment is identified as a SYNACK segmented by the TCP flags. The value of the SYNACK segment is set to 1 due to the “Ack=1” field in the TCP header

**5. What is the sequence number of the TCP segment containing the header of the HTTP POST command? Note that in order to find the POST message header, you’ll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with the ASCII text “POST” within its DATA field4,5. How many bytes of data are contained in the payload (data) field of this TCP segment? Did all of the data in the transferred file alice.txt fit into this single segment?**

* Bites of data: 152319 bytes
* The data is split across multiple packets

**6. Consider the TCP segment containing the HTTP “POST” as the first segment in the data transfer part of the TCP connection. At what time was the first segment (the one containing the HTTP POST) in the data-transfer part of the TCP connection sent? At what time was the ACK for this first data-containing segment received? What is the RTT for this first data-containing segment? What is the RTT value the second data-carrying TCP segment and its ACK? What is the EstimatedRTT value (see Section 3.5.3, in the text) after the ACK for the second data-carrying segment is received? Assume that in making this calculation after the received of the ACK for the second segment, that the initial value of EstimatedRTT is equal to the measured RTT for the first segment, and then is computed using the EstimatedRTT equation on page 242, and a value of = 0.125. Note: Wireshark has a nice feature that allows you to plot the RTT for each of the TCP segments sent. Select a TCP segment in the “listing of captured packets” window that is being sent from the client to the gaia.cs.umass.edu server. Then select: Statistics->TCP Stream Graph- >Round Trip Time Graph.**

* First segment containing HTTP POST data was sent at: 12.848092 seconds
* The first segment containing the HTTP POST data was sent at: 12.848092 seconds.
* The ACK for this first data-containing segment was received at 12.959720 seconds.
* The Round Trip Time (RTT): 12.959720 - 12.848092 = 0.111628 seconds.
* For the second data-carrying TCP segment and its ACK: 12.959720 seconds
* The ACK for the second segment was received: 13.014375 seconds.
* The RTT for the second segment:13.014375 - 12.959720 = 0.054655 seconds.
* The EstimatedRTT value: α (alpha) value of 0.125.
* EstimatedRTT is the RTT for the first segment: = (1 - 0.125) \* 0.111628 + 0.125 \* 0.054655 = 0.0965795 seconds.

**7. What is the length (header plus payload) of each of the first four data-carrying TCP segments?**

The length (header plus payload) of each of the first four data-carrying TCP segments are as follows:

* Segment 1: 51487 bytes
* Segment 2: 12438 bytes
* Segment 3: 1382 bytes
* Segment 4: 23494 bytes

**8. What is the minimum amount of available buffer space advertised to the client by gaia.cs.umass.edu among these first four data-carrying TCP segments? Does the lack of receiver buffer space ever throttle the sender for these first four data-carrying segments?**

* The minimum amount of available buffer space advertised to the client by gaia.cs.umass.edu among the first four data-carrying TCP segments can be determined by examining the TCP window size field in the ACK packets. The lack of receiver buffer space may throttle the sender if the advertised window size becomes zero or too small.

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

* To identify retransmitted segments in the trace file, we look for duplicated segments or segments with the same sequence number but different acknowledgment numbers. We can use Wireshark's "tcp.analysis.retransmission" filter to isolate retransmitted segments and check for the retransmission flag (RST) in TCP flags.

**10. How much data does the receiver typically acknowledge in an ACK among the first ten data-carrying segments sent from the client to gaia.cs.umass.edu? Can you identify cases where the receiver is ACKing every other received segment (see Table 3.2 in the text) among these first ten data-carrying segments?**

* To determine the amount of data the receiver typically acknowledges in an ACK among the first ten data-carrying segments, we examine the acknowledgment numbers in the ACK packets. If the receiver ACKs every other received segment, we observe a pattern where the difference between consecutive acknowledgment numbers alternates between the size of one segment and the size of two segments.

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

* To calculate the throughput for the TCP connection, we measure the total bytes transferred during the connection using Wireshark's "tcp.analysis.bytes\_in\_flight" filter. We determine the duration of the connection from the timestamps of the first and last packets. Then, we divide the total bytes transferred by the duration to obtain the throughput.

**12 - 14. Use the Time-Sequence-Graph(Stevens) plotting tool to view the sequence number versus time plot of segments being sent from the client to the gaia.cs.umass.edu server. Consider the “fleets” of packets sent around t = 0.025, t = 0.053, t = 0.082 and t = 0.1. Comment on whether this looks as if TCP is in its slow start phase, congestion avoidance phase or some other phase. Figure 6 shows a slightly different view of this data.**

* It seems as if the TCP is in a slow starting phase, after 2.975 seconds the chart spikes up vertically.
* The behavior indicated the segments may not be being sent at regular individuals but a burst of activity instead. This could be due to network congestion, I am using UNT wifi which could be a factor in the periodicity.