Lab2

> Ethan Fison and Zan Montieth

[20 points] (a) [14 points] Work through the “Wireshark Lab: TCP” and answer all questions. For question 5 in the lab, make sure you list absolute, not relative sequence numbers (b) [3 points] What implementation of TCP is being used by your computer? How do you know? Submit your answers as wireshark tcp.txt or wireshark tcp.pdf

- wireshark lab: TCP from the book.

    1. 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?

        - IP address: 192.168.1.31
        - TCP port number: 50350

    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 number: 80

    3. What is the IP address and TCP port number used by your client computer (source) to transfer the file to gaia.cs.umass.edu

        - IP address: 192.168.1.31
        - TCP port number: 50350

    4. 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? What is it in the segment that identifies the segment as a SYN segment?

        - the sequence number is 0 in the trace, this initiates the TCP connection.
        - We no that it is a SYN message because the SYN flag is set to 0.

    5. 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 the value of the Acknowledgement field in the SYNACK segment? How did gaia.cs.umass.edu determine that value? What is it in the segment that identifies the segment as a SYNACK segment? make sure you list absolute,not relative sequence numbers.

        - The sequence number is 0
        - the acknkowledment number is 1
        - Gaia added 1 to our sequence number and sent it back as the acknowledgment number
        - both Syn and Ack are set to 1

6. 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.

    - The segment number is 4

7. 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)? 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 Section 3.5.3, page 242 in text) after the receipt of each ACK? Assume that the value of the EstimatedRTT is equal to the measured RTT for the first segment, and then is computed using the EstimatedRTT equation on page 242 for all subsequent segments. 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.

    - Segment 1 has sequence number 1
    - Segment 2 has sequence number 694
    - Segmemt 3 has sequence number 3614
    - Segment 4 has sequence number 5074
    - Segment 5 has sequence number 6534
    - Segment 6 has sequence number 7994

	Sent Time	**Ack Time**	**RTT**
Segment 1	36.162438	0.126155	0.1224292
Segment 2	36.163019	0.126157	0.1224292
Segment 3	36.163021	0.126158	0.1224292
Segment 4	36.163022	0.126163	0.1224292
Segment 5	36.163027	0.126164	0.1224292
Segment 6	36.163028	0.126165	0.1224292

    - estimated RTT = 0.875 * EstimatedRTT + 0.125 * SampleRTT

Segment	**EstimatedRTT**	**equation**
1	0.01530365	0.125 * 0.1224292
2	0.028702194	0.875 * 0.01530365 + 0.125 * 0.122492
3	0.04042592	0.875 * 0.028702194 + 0.125 * 0.122492
4	0.05068418	0.875 * 0.04042592 + 0.125 * 0.122492
5	0.059660157	0.875 * 0.05068418 + 0.125 * 0.122492
6	0.067514137	0.875 * 0.059660157 + 0.125 * 0.122492

8. What is the length of each of the first six TCP segments?
    - The first one is 693 bytes
    - the other 5 are 1460 bytes
9. What is the minimum amount of available buffer space advertised at the received for the entire trace? Does the lack of receiver buffer space ever throttle the sender?
    - the minimum amount of availbale buffer space advertised is 5840 bytes
    - The sender is never throttled because our reciever window kept growing.
10. Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question?
    - There were no retransmitted packets
    - We know this because the sequence number never skips a number
11. How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment (see Table 3.2 on page 250 in the text).
    - it typically acknowledges 1460 Bytes
    - yes, sometime the reciever may ACK every other segement causing a different number to be ACK
12. What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.
    - the throughput for the TCP connection is 125KByte/sec
    - we found this using the throughput graph in wireshark
13. 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. Can you identify where TCP's slowstart phase begins and ends, and where congestion avoidance takes over? Comment on ways in which the measured data differs from the idealized behavior of TCP that we've studied in the text.
    - the slowstart phase starts when the HTTP segment is sent
    - we never exited the slowstart phase because the file was not large enough and the server was not sending fast enough for us to exit it.
    - the idealized behavior of TCP expects the sender to send the packet with no limitations using their full bandwidth. it also expects large file sizes. In this case it was a relatively small file and we can assume the sender didn't use their full bandwidth
14. Answer each of two questions above for the trace that you have gathered when you transferred a file from your computer to gaia.cs.umass.edu
    - We noticed the same behavior when sending a file back.

- What implementation of TCP is being used by your computer? How do you know?
    - The computer we used was using internet protocol Version 4, we found this by going into our internet settings and looking there.