CSC358H5: Principles of Computer Networking — Winter 2025

Wireshark Lab 3: TCP

Due Date: Friday, April 4, 11:59 PM

1 What is TCP?

The Transmission Control Protocol (TCP) is used to transfer data reliably across networks. Unlike the User Datagram Protocol (UDP), which doesn't require a connection, TCP establishes a connection between a client and a server, ensuring that no data is lost and arrives in the correct order. As a result, TCP is typically used for applications such as web browsing, emails, and file transfers.

1.1 Three-way handshake

The TCP protocol first carries out a connection process known as the three-way handshake. The three-way handshake has the following steps:

1. **SYN (Synchronize)**: The client sends a SYN packet to the server. This packet contains a random sequence number (*e.g.*, Seq = X) that the client will use to track the data sent.
2. **SYN-ACK (Synchronize-Acknowledgment)**: Once the server receives the SYN packet, the server responds with a SYN-ACK packet. This packet acknowledges the client's SYN by setting the acknowledgment number to X + 1 and includes its own random sequence number (*e.g.*, Seq = Y).
3. **ACK (Acknowledgment)**: Finally, the client sends an ACK packet to the server, acknowledging the server's SYN-ACK by setting the acknowledgment number to Y + 1. At this point, the connection is established, and data transfer can begin.

2 Gathering a Packet Trace and Analyzing The Capture

This section is not required for you to complete the lab, but is encouraged for learning purposes. This sections will obtain a packet trace of a TCP file transfer. The file will be transferred using the HTTP POST method.

2.1 Gathering a TCP Packet Trace

1. Download `alice.txt` from: `http://gaia.cs.umass.edu/wireshark-labs/alice.txt` and save it as a `.txt` file.
2. Visit `http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html`.
3. Use the "Browse" button to select your `alice.txt` file.
4. Start Wireshark and begin packet capture.
5. Click "Upload alice.txt file" and wait for the success message.
6. Stop Wireshark capture and save the trace.

2.2 Analyzing the Capture

After filtering for the http packets the http POST request can be seen as in Figure 1.

- From the packet it can be seen that about 152kb of data was transferred which is too large for a single TCP segment. From Figure 1 the arrow indicates that the data was broken into 106 TCP segments.

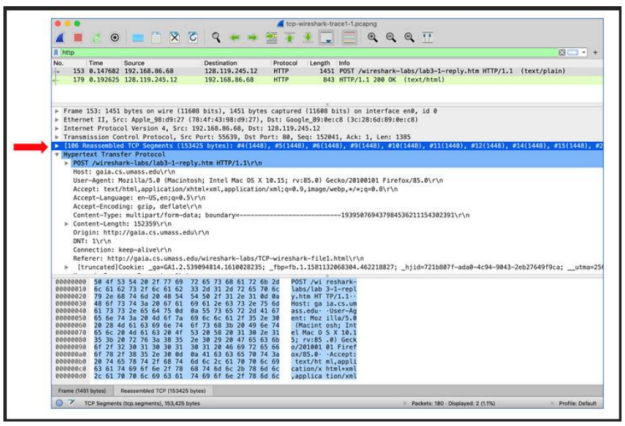

Figure 1: HTTP POST request

- These segments can get dropped or arrive in different orders which is what TCP handles in the Network Layer.

Now the packets can be filtered by tcp to further analyze the capture. As shown in Figure 2 the packets required in the TCP Handshake are also captured.

- The TCP SYN is sent to the server with Sequence number 0
- The TCP SYNACK is returned from gaia which has an ACK number one more than the sequence number. Additionally it has its own sequence number which also happens to be 0.
- Finally the ACK is sent to the server which has an ACK number of 1 (1 more than the sequence number of SYNACK)

Once this handshake is completed it can be seen that 1448 bytes of data is sent from the next TCP packet. Once these bytes are sent the next 1448 bytes are sent with sequence number 1449 (Packet 5). Additionally, the server responds back with the ACK packets once it has received the packet for instance packet 7 is an ACK packet is ACK value of 1449 indicating it has received 1448 bytes.

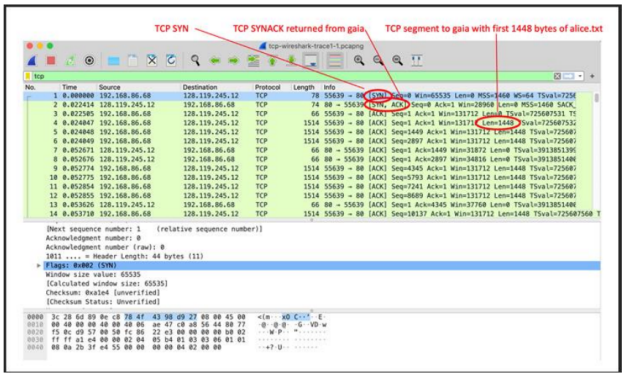

Figure 2: Three way handshake packets

3 TCP Questions

Download tcp-wireshark-trace1-1.pcapng from Quercus and answer the following questions.

Q1. 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." You may want to refer to the "Introduction to Wireshark" document if you are uncertain about the Wireshark windows.

Q2. What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection? Ensure that you are filtering by tcp so that the series of TCP segments sent between your computer and gaia.cs.umass.edu can be analyzed.

Q3. 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 this TCP segment that identifies the segment as a SYN segment? Write the field name. 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)? Answer with Yes or No.
[**NOTE:** This is the "raw" sequence number carried in the TCP segment itself; it is NOT the packet number in the "No." column in the Wireshark window. Remember there is no such thing as a "packet number" in TCP or UDP. Also note that this is not the relative sequence number with respect to the starting sequence number of this TCP session.]

Q4. 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? Write the field name. What is the value of the Acknowledgment number in the SYNACK segment?

Q5. What is the sequence number of the TCP segment containing the header of the HTTP POST command?
[**NOTE:** 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 field.]

Q6. Consider the TCP segment containing the HTTP "POST" as the first segment in the data transfer part of the TCP connection.

- **6.a.** At what (relative) time was the first segment (the one containing the HTTP POST) in the data-transfer part of the TCP connection sent?
- **6.b.** At what (relative) time was the ACK for this first data-containing segment received?
- **6.c.** What is the RTT for this first data-containing segment?
- **6.d.** What is the RTT value for the second data-carrying TCP segment and its ACK?
- **6.e.** Using the following function, calculate the EstimatedRTT value after the ACK for the second data-carrying segment is received.

$$\text{EstimatedRTT} = (1 - \alpha) \cdot \text{EstimatedRTT} + \alpha \cdot \text{SampleRTT}.$$

In making this calculation, assume that after receiving the ACK for the second segment the initial value of EstimatedRTT is equal to the measured RTT for the first segment. Furthermore, assume that $\alpha = 0.125$.
[**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.]

Q7. What is the header length of each of the first four data-carrying TCP segments? What is the payload length?
[**NOTE:** he TCP segments in the tcp-wireshark-trace1-1 trace file are all less than 1480 bytes. This is because the computer on which the trace was gathered has an interface card that limits the length of the maximum IP datagram to 1500 bytes, and there is a minimum of 40 bytes of TCP/IP header data. This 1500-byte value is a fairly typical maximum length for an Internet IP datagram.]

Q8. 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?

Q9. Does there exist any retransmitted segment in the trace file? [**HINT:** Use TCP statistics in Wireshark instead of manually looking through the packets].

Q10. How much data (in bytes) does the receiver typically acknowledge in an ACK among the first ten data-carrying segments sent from the client to gaia.cs.umass.edu? Within the first fifteen carrying data segments, there are two times where the receiver is ACKing every other received segment (instead of every single segment). List (in non-descending order) the ACK numbers for these two packets.

Q11. (Not for credit) What is the throughput (bytes transferred per unit time) for the TCP connection? 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."

4 TCP Congestion Control in Action

Now we examine the amount of data sent per unit time from the client to the server. We'll use one of Wireshark's TCP graphing utilities **Time-Sequence-Graph(Stevens)** to plot out data.

- Select a client-sent TCP segment in the Wireshark's "listing of captured-packets" window corresponding to the transfer of alice.txt from the client to gaia.cs.umass.edu.
- Then select the menu: Statistics→TCP Stream Graph→ Time-Sequence-Graph(Stevens).
- You should see a plot that looks similar to the plot in Figure 3

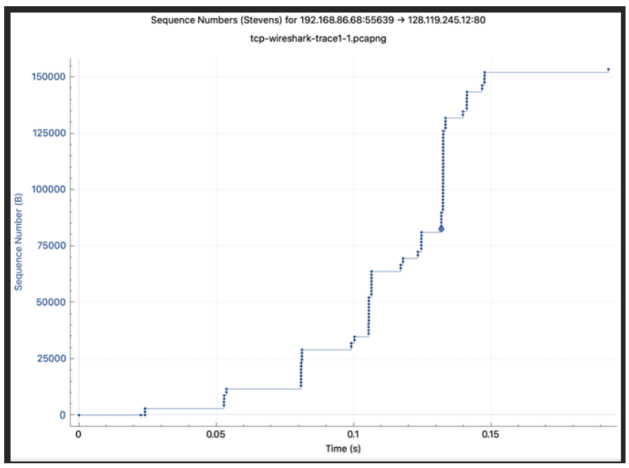

Figure 3: A sequence-number-versus-time plot (Stevens format) of TCP segments.

Here, each dot represents a TCP segment sent, plotting the sequence number of the segment versus the time at which it was sent. Note that a set of dots stacked above each other represents a series of packets (sometimes called a "fleet" of packets) that were sent back-to-back by the sender.

Q12. (not for credit) 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.

Q13. (not for credit) These "fleets" of segments appear to have some periodicity. What can you say about the period?

Q14. (not for credit) 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