National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Department of Electrical Engineering

Faculty Member: Sir Hassan khaliq **Dated: 4/26/2023**

Semester:6th **Section: C**

EE-357 Computer and Communication Networks

Wireshark – TCP (Transmission Control Protocol)

		PLO5/ CLO3		PLO5/ CLO3	PLO5/ CLO3	PLO5/ CLO3
Name	**Reg. No**	**Viva / Quiz / Lab Performance** **5 Marks**	**Analysis of data in Lab Report** **5 Marks**	**Modern Tool Usage** **5 Marks**	**Ethics and Safety** **5 Marks**	**Individual and Team Work** **5 Marks**
Imran Haider	**332569**					
Muhammad Ahmed Mohsin	**333060**					
Amina Bashir	**343489**					

National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

1 Table of Contents

National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Wireshark – TCP (Transmission Control Protocol)

2 OBJECTIVE OF THIS LAB:

In this lab, we will explore several aspects of the Transmission Control Protocol.

3 INSTRUCTIONS:

- Read carefully before starting the lab.
- These exercises are to be done individually.
- You are supposed to provide the answers to the in-line questions in this document and upload the completed document to your course's LMS site.
- **For all questions, you must not only answer the question, but also supply all necessary information regarding how you arrived at the answer (e.g., use screenshots/ accompanying text, etc.) Use red font color to distinguish your replies from the rest of the text.**
- Avoid plagiarism by copying from the Internet or from your peers. You may refer to source/ text but you must paraphrase the original work.

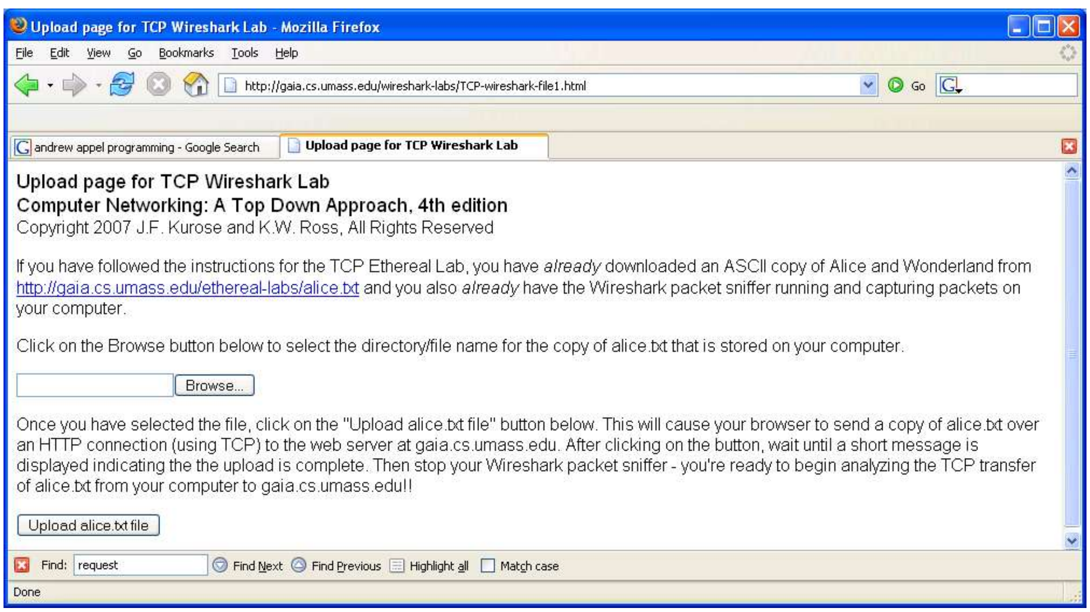
4 BACKGROUND:

In this lab, we will investigate the behavior of the celebrated TCP protocol in detail. We will do so by analyzing a trace of the TCP segments sent and received in transferring a 150KB file (containing a text file) from your computer to a remote server. We will study TCP's use of sequence and acknowledgement numbers for providing reliable data transfer; we will see TCP's congestion control algorithm – slow start and congestion avoidance – in action; and we will look at TCP's receiver-advertised flow control mechanism. We will also briefly consider TCP connection setup and we will investigate the performance (throughput and round-trip time) of the TCP connection between your computer and the server.

- Capturing a bulk TCP transfer from your computer to a remote server
- Before beginning our exploration of TCP, we will need to use Wireshark to obtain a packet trace of the TCP transfer of a file from your computer to a remote server. You will do so by accessing a Web page that will allow you

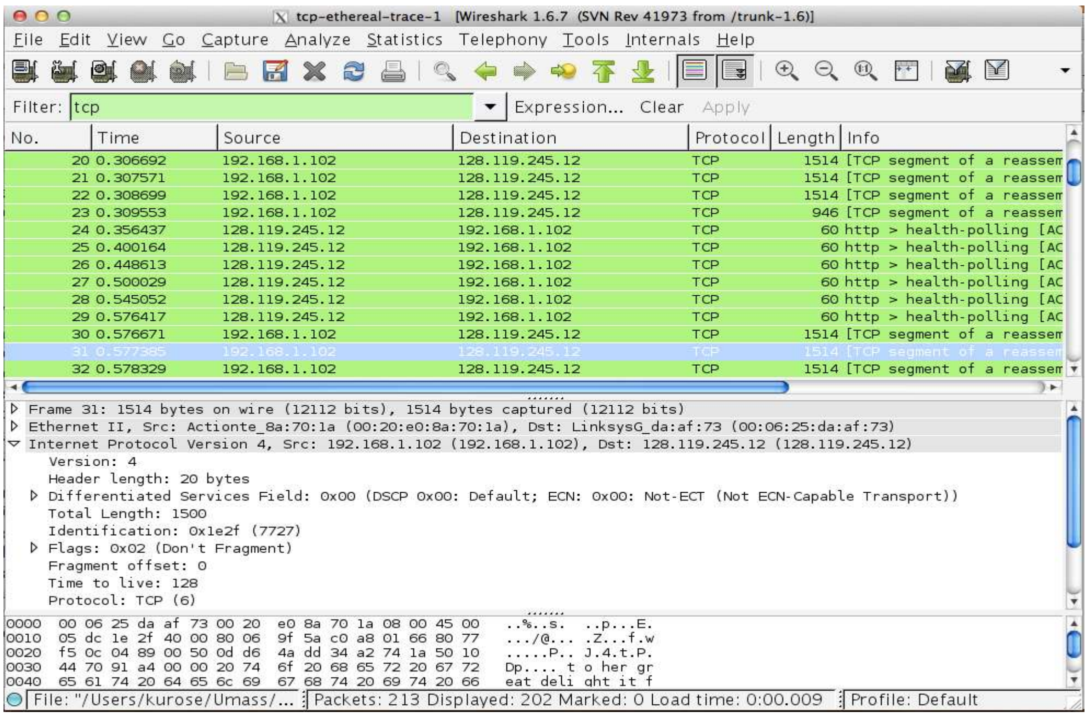
to enter the name of a file stored on your computer (which contains an ASCII text), and then transfer the file to a Web server using the HTTP POST method. We are using the POST method rather than the GET method as we would like to transfer a large amount of data from your computer to another computer. Of course, we will be running Wireshark during this time to obtain the trace of the TCP segments sent and received from your computer.

- Do the following:
- Start up your web browser. Go the http://gaia.cs.umass.edu/wireshark-labs/alice.txt and retrieve an ASCII copy of *Alice in Wonderland.* Store this file somewhere on your computer.
- Next, go to http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html
- You should see a screen that looks like:



- Use the *Browse* button in this form to enter the name of the text file (full path name) on your computer (or do so manually). Do not yet press the Upload button.
- Now, startup Wireshark and begin packet capture *(Capture->Start)*
- Returning to your browser, press the Upload button to upload the file to the server. Once the file has been uploaded, a short congratulations message will be displayed in your browser window.

- Stop Wireshark packet capture. Your Wireshark window should look similar to the window shown below.



5 A FIRST LOOK AT THE CAPTURED TRACE

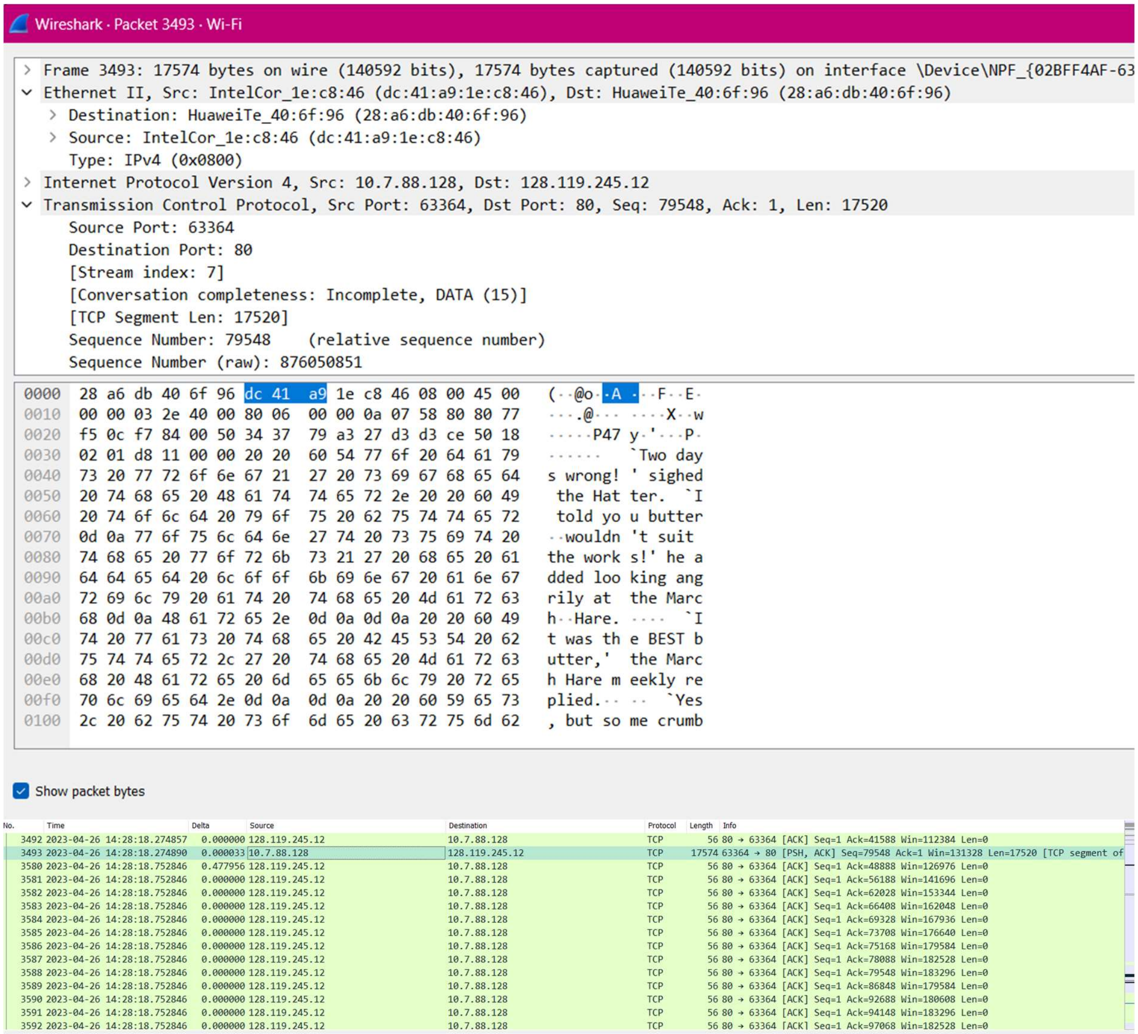
Before analyzing the behavior of the TCP connection in detail, you should take a high level view of the trace.

- First, filter the packets displayed in the Wireshark window by entering “tcp” (lowercase, no quotes, and don’t forget to press return after entering!) into the display filter specification window towards the top of the Wireshark window.

What you should see is series of TCP and HTTP messages between your computer and gaia.cs.umass.edu. You should see the initial three-way handshake containing a SYN message. You should see an HTTP POST message. Depending on the version of Wireshark you are using, you might see a series of "HTTP Continuation" messages being sent from your computer to gaia.cs.umass.edu. There is actually no such thing as an HTTP Continuation message – this is Wireshark's way of indicating that there are multiple TCP segments being used to carry a single HTTP message. In more recent versions of Wireshark, You will see "[TCP segment of a reassembled PDU]" in the Info column of the Wireshark display to indicate that this TCP segment contained data that belonged to an upper layer protocol message (in our case here, HTTP). You should also see TCP ACK segments being returned from **gaia.cs.umass.edu** to your computer.

Answer the following questions, by opening the Wireshark captured packet file ***tcp-ethereal-trace-1***. Whenever possible, when answering a question, you should hand in a screenshot of the packet(s) within the trace that you used to answer the question asked. Annotate the screenshot to explain your answer. To print a packet, use *File->Print*, choose *selected packet only*, choose *Packet summary line,* and select the minimum amount of packet detail that you need to answer the question.

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? 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.**



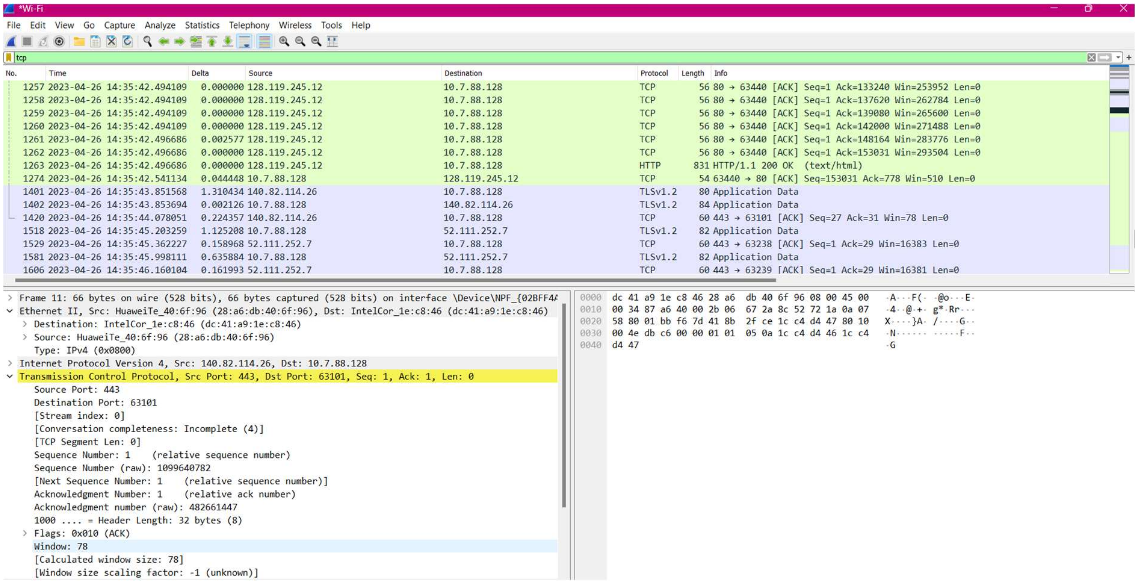
According to above figure, the client computer (source)'s IP address is 10.7.88.128 and the TPC port number is 63364.

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**?

According to above figure, the IP address of gaia.cs.umass.edu is 128.119.245.12 and the TCP port number is 80.

Now, create your own trace and answer the following questions:

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?**



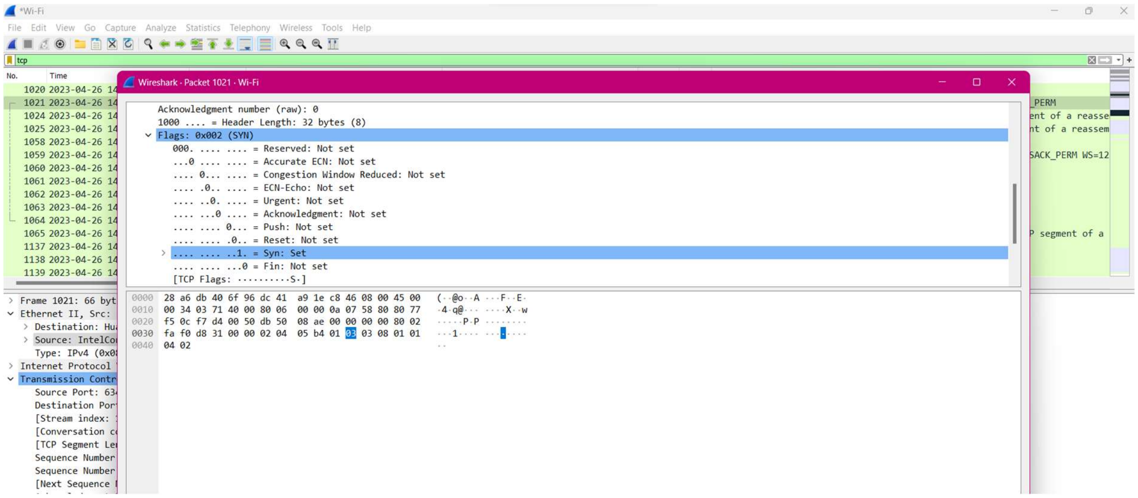
The TCP port number is 443 and the source IP address is 192.82.114.26.

Since this lab is about TCP rather than HTTP, you need to change Wireshark's "listing of captured packets" window so that it shows information about the TCP segments containing the HTTP messages, rather than about the HTTP messages. To have Wireshark do this, select *Analyze->Enabled Protocols.* Then uncheck the HTTP box and select *OK*. You should now see a Wireshark window that looks like:

6 TCP BASICS

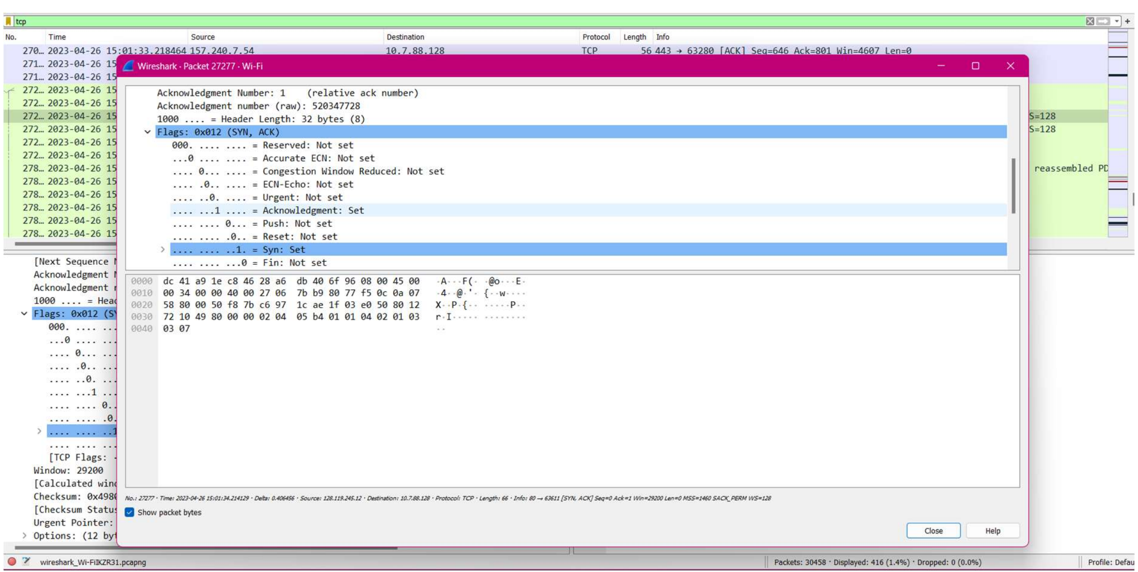
Answer the following questions for the TCP segments:

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?**



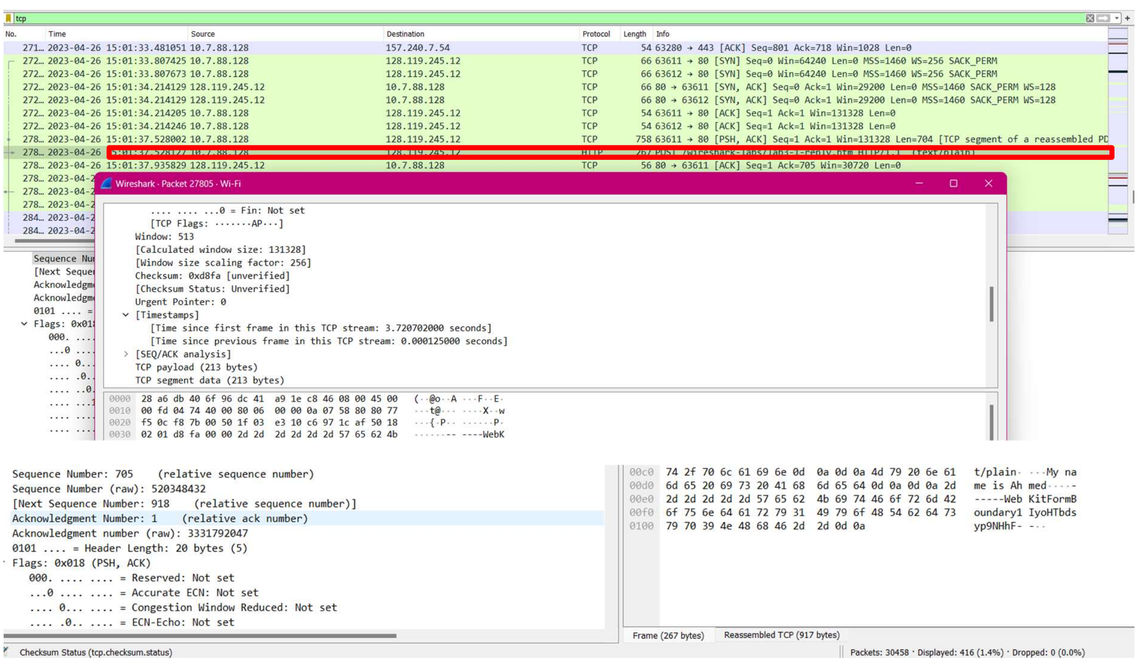
The sequence number of the TCP SYN segment is 0 since it is used to imitate the TCP connection between the client computer and gaia.cs.umass.edu. According to above figure, in the Flags section, the Syn flag is set to 1 which indicates that this segment is a SYN segment.

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?**



According to the above figure, the sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN is 0. The value of the acknowledgement field in the SYNACK segment is 1. The value of the ACKnowledgement field in the SYNACK segment is determined by the server gaia.cs.umass.edu. The server adds 1 to the initial sequence number of SYN segment form the client computer. For this case, the initial sequence number of SYN segment from the client computer is 0, thus the value of the ACKnowledgement field in the SYNACK segment is 1. A segment will be identified as a SYNACK segment if both SYN flag and Acknowledgement in the segment 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 will 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.**



Hence, we can observe that it is 705th in sequence number.

7 CONCLUSION

In this lab report, I performed tracing of TCP elements and then worked inside the segments. I used the tcpdump command to capture TCP traffic and then used the Wireshark network analyzer to view the captured traffic. I was able to identify the different TCP elements in the captured traffic, including the source and destination IP addresses, the source and destination port numbers, the sequence numbers, and the acknowledgment numbers. I was also able to view the data that was being transmitted in the TCP segments.

I then worked inside the TCP segments to view the different fields that are present in each segment. I was able to identify the different fields, such as the source and destination IP addresses, the source and destination port numbers, the sequence numbers, the acknowledgment numbers, the window size, the checksum, and the urgent pointer. I was also able to view the data that was being transmitted in the TCP segments.

This lab report gave me a good understanding of how TCP works and how to trace TCP traffic. I was able to identify the different TCP elements in the captured traffic and view the data that was being transmitted in the TCP segments. This knowledge will be useful for me in my future studies and work.