CIS 457: Data Communications

Week 12 Assignment: Working with TCP Protocol and TCP's Retransmission Timer Setup
Point Value: **10** Pts, late submission policy (20%/day for up to 5 days)
Due Date: 4/5/2024 by 12:00 AM EST
Submission format:

The following reports will be submitted on Blackboard (BB) using each group member's account.
(1) The entire assignment's solution in a PDF format (your contribution to the assignment)
(2) Group's Agreed-upon solution in a PDF format.
Note: Missing an individual's contribution leads to receiving **NO** credit for the assignment.

Lab Objectives

In this lab:

- We'll investigate the process of establishing a TCP connection.
- We'll study TCP's use of *sequence* and *acknowledgment* numbers to provide reliable data transfer.
- We'll investigate how the retransmission timer's timeout value is set to provide reliable data transfer.

Before beginning this lab, you should review the TCP protocol in the text and your lecture notes.

Part I: TCP Basics

1. Capturing a bulk TCP transfer from your computer to a remote server. Before beginning our exploration of TCP, we'll need to use Wireshark to obtain a packet trace of the TCP transfer of a file from your computer to a remote server. You'll do so by accessing a Web server that will allow you to enter the name of a file stored on your computer (which contains the ASCII text of Alice in Wonderland), and then transfer the file to a Web server using the HTTP POST method. We're using the POST method rather than the GET method as we'd like to transfer a large amount of data from your computer to another computer. Of course, we'll be running Wireshark during this time to obtain the trace of the TCP segments sent and received from your computer.

2. **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
   - Use the Browse button in this form to enter the name of the file (full path name) on your computer containing Alice in Wonderland (or do so manually). **Don't** yet press the "Upload alice.txt file" button.
   - Now **start up** **Wireshark** and begin packet capture.
   - Returning to your browser, press the "Upload alice.txt file" button to upload the file to the gaia.cs.umass.edu server. Once the file has been uploaded, a short congratulations message will be displayed in your browser window.
   - Stop Wireshark packet capture. **Save** this trace in your computer with a name **TCP-mine**.

Let us take a first look at the captured trace. Before analyzing the behavior of the TCP connection in detail, let's take a high-level view of the trace. First, filter the packets displayed in the Wireshark window by entering "tcp". 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.

Answer the following questions based on your captured Wireshark trace:

1. **What is the IP address and TCP port number used by your client computer (source) that is transferring the file to gaia.cs.umass.edu? (0.5Pt)**
2. **What is the IP address of** gaia.cs.umass.edu**? (0.5Pt)**
3. **On what port number is it sending and receiving TCP segments for this connection? (0.5Pt)**

Recall from our discussion in the earlier HTTP protocol, that is 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'll 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.

In the following section of the lab, we're looking for a series of TCP segments sent between the sending computer and gaia.cs.umass.edu. For this section, in order for me to maintain consistency among all students' answers, **all students will use a common Wireshark capture file that I obtained after uploading the same text file to the gaia.cs.umass.edu's web server. My machine has an ip address of 192.168.1.102**. So, please use the Wireshark packet file **TCP-1** posted on Blackboard to 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 **(use Wireshark's interpretation of the sequence number)? What is it in the segment that identifies it as an SYN segment? (1Pt)**

5. **What is the sequence number of the SYN-ACK segment sent by** gaia.cs.umass.edu **to the client computer in reply to the SYN? What is the value of the Acknowledgement number in the SYN-ACK segment? (use Wireshark's interpretation of the sequence number) What is it in the segment that identifies it as a SYN-ACK segment? (2Pts)**

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 using the Wireshark bottom window, looking for a http message that contains a TCP segment payload with a "POST" is listed within its DATA field. (1Pt)**

7. **Use Jacobson algorithm to solve this problem. Consider the TCP segment containing the HTTP POST in this TCP connection.**
   What are the sequence numbers of the segments sent to carry the "alice.txt file" (including the segment containing the HTTP POST)? At what time were the first segment sent? When were the ACK for the first segment received? Given the difference between when each TCP segment was sent, and when its acknowledgement was received, what are the observed RTT value for the first segment? What is the EstimatedRTT value after the receipt of each ACK for the first and second segments? What is the length of the first TCP segment? (6Pts)

8. **Are there any retransmitted segments in the trace file? What did you check for in order to answer this question? (1Pt)**
9. **Approximately, what is the throughput for the TCP connection? Explain how you calculated this value. (2Pts)**

Please note that the TCP segments in the **tcp -1** trace file are all less than 1460 bytes. This is because my computer has an Ethernet card that limits the length of the maximum IP packet to 1500 bytes (40 bytes of TCP&IP header data and 1460 bytes of TCP payload/MSS). The 1500B is the standard maximum length that is allowed by Ethernet data link.

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.**

10. **What can you identify from the Time-Sequence-Graph (Stevens)? (1.5Pt)**

Reference:

- Computer Networks, A Top-down Approach, 8th ed., J.F. Kurose and K.W. Ross, Addison Wesley/Pearson.