Computer Networks CSE 4344
Project 2
Transmission Control Protocol Analysis using Wireshark

Instructor: **Sajib Datta**
GTA (Section 002): Jees Augustine

Fall 2018

"Tell me and I forget. Show me and I remember. Involve me and I understand." ***Chinese proverb***

Objectives

- To understand and analyze *Transmission Control Protocol (TCP)*.
- To get to know the *TCP*'s connection parameters and its behavior.
- To get familiar with capturing packet trace using Wireshark.

In this lab, we'll investigate the behavior of the celebrated TCP protocol in detail. We have provided you a video which can be used during the experiments. During the experiment you will upload the video from your computer to Youtube[1] and will collect the traces using Wireshark. Wireshark is available in this Link. We'll study TCP's use of sequence and acknowledgement numbers for providing reliable data transfer; we'll see TCP's congestion control algorithm – slow start and congestion avoidance – in action; and we'll look at TCP's receiver-advertised flow control mechanism. We'll also briefly consider TCP connection setup and we'll investigate the performance (throughput and round-trip time) of the TCP connection between your computer and **youtube.com**.

Before beginning this lab, you'll probably want to review sections 3.5 and 3.7 in the text.

Due Date

November 09, 2018 (Friday) 11:59 PM[2]

[1] https://www.youtube.com
[2] All Submissions should be completed through BlackBoard

Submission Guidelines

- Submit a single zipped file with the naming convention,

$$< your\_UTA\_id > \_ < your\_name > .zip$$

- Your submission should have the following items to be considered for evaluation,
  (a) The **PDF** document, that is the answer to the questions in the assignment [3]
  (b) The name of the solution document should be *your_name_ID_No.pdf*.
  (c) The document should have your name in the as header or footer in every page along with page nuumbers.
  (d) Include a screenshots of the packet(s) within the trace that you used to answer a question whenever possible. Highlight the relevant item(s) in the screenshot to explain your answer.
  (e) Your own captured trace file wherever its necessary.

- Make sure you write your **name** and your **UTA ID** in the final document that you are submitting.

- Make sure that submissions of the zipped file is through *UTA BlackBoard*[4].

- Late submission will be accepted only through email [5]. Address all emails to your Professor and add a copy to TA.

- There will be a reduction of 10 points for the first day and 5 pointers each for each subsequent day.

Section 1: HTTP over TCP

The following steps should be followed to complete the experiment.

1. Launch Wireshark and select the interface you want to capture from. A screenshot is provided as Figure 1 for assistance on how to do this. (Wi-Fi is recommended).

   [The images may vary based on the operating system and version of the wireshark you are using. The section details the essential procedure not the exact steps]

[3] Strictly this should be *.pdf* file as this enables us to read the answer in a way you want us to read it
[4] Please strictly follow the naming convention of the zipped file
[5] Please mention the subject line as *"CSE 5344 - Project 2"*

Figure 1: The start up screen of Wireshark with options to choose for the experiments. You are encouraged to use the wireless interface if you are using WiFi as a means to connect to Internet. We encourage you to take some time to get familiarized with Wireshark and its functionality, interfaces (graphical) and options as its crucial to performing the experiments. This is an excellent place to start with.

2. Now begin/start capturing the packets from the interface you've selected above.
3. Click start to begin capturing.
4. Start up your web browser.
5. Go to **uta.edu**
6. Stop packet capturing.

Observations

- What you should see a series of TCP and HTTP messages between your computer and uta.edu .
- You should see the initial three-way handshake containing a SYN 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 uta.edu.
- 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 uta.edu to your computer.

Problem Set 1

1. What is the IP address and TCP port number used by your client computer (source) to browse the page uta.edu.

   Use the 'GET' message to answer the following questions.

2. What is the TTL value that is used in this communication ?
3. Did you Use IPV4 or IPV6 for communication ?
4. Does your optional field has some particular information or not.
5. Is the Packet Fragmented
6. What is the TCP segment length ?
7. What is the Sequence Number of TCP segment *(you can use the relative sequence number).*
8. Calculate the *acknowledgement number* based on the the two questions above. Verify your solution with the *Wireshark* values.
9. What are the fields in the TCP Flags. *No need to give any values but give the field names given in Wireshark*
10. What is the IP address of uta.edu ? On what port number is it sending and receiving TCP segments for this connection?

Section 2 : Analysing the Connection Parameters in TCP

The following steps should be followed to complete the experiment.

1. Download the video *Inventions of 2014.mp4* put up on blackboard [6].
2. Login to your own **youtube.com** account.
3. Start capturing the packets at the wireless interface using Wireshark

[6] Courtesy: Internet

4. ***To avoid any copyright issues, you should make the video upload private. Disclaimer could be found in the footnote***[7]. Upload the video given to your youtube.com account

5. Stop capture.

Observations

- What you should see a series of TCP messages between your computer and youtube.com.
- You should see the initial three-way handshake containing a SYN 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 uta.edu.
- 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 youtube.com to your computer.

Problem Set 2

1. What is the sequence number (absolute) of the TCP SYN segment that is used to initiate the TCP connection between the client computer and youtube.com?
2. What is it in the segment that identifies the segment as a SYN segment?
3. What is the sequence number of the SYNACK segment sent by youtube.com to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment?
4. How did youtube.com determine that value? What is it in the segment that identifies the segment as a SYNACK segment?

[7] Disclaimer: Neither the instructor (along with Graduate Teaching Assistants) nor the University will be responsible for any legal copyright issues

Section 3 : Analysis of the trace provided

The following steps should be followed to complete the experiment.

1. We have provided you with a < *kayak.pcapng* > file along with the project which is a packet trace run to **kayak.com**.
2. Download the *kayak.pcapng* file from blackboard
3. Open up the *kayak.pcapng* file provided using wireshark and start analysing the packet.

Problem Set 3

1. What is the sequence number of the TCP segment containing the ***first*** HTTP POST command? [a]

2. Consider the TCP segment containing the HTTP POST as the first segment in the TCP connection. *i)* What are the sequence numbers of the first four segments in the TCP connection (including the segment containing the HTTP POST)? *ii)* At what time was each segment sent? *iii)* When was the ACK for each segment received? *iv)* Given the difference between when each TCP segment was sent, and when its acknowledgement was received, *v)* what is the RTT value for each of the four segments? *vi)* What is the EstimatedRTT value (see Section 3.5.3, page 239 in text) after the receipt of each ACK? *vii)* 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 239 for all subsequent segments[b].

3. What is the length of each of the first four TCP segments?

4. What is the minimum amount of available buffer space advertised at the receiver for the entire trace?

5. Does the lack of receiver buffer space ever throttle the sender?

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

7. How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACK-ing every other received segment (see Table 3.2 on page 247 in the text).

8. What is the throughput (bytes transferred per unit time) for the TCP connection (Just consider a single connection)? Think on how to calculate the throughput!

9. Explain how you calculated this value.

[a] 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.

[b] 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 **kayak.com**. Then select: Statistics→ TCP Stream Graph→ Round Trip Time Graph.

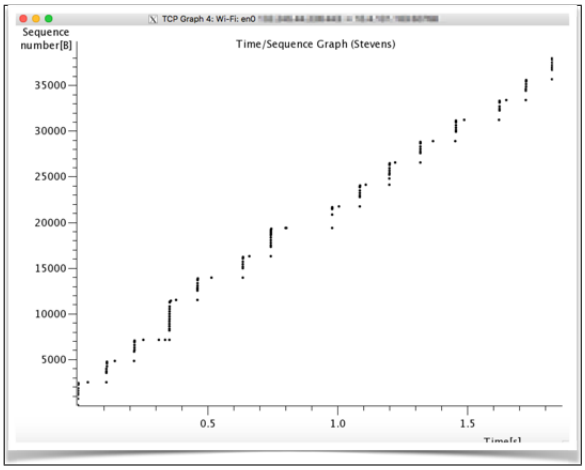

Figure 2: Here, each dot represents a TCP segment sent, plotting the sequence number of the segment versus the time at which it was sent. Note that a set of dots stacked above each other represents a series of packets that were sent back-to-back by the sender.

Section 4 : TCP congestion control in action

Now examine the amount of data sent per unit time from the client to the server. Rather than (tediously!) calculating this from the raw data in the Wireshark window, we'll use one of Wireshark's TCP graphing **utilities→Time-Sequence-Graph(Stevens)** to plot out data.

The following steps should be followed to complete the experiment.

1. Go to youtube.com
2. Locate the image which you have uploaded
3. Go to video manger of youtube.com.
4. You will have the option to download the video that you have uploaded earlier in experiment 2.
5. Start your packet capture with Wireshark
6. Start downloading the video as *.mp4* video
7. After the download stop the Wireshark capture.
8. Select a TCP segment in the Wireshark's "*listing of captured-packets*" window.
9. Select the menu: **Statistics→TCP Stream Graph→Time-Sequence-Graph(Stevens)**. You should see a plot that *looks similar* (Remember Figure 2 is a sample graph and you will have to conduct the experiment in your computer to obtain one. Add it's screenshot as well) to the following plot, which was created from the captured packets in the packet trace provided.

Answer the following questions for the TCP segments in the packet trace,

Problem Set 4:

Use the Time-Sequence-Graph(Stevens) plotting tool to view the sequence number versus time plot of segments being sent from **youtube.com** to your computer.
Answer each of three questions below for the trace that you have gathered when you transferred a file to your computer from youtube.com.

1. Can you identify where TCP's slow-start phase begins and ends.
2. Where congestion avoidance takes over? *Highlight these areas* .
3. Comment on ways in which the measured data differs from the idealized behavior of TCP that we've studied in the text.

Wish You All The Best