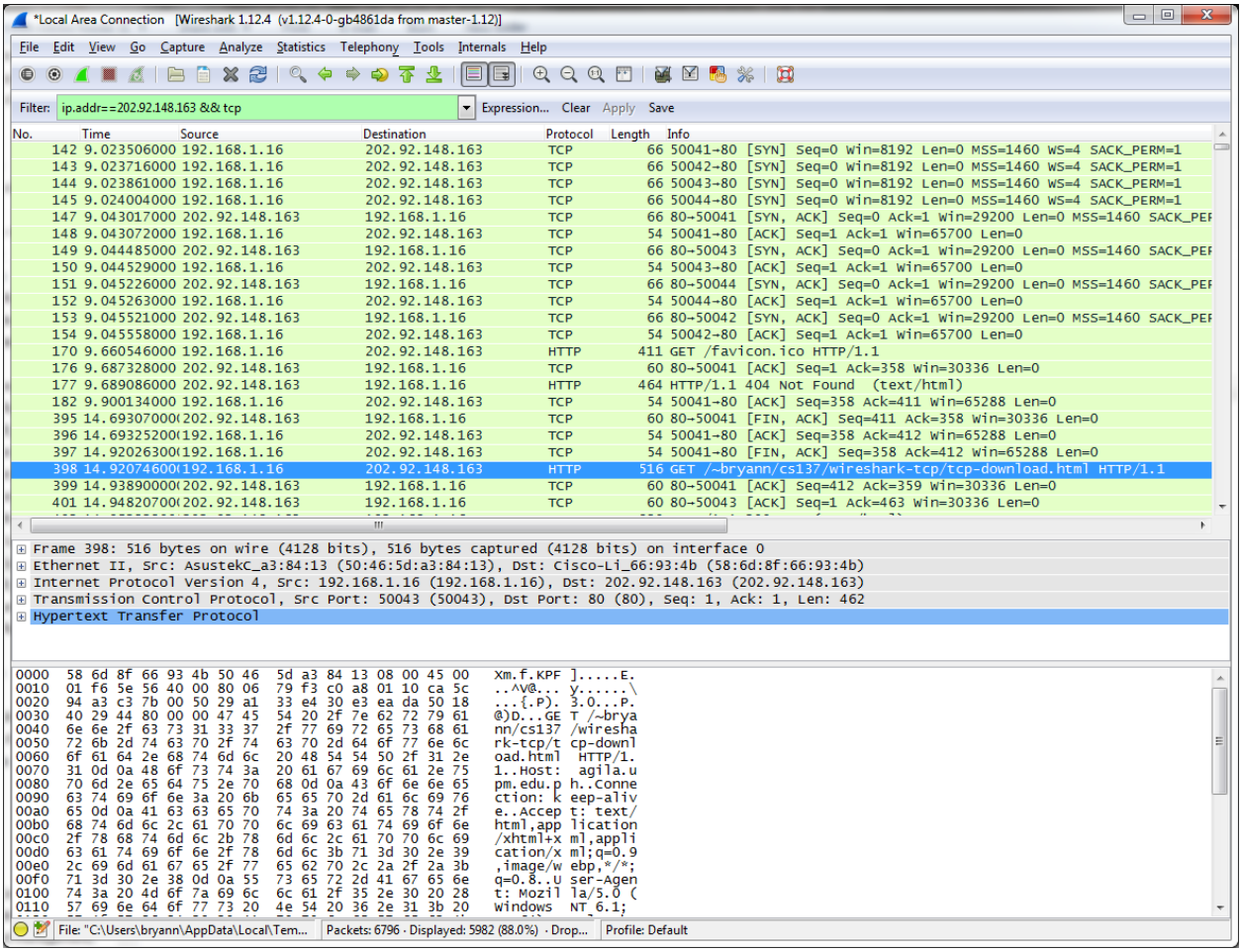
Wireshark Lab: TCP

In this lab, we will look into the TCP protocol in detail. We will be analyzing a trace of TCP segments send and received in transferring a file (WinSCP installer) from a remote server to your computer. We'll look at TCP's use of sequence and acknowledgement numbers for providing reliable data transfer. We will also look at TCP's receiver-advertised flow control mechanism. We will also consider TCP connection setup and investigate the performance (throughput and round-trip time) of the TCP connection between your lab computer and the server.

Capturing the bulk TCP transfer from your computer to a remote server

We will access a site that will allow us to download a big file.

- Start up Wireshark and begin packet capture.
- Go to http://agila.upm.edu.ph/~bryann/cs138/wireshark-tcp/tcp-download.html and click on the link to download the WinSCP installer. Note that you are not going to install the WinSCP installer for this activity.
- After the download has completed, stop Wireshark packet capture. Your Wireshark window should look similar to the window shown below.



A first look at the captured trace

Filter the packets displayed in Wireshark by entering "ip.addr==202.92.148.163 && tcp" into the filter specification to list only the TCP segments sent to and from agila.upm.edu.ph. What you will be looking at is the series of TCP and HTTP messages between your computer and agila.upm.edu.ph. You should see the initial three-way handshake containing a SYN message.

1. Place the captured three-way handshake packets with their corresponding TCP fields displayed.

You should also see the HTTP response message of agila to your computer after you sent a GET request message for the installer. If you look at the detail of this packet, you will see a "Reassembled TCP Segments" part. This is actually not part of that packet but is displayed by Wireshark to indicate that there are multiple TCP segments being used to carry a single HTTP message. The different segments of this message should be above the message with "[TCP segment of a reassembled PDU]" in their info column. You should also see TCP ACK segments being returned from agila.upm.edu.ph to your computer.

Answer the following questions and if applicable, printout of the packet used to answer the question should be included in the answer with the fields used as reference to your answers highlighted.

2. What is the IP address and TCP port number used by the client computer (source) that is downloading the file from agila.upm.edu.ph?
3. On what port number is agila.upm.edu.ph sending and receiving TCP segments for this connection?

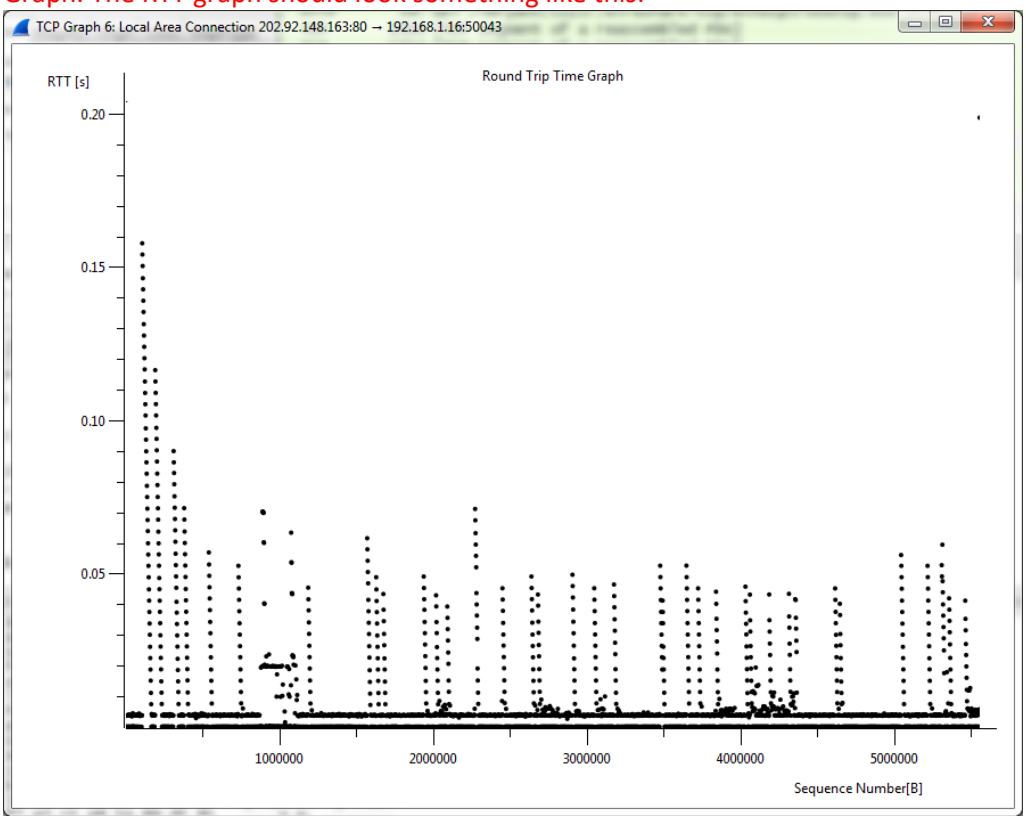
TCP Basics

With the HTTP response message to your GET request to download the installer, 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 agila.upm.edu.ph that is used to download the installer? What is it in the segment that identifies the segment as a SYN segment?
5. With respect to your answer in no. 4, how were you able to determine that this is the correct SYN segment for the TCP connection used in downloading the installer? Note that browsers usually make multiple TCP connections to a server.
6. Using the same TCP connection mentioned in no. 4, what is the sequence number of the SYN ACK segment sent by agila.upm.edu.ph to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYN ACK segment? How did agila.upm.edu.ph determine that value? What is it in the segment that identifies the segment as a SYN ACK segment?
7. What is the sequence number of the TCP segment containing the HTTP status code (200 OK)? Note that in order to find that segment, you'll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with a 200 OK status code within its DATA field.
8. Consider the TCP segment containing the 200 OK status code 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 200 OK status code)? 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 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 in our lecture for all subsequent segments.

Note: Wireshark has a 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 agila.upm.edu.ph server. Then select Statistics → TCP Stream Graph → Round Trip Time Graph. The RTT graph should look something like this:



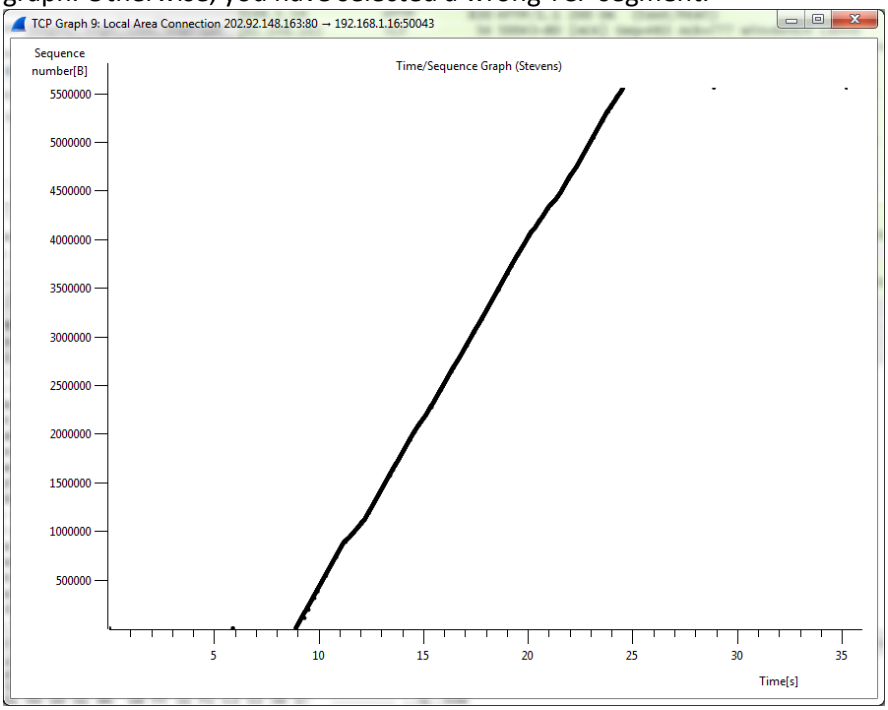
If nothing appears in your graph, that means you have selected the wrong segment. The segment that you should select should be part of the segments that will be reassembled.

9. What is the length of each of the first six TCP segments?
10. What is the minimum amount of available buffer space advertised at the received window for the entire trace? Does the lack of receiver buffer space ever throttle the sender?
11. Are there any retransmitted segments? What did you check for in order to answer this question?
12. How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment?
13. What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.

TCP congestion control in action

Let's 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 our data.

- Select a TCP segment (one that is part of TCP segments that will be reassembled) in Wireshark's listing of captured packets window. Then select from the menu: Statistics → TCP Stream Graph → Time-Sequence Graph (Stevens). You should see a graph that looks similar to the following graph. Otherwise, you have selected a wrong TCP segment.



14. 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 agila.upm.edu.ph 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 lecture.