School of Computer Science and Cybersecurity

CUC

Lab Report #

|  |  |
| --- | --- |
| Lab Name | Wireshark Lab5:TCP |
| Course Name | Computer Networks |

|  |  |  |  |
| --- | --- | --- | --- |
| Name | 赵婧宇 | Student ID | 201711123028 |

|  |  |
| --- | --- |
| Partners |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Date | 2019.06.18 | Lab Location | #48 |

|  |
| --- |
| Notes： |

**Section I Introduction**

Prepares the reader to understand the whole experiment.

|  |  |
| --- | --- |
| **Must Have:**  1. Clearly stated purpose of the experiment  2. Important background and/or theory | **May include:**  1. Description of specialized equipment  2. Justification of experiment's importance |

In this lab, we’ll investigate the behavior of the celebrated TCP protocol in detail. We’ll do so by analyzing a trace of the TCP segments sent and received in transferring a 150KB file (containing the text of Lewis Carrol’s Alice’s Adventures in Wonderland) from your computer to a remote server. 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 the server.

**Section II Methods & Materials**

Can be lists or even "refer to lab manual" where appropriate.

**Section III Procedure & Results**

Describes ACTUAL process, especially changes from planned method.

|  |
| --- |
| * **number** and **title** tables and graphs correctly and clearly * draw attention to key points in tables or graphs with a sentence * provide sample calculation only * state key result in sentence form |

**1. Capturing a bulk TCP transfer from your computer to a remote**

**server**

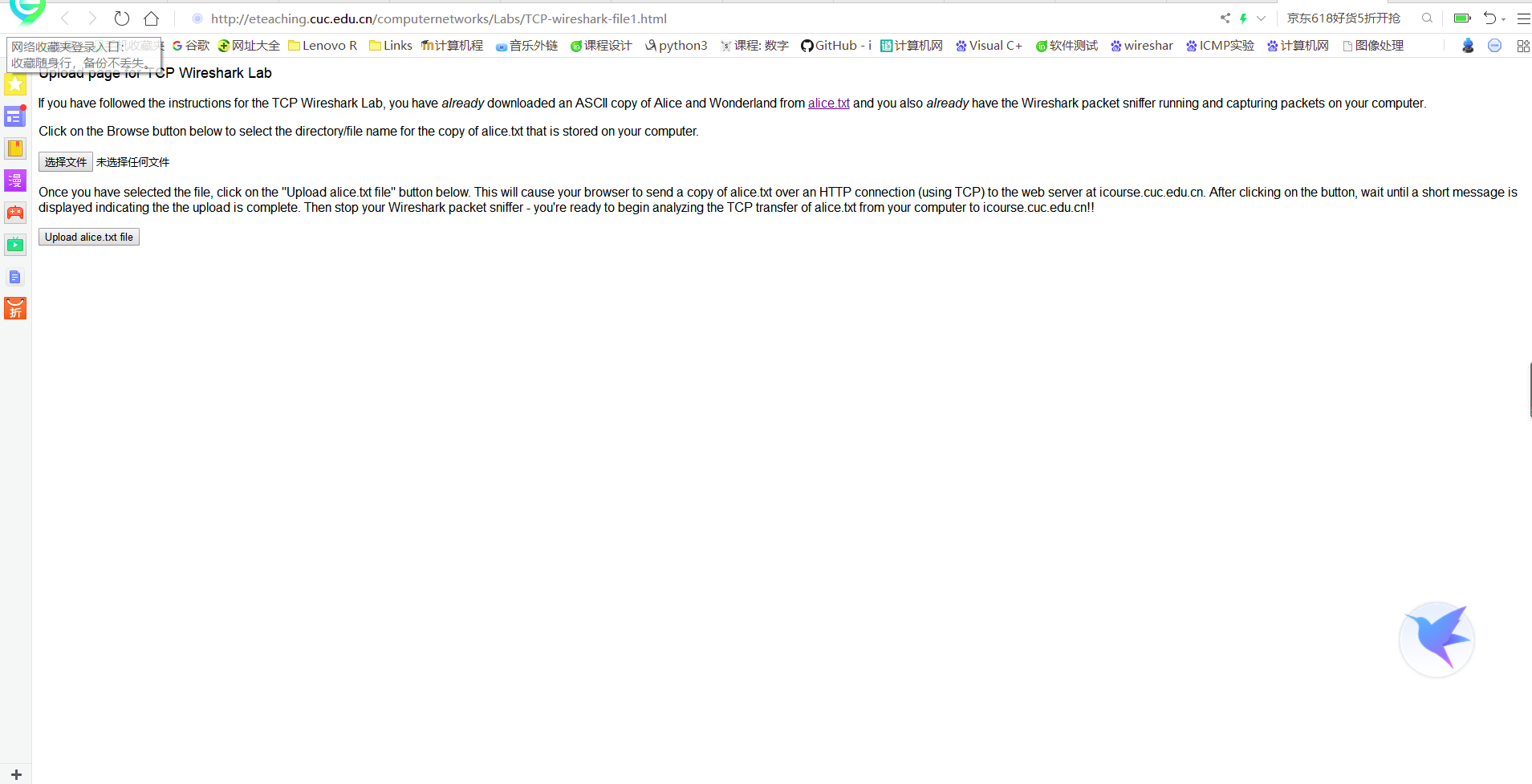
Before beginning our exploration of TCP, we’ll need to use Wireshark to obtain a packet trace of the TCP transfer of a file from your computer to a remote server. You’ll do so by accessing a Web page that will allow you to enter the name of a file stored on your computer (which contains the ASCII text 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.

Do the following:

• Start up your web browser. Go the <http://eteaching.cuc.edu.cn/computernetworks/Labs/alice.txt> and retrieve an ASCII copy of Alice in Wonderland. Store this file somewhere on your computer.

• Next go to <http://eteaching.cuc.edu.cn/computernetworks/Labs/TCP-wireshark-file1.html>.

• You should see a screen that looks like:

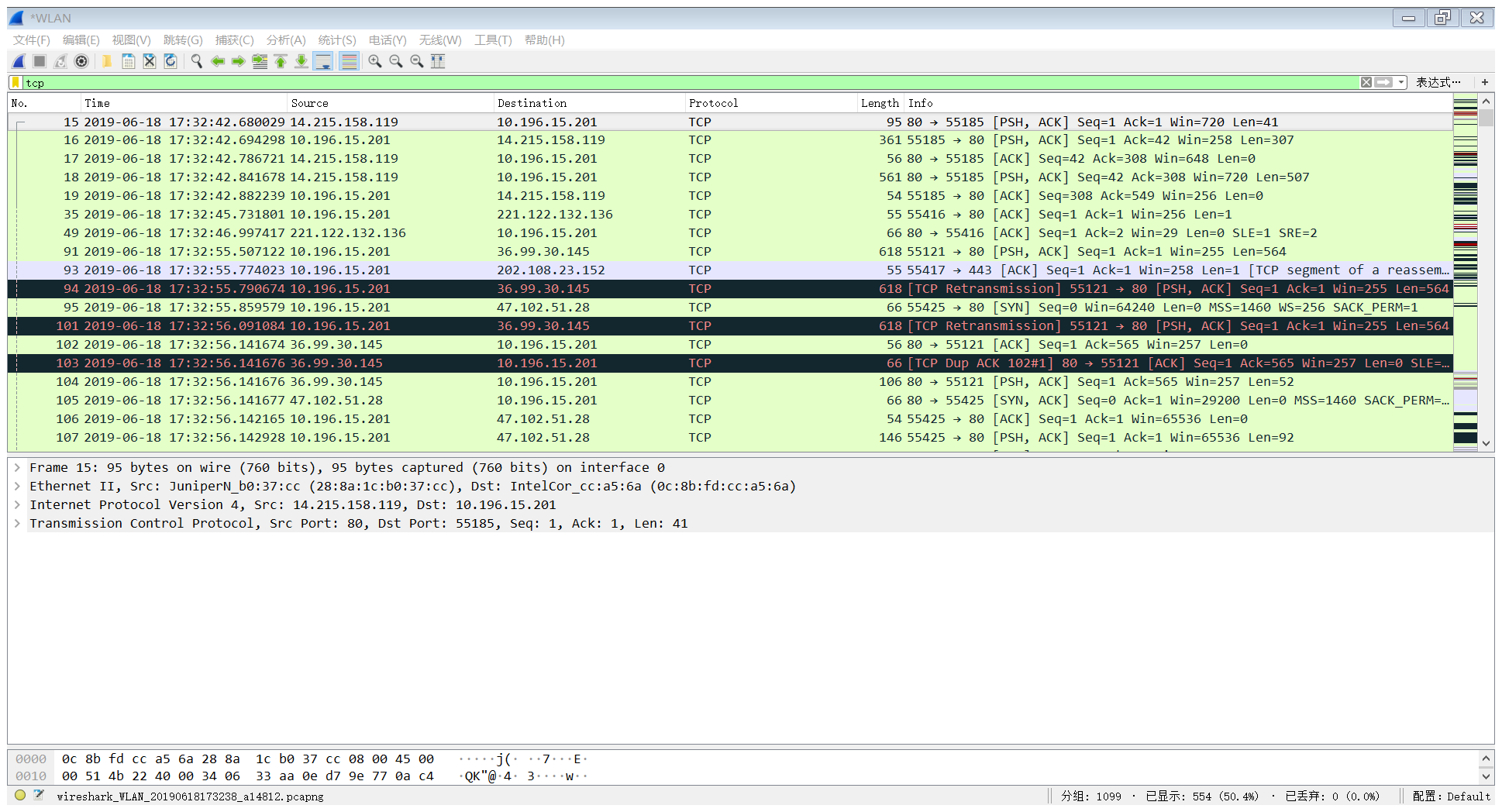


Use the Browse button in this form to enter the name of the file (full path name) on your computer containing Alice in Wonderland (or do so manually). Don’t yet press the “Upload alice.txt file” button.

• Now start up Wireshark and begin packet capture (Capture->Start) and then press OK on the Wireshark Packet Capture Options screen (we’ll not need to select any options here).

• 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. Your Wireshark window should look similar to the window shown below.



**2. A first look at the captured trace**

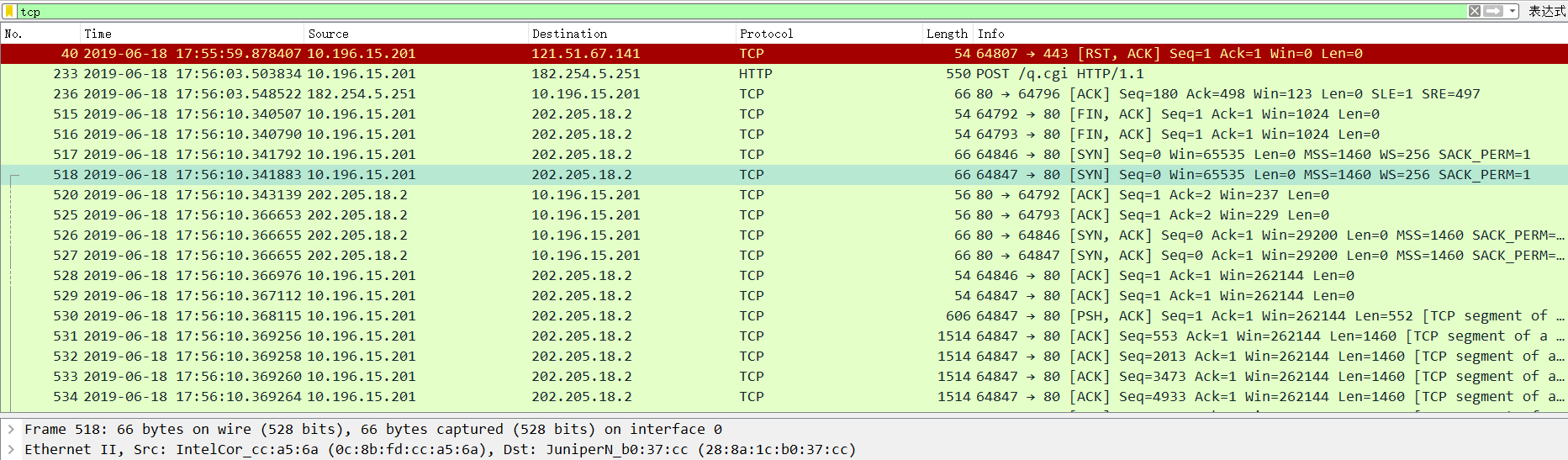
Before analyzing the behavior of the TCP connection in detail, let’s take a high level view of the trace.

SYN SYN,ACK ACK项

• 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 eteaching.cuc.edu.cn. 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’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 the web server to your computer.

Since this lab is about TCP rather than HTTP, let’s 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:



**3. TCP Basics**

Answer questions.

**4. 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 out data.

Select a TCP segment in the Wireshark’s “listing of captured-packets” window. Then select the menu : Statistics->TCP Stream Graph-> Time-Sequence-Graph(Stevens). You should see a plot that looks similar to the following plot.



**Section IV Discussion**

Answer the questions in the section [what to hand in] of the lab guide, includes two aspects:

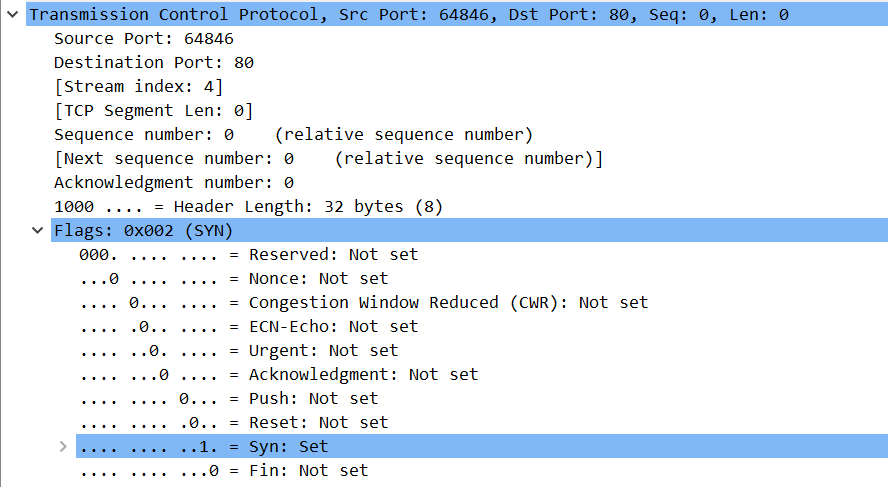
1. What is the IP address and TCP port number used by the client computer (source) that is transferring the file to eteaching.cuc.edu.cn? To answer this question, it’s probably easiest to select an HTTP message and explore the details of the TCP packet used to carry this HTTP message, using the “details 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.



* 1. What is the IP address of eteaching.cuc.edu.cn? On what port number is it sending and receiving TCP segments for this connection?



* 1. 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?

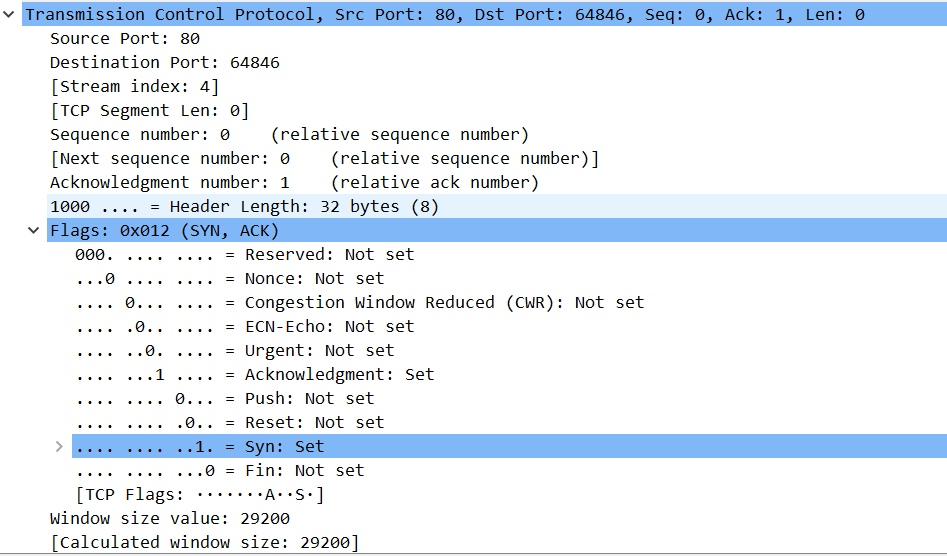


(1) SYN sequence number =0

(2) .... ...0 ... = Acknowledgment: Not Set

.... ... ..1. = Syn: Set

* 1. What is the sequence number of the SYN ACK segment sent by eteaching.cuc.edu.cn to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYN ACK segment? How did gaia.cs.umass.edu determine that value? What is it in the segment that identifies the segment as a SYN ACK segment?



（1）Sequence number: 0 (relative sequence number)

Acknowledgment number: 1 (relative ack number)

（2）Acknowledgement value= initiate sequence number of the TCP SYN segment+1

（3）.... ...1 .... = Acknowledgment: Set

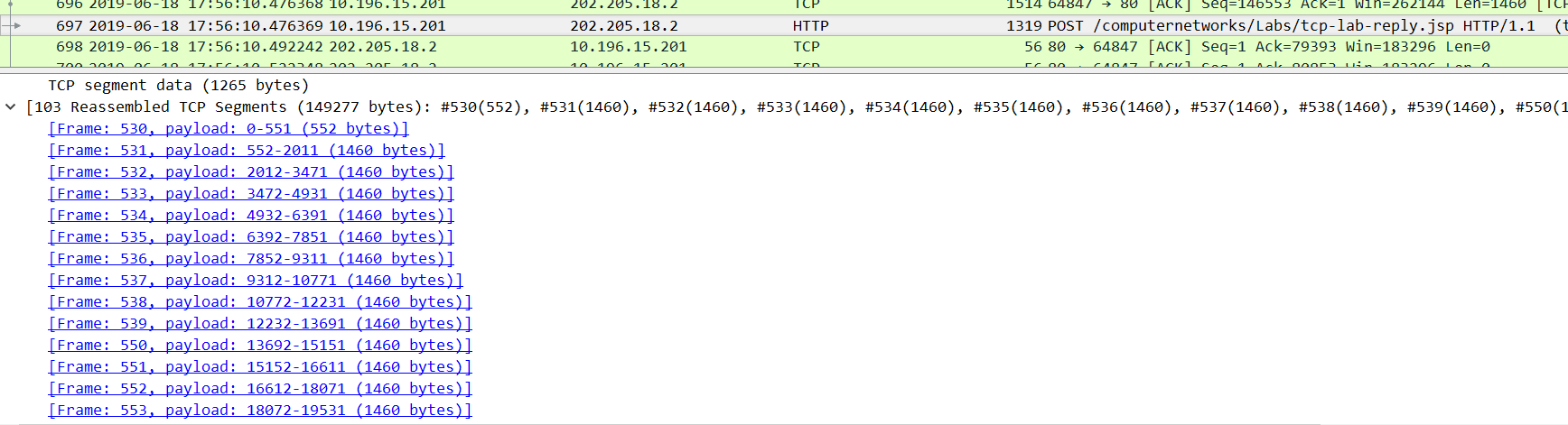
.... .... ..1. = Syn: Set

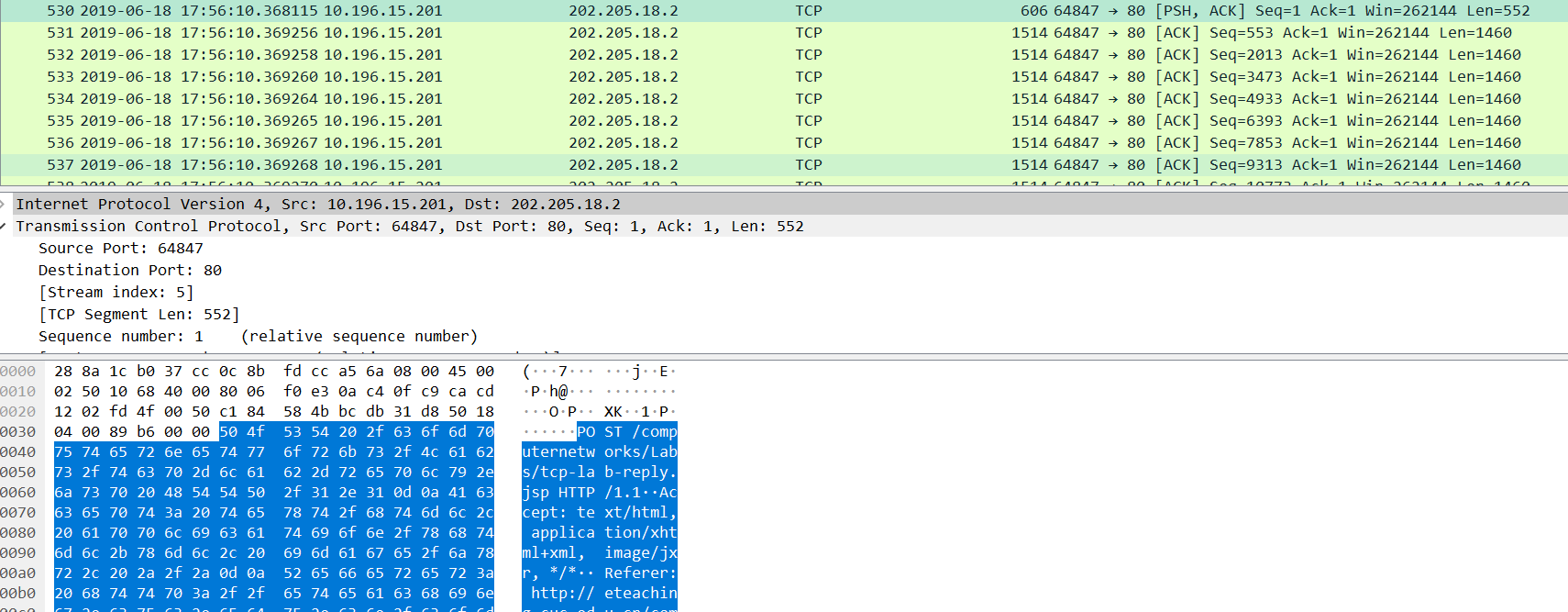
* 1. What is the sequence number of the TCP segment containing the HTTP POST command? Note that in order to find the POST command, you’ll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with a “POST” within its DATA field.

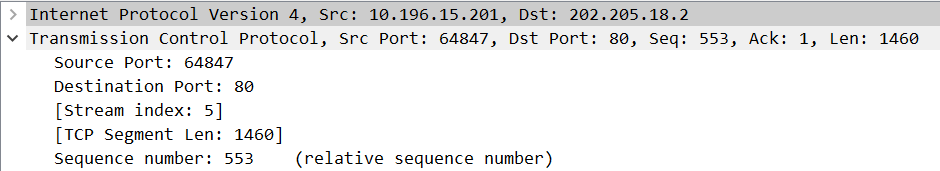


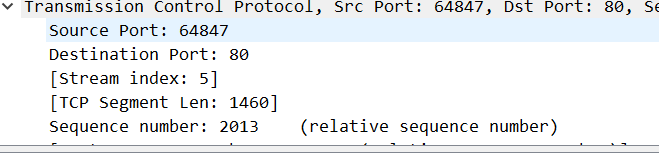
* 1. Consider the TCP segment containing the HTTP POST as the first segment in the TCP connection. What are the sequence numbers of the first six segments in the TCP connection (including the segment containing the HTTP POST)? At what time was each segment sent? When was the ACK for each segment received? 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 the text for all subsequent segments.

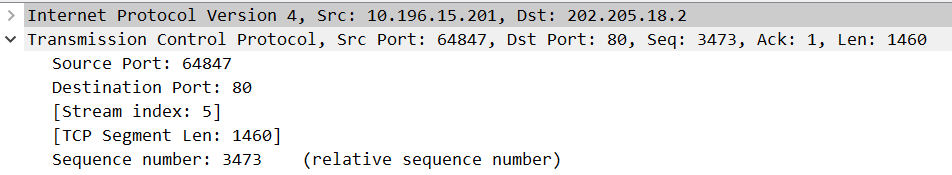
Note: Wireshark has a nice feature that allows you to plot the RTT for each of the TCP segments sent. Select a TCP segment in the “listing of captured packets” window that is being sent from the client to the eteaching.cuc.edu.cn server. Then select: Statistics->TCP Stream Graph>Round Trip Time Graph.

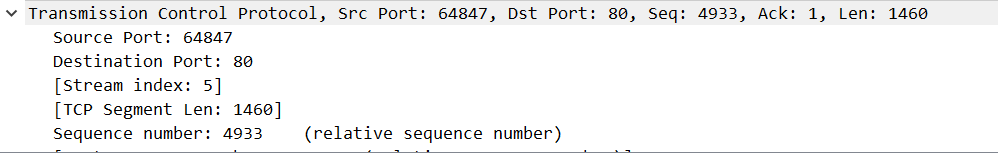
(1) 

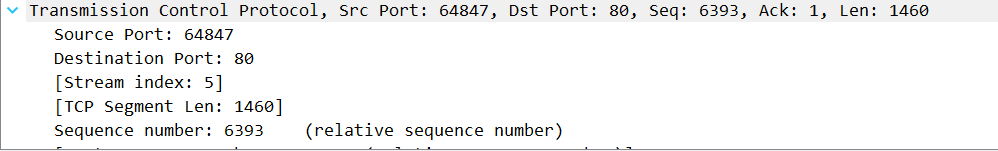




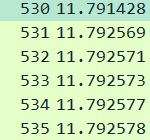




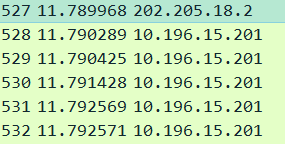




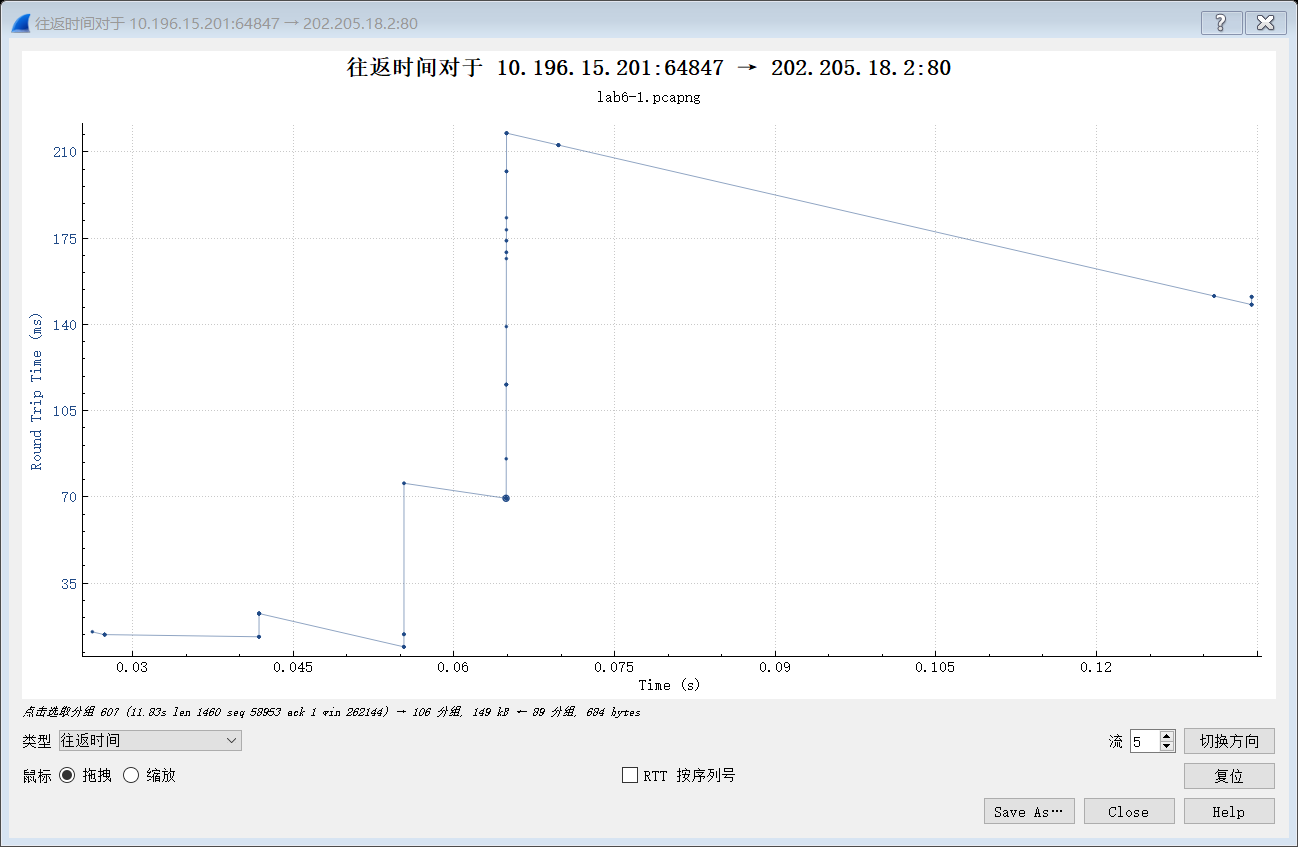
(2)



(3)



(4)



* 1. What is the length of each of the first six TCP segments?













* 1. What is the minimum amount of available buffer space advertised at the received for the entire trace? Does the lack of receiver buffer space ever throttle the sender?

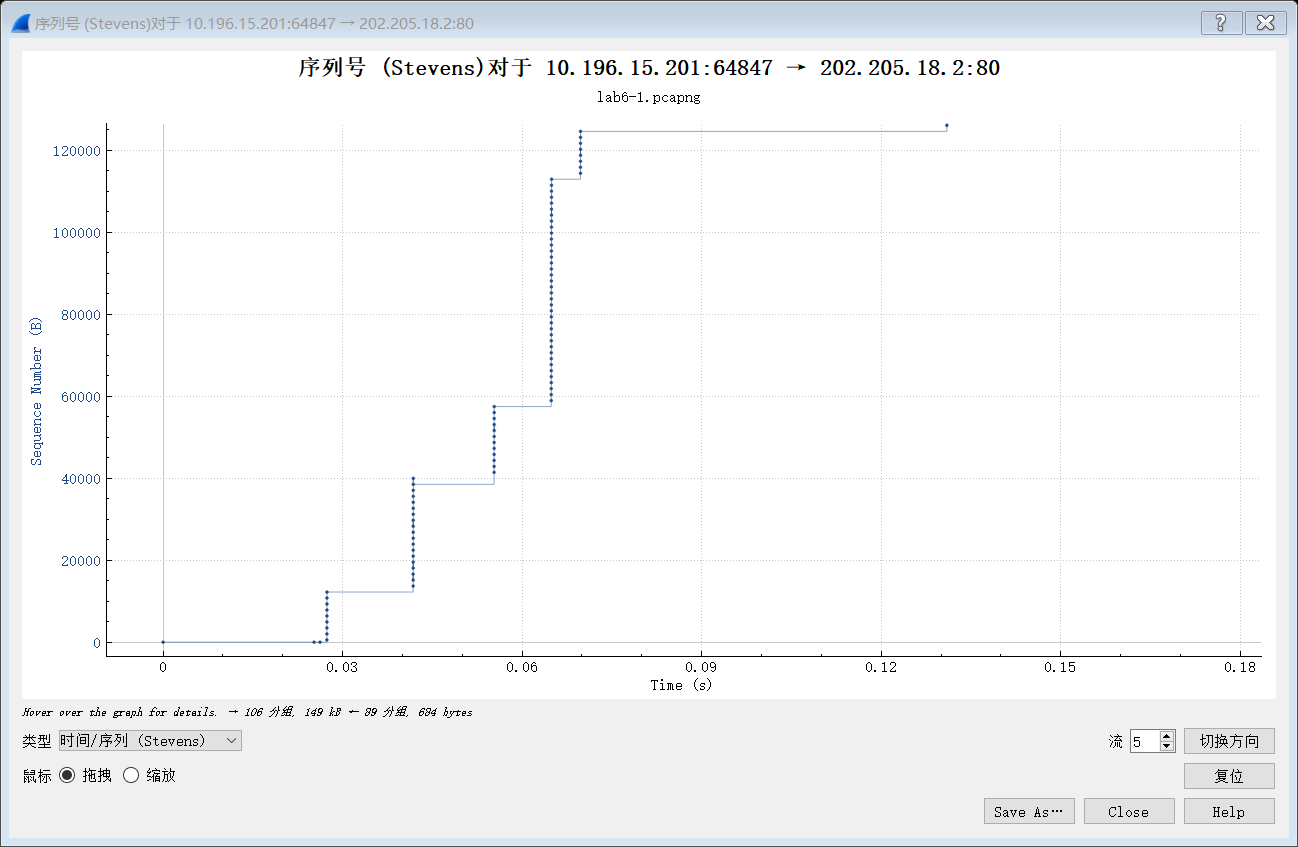
Answer: 29200; 262144; No





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

Answer: No. Because the sequence number is increasing



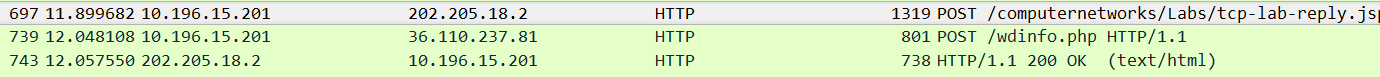
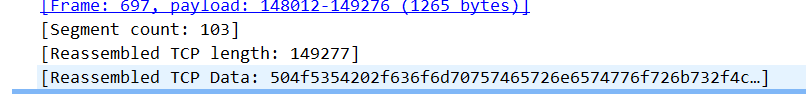
* 1. How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment.





2013-553=1460

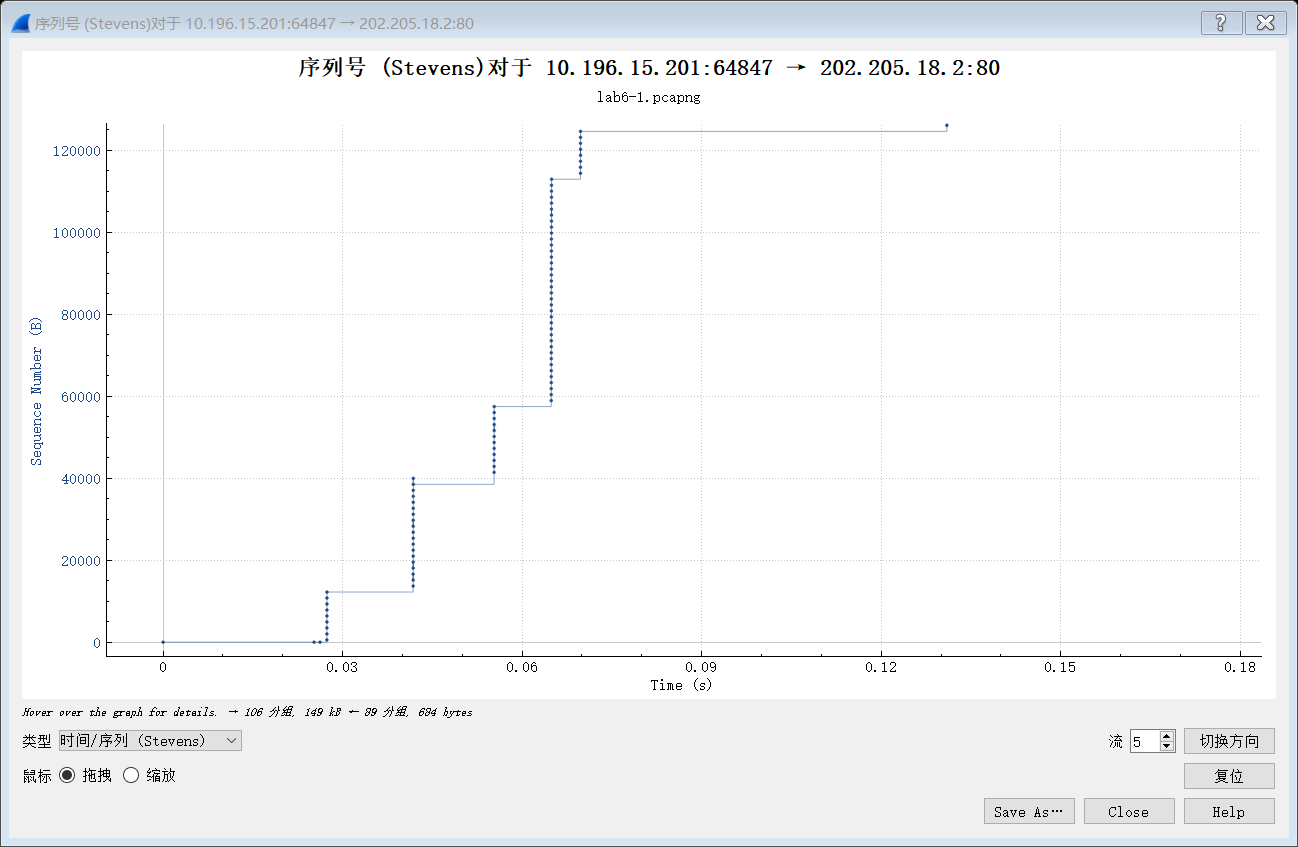
* 1. What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.



Answer: 149277/(12.057550-12.048108)=149277/0.009442=15,809,891.972039 bytes/sec

* 1. 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 eteaching.cuc.edu.cn 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 text.

Answer: We can see from the figure above that TCP Slow Start begins at the start of the connection. The identification of the TCP slow start phase and congestion avoidance phase depends on the value of the congestion window size of this TCP sender, we can tell easily where TCP’s slow ends and where congestion avoidance takes over.



**Section V Conclusion**

States what is known as a result of the experiment.

|  |  |
| --- | --- |
| **Must do:**  1. State what's known  2. Justify that statement | **May do:**  1. State significance of findings  2. Suggest further research |

Understanding the application layer TCP protocol through wireshark packet capture.