**电子科技大学**

**实**

**验**

**报**

**告**

学生姓名：李昊霖

学号：2019270102007

课程名称：计算机网络基础

指导教师：杨宁

日期： 2021 年 4 月 1 日

实验项目名称： 3-1 wireshark-TCP实验

报告评分： 教师签字：

**一、实验概要**

**实验目的:**

通过进行Wireshark抓包实验，学习TCP 的 SEQ 和 ACK 序列号和确认号在 TCP 协议中的作用；学习TCP 建立连接三次握手的过程； TCP 的拥塞控制算法；进行 TCP 连接性能的计算；加强对 TCP 报文段结构的了解。

**实验内容**：

通过电脑向远程服务器传输一份 150KB 的文件（一份 Lewis Carrol 的“爱丽丝梦游仙境”文本），并分析 TCP 传输内容的发送和接收过程是如何实现的。 我们将研究 TCP 对序列和确认号的使用，以提供可靠的数据传输；以及 TCP 的拥塞控制算法——慢启动和拥塞避免，在此过程中，我们会看到 TCP 的接收器发送流量控制的机制。 我们还将简要地观察 TCP 连线的设置，并研究计算机和服务器之间 TCP 连线的性能（吞吐量和往返时间）

**二、实验步骤、数据及分析结果**

**实验步骤:**

（一）Capturing a bulk TCP transfer from your computer to a remote

server

1.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.

2.Next go to http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html.

3.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.

4.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).

5.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.

6.Stop Wireshark packet capture. Your Wireshark window should look similar to

（二）A first look at the captured 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.

（三）TCP congestion control in action

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, which was created from the captured packets in the packet trace tcp-ethereal trace-1 in http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip (see earlier footnote ):

**实验数据及分析:**

**回答指导书问题：**

问题1-3：

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

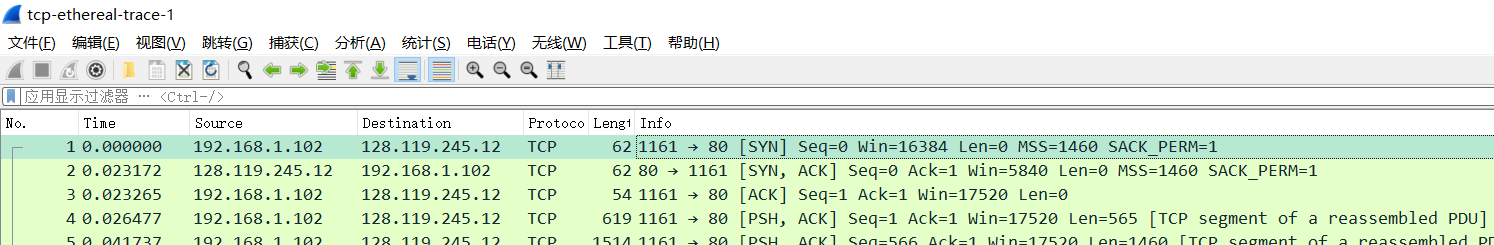
probably easiest to select an HTTP message and explore the details of the TCP

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.

答：IP地址为192.168.1.102，源端口号为1161，如下图。



2. What is the IP address of gaia.cs.umass.edu? On what port number is it sending

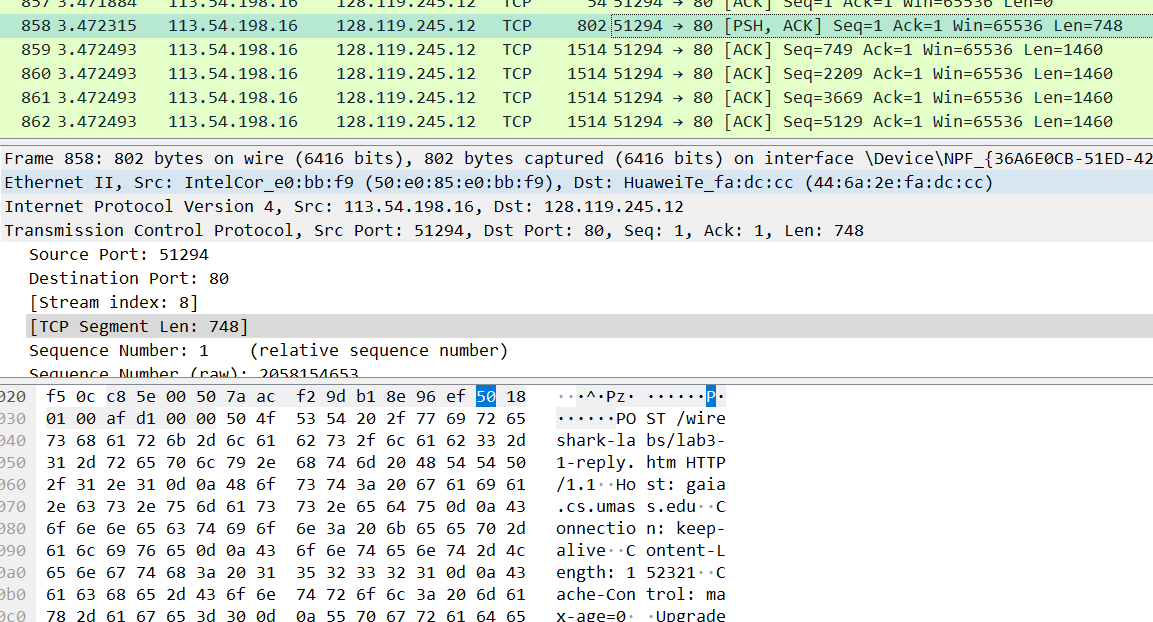
and receiving TCP segments for this connection?

答：IP地址为128.119.245.12，目的端口号为80

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?

答：IP地址为113.54.198.16，源端口号为51294，如下图。



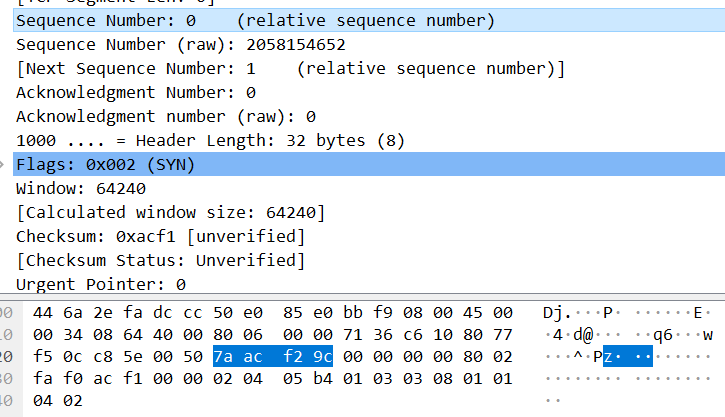
问题4-12：（以下问题均分析自己的抓包文件）

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?

答：相对序列号0，绝对序列号7aacf29c(十六进制)；Flags标识其为SYN类型。



5. What is the sequence number of the SYNACK segment sent by gaia.cs.umass.edu

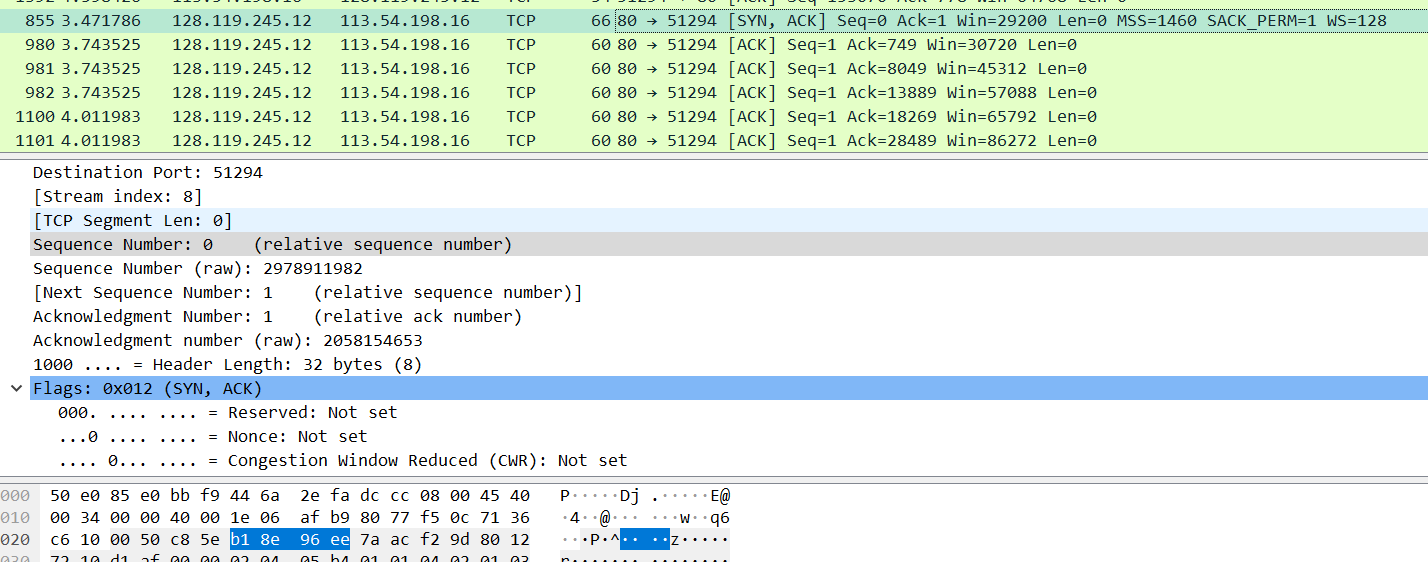
to the client computer in reply to the SYN? What is the value of the

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?

答：相对序列号0，绝对序列号b18e96ee (十六进制)；Acknowledgment栏位值为1；Flags标识其为SYN,ACK类型。



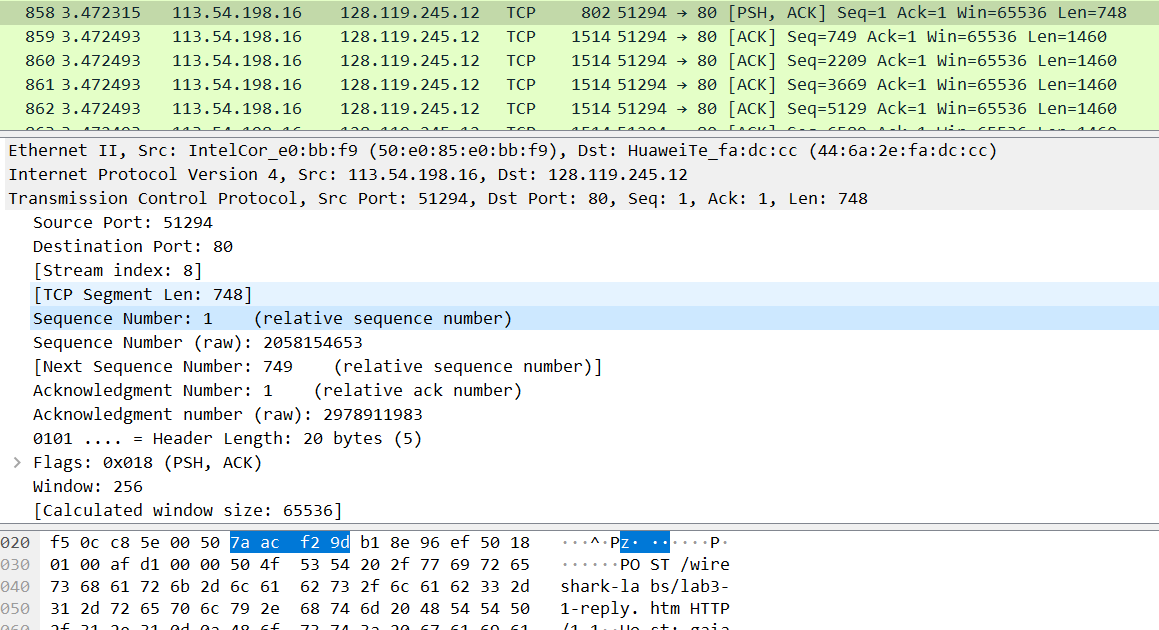
6. What is the sequence number of the TCP segment containing the HTTP POST

command? Note that in order to find the POST command, you’ll need to dig into

the packet content field at the bottom of the Wireshark window, looking for a

segment with a “POST” within its DATA field.

答：相对序列号1，绝对序列号7aacf29d(十六进制)；如图标蓝序列号正下方为POST字段。



7. 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 (see Section 3.5.3, page 242 in

text) 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 on page 242 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

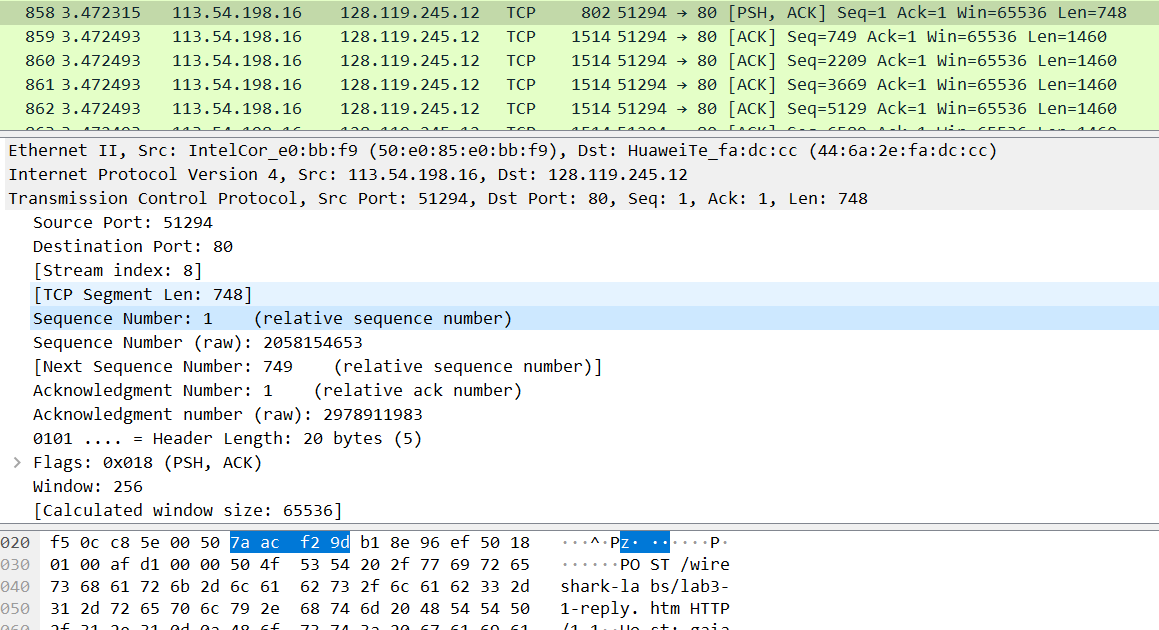
captured packets” window that is being sent from the client to the

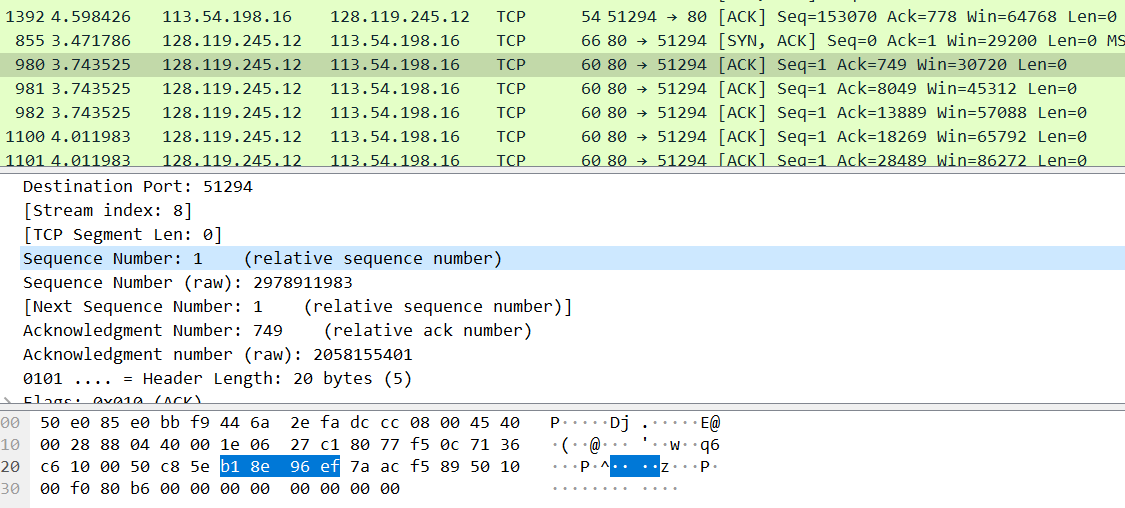
gaia.cs.umass.edu server. Then select: *Statistics->TCP Stream Graph-*

*>Round Trip Time Graph.*

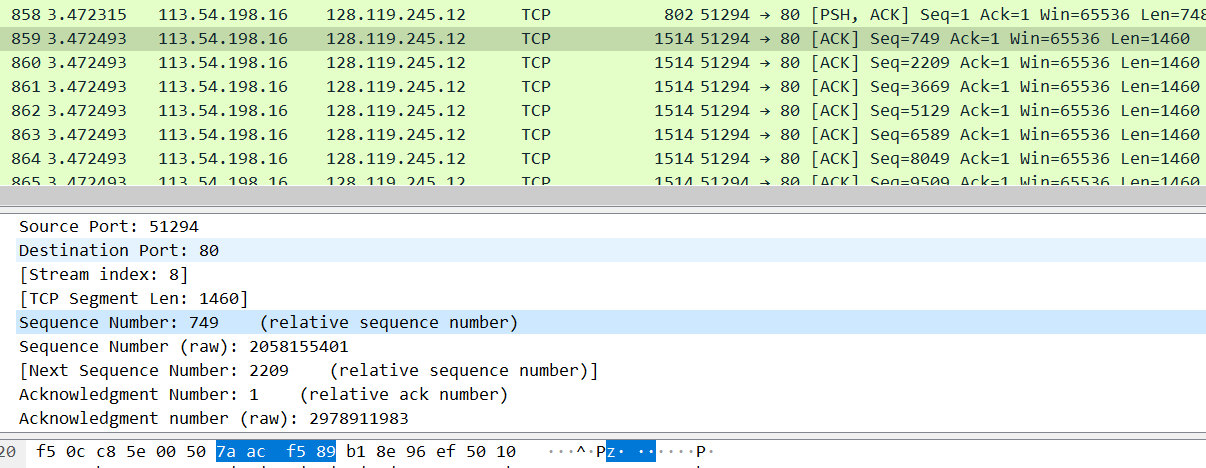
答：

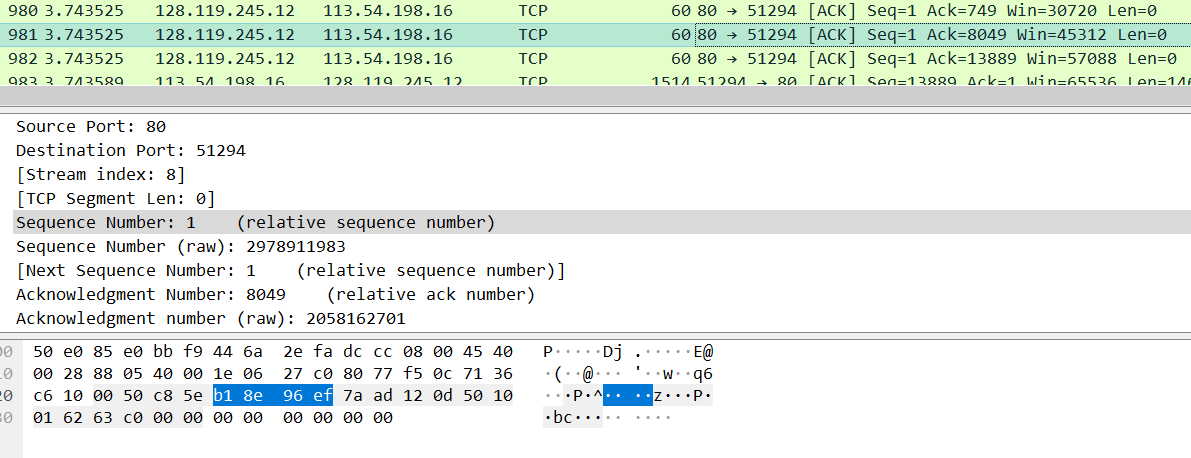
1. 相对序列号1，绝对序列号7aacf29d(十六进制)；发送时间3.472315s，收到时间3.743525s，RTT= 0.271210000s，两份报文截图如下。



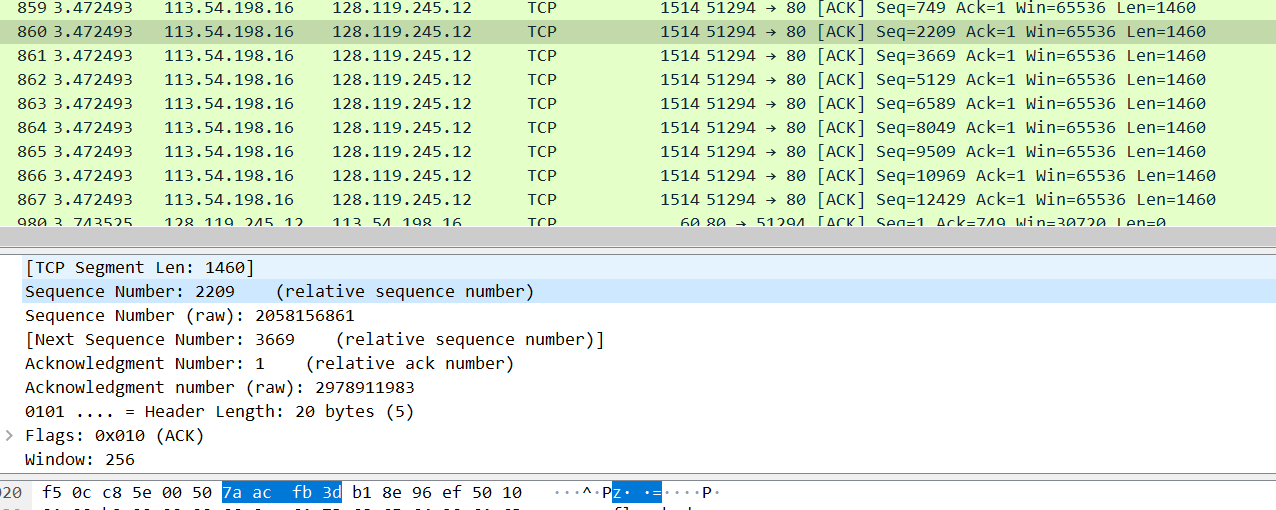


1. 相对序列号749，绝对序列号7aacf589(十六进制)；发送时间3.472493s，收到时间3.743525s，RTT= 0.271032000s，两份报文截图如下（确认报文直接确认了后五份报文）。

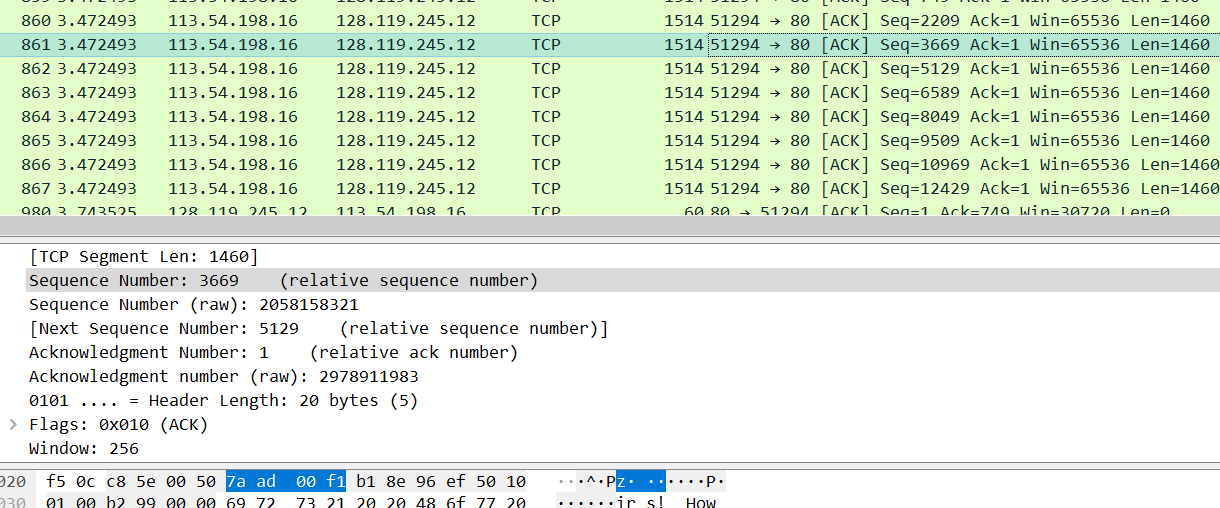




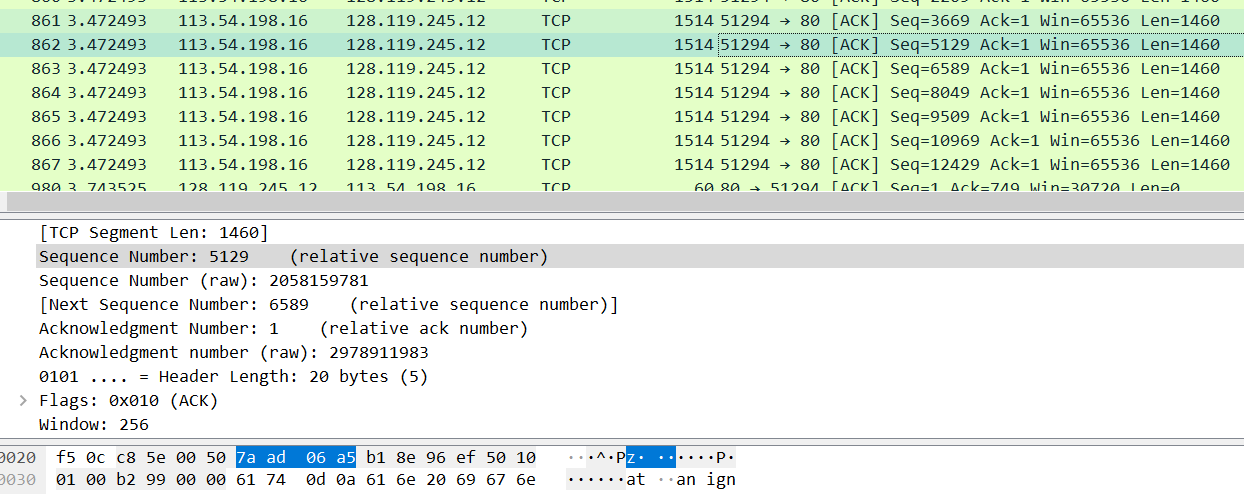
1. 相对序列号2209，绝对序列号7aacfb3d(十六进制)；发送时间3.472493s，收到时间3.743525s，RTT= 0.271032000s，确认报文同2（除序列号外其它同2）。



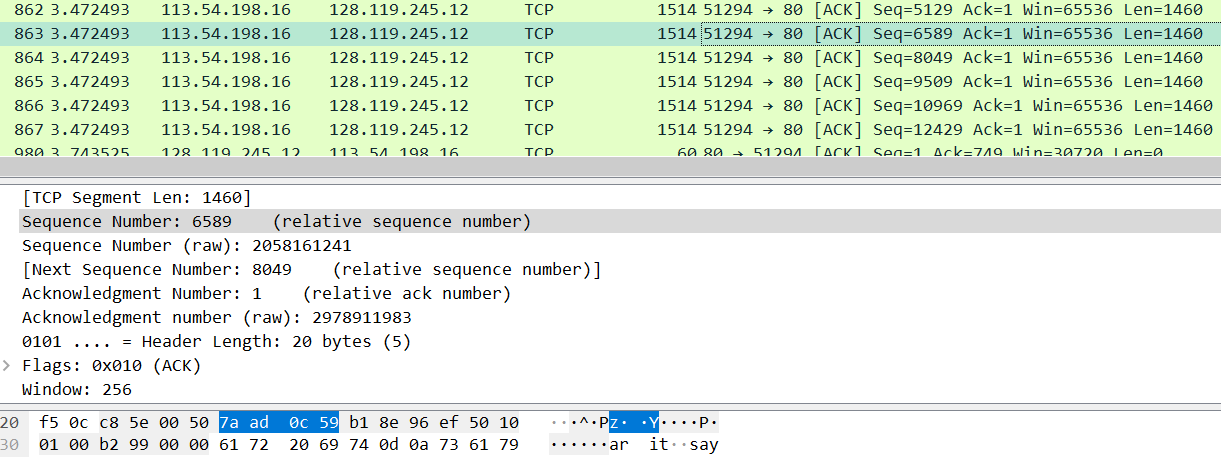
1. 相对序列号3669，绝对序列号7aad00f1(十六进制)，其它同2。



1. 相对序列号5129，绝对序列号7aad06a5(十六进制) ，其它同2。

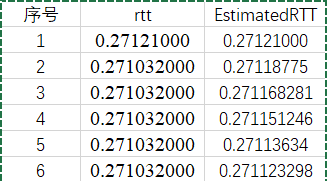


1. 相对序列号6589，绝对序列号7aad0c59(十六进制) ，其它同2。



计算EstimatedRTT：第一个 EstimatedRTT的值等于第一个区段的测量 RTT

公式EstimatedRTT=EstimatedRTT\*0.875+0.125\*sampleRTT，结果如下



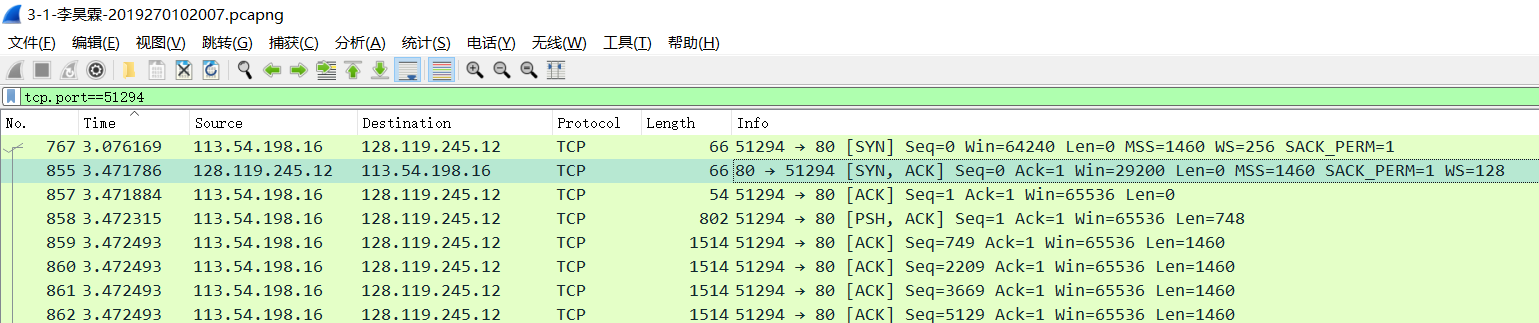
8. What is the length of each of the first six TCP segments?

答：第一个为748bytes，其余均为1460bytes。

9. 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?

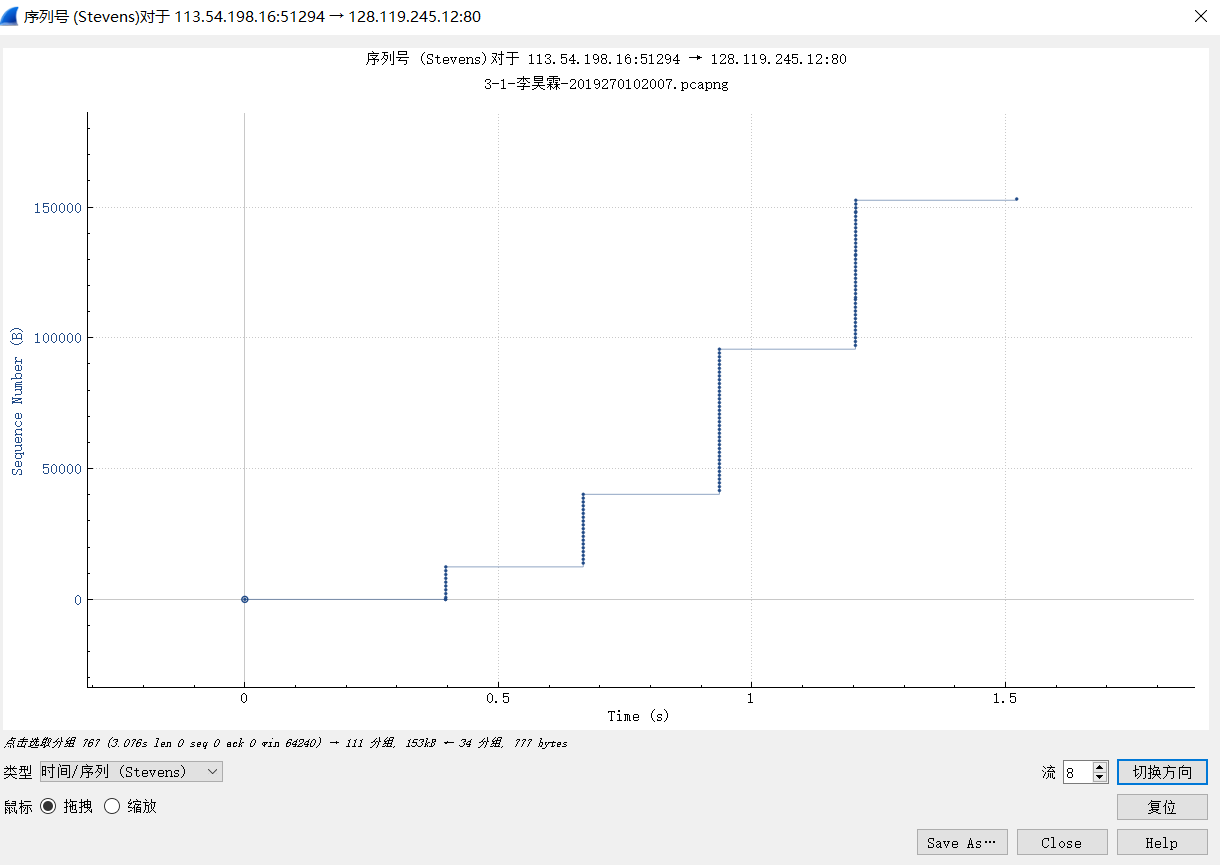
答：最小可用缓冲空间为窗口大小29200，是建立连接时的第二次握手；缺少接收器缓冲区空间会限制发送方传送 TCP 区段。



10. Are there any retransmitted segments in the trace file? What did you check for (in

the trace) in order to answer this question?

答：无重传区段，如下图，序列号不断增大。

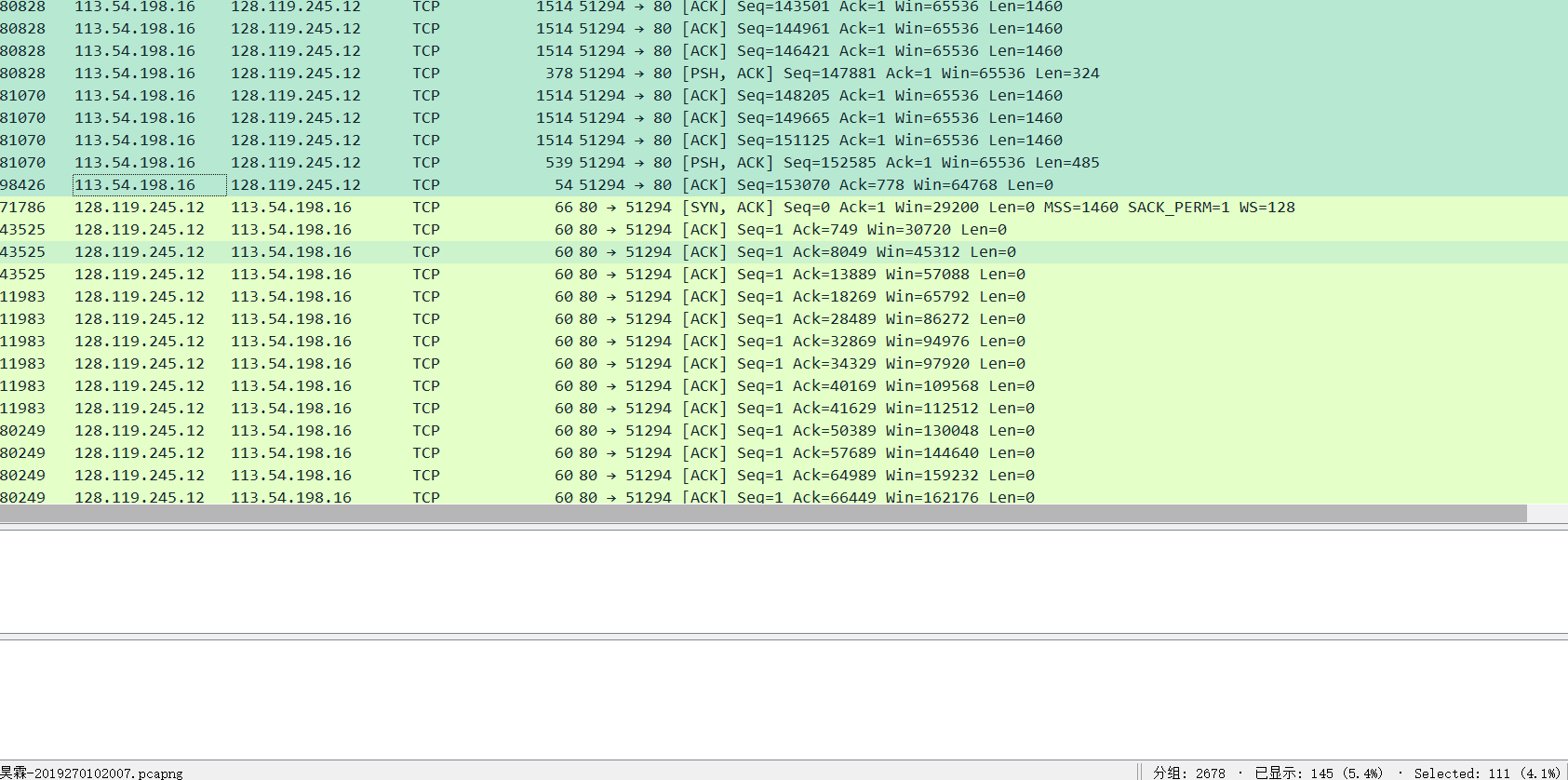


11. How much data does the receiver typically acknowledge in an ACK? Can you

identify cases where the receiver is ACKing every other received segment (see

Table 3.2 on page 250 in the text).

答：如下图，共111份源自客户端的TCP区段，而确认ACK为34份，平均一次确认3.26个TCP区段；有接收方一次确认多个区段的情况，如第七题中后五个区段一次确认。



12. What is the throughput (bytes transferred per unit time) for the TCP connection?

Explain how you calculated this value.

答：Time since first frame in this TCP stream: 1.204901000 seconds，Reassembled TCP length: 153069bytes，吞吐量=153069/1.204901=127038.6b/s=127Kb/s

问题13-14：

13. Use the *Time-Sequence-Graph(Stevens*) plotting tool to view the sequence

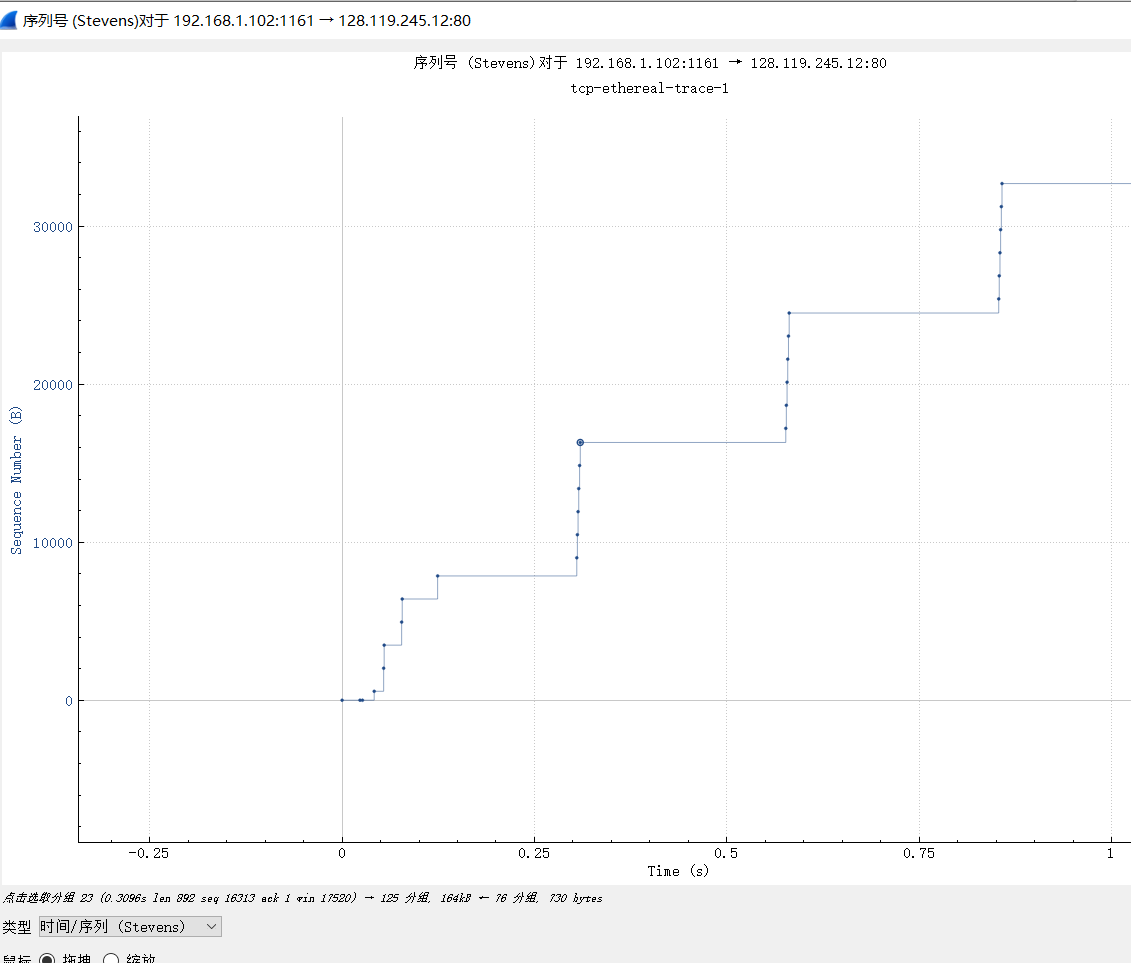
number versus time plot of segments being sent from the client to the gaia.cs.umass.edu server. Can you identify where TCP’s slowstart phase begins

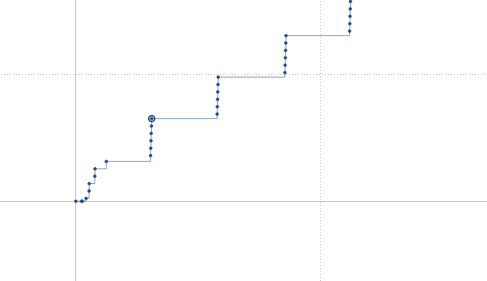
and ends, and where congestion avoidance takes over? Comment on ways in

which the measured data differs from the idealized behavior of TCP that we’ve

studied in the text.

答：时序图如下， 分析下图可看出，在t=0.3s（图中间的上升段）以后发送的报文段数量保持稳定，而之前发送报文段的数量逐渐增长；据此可判断，在t<0.3s时属于慢启动阶段，而此后拥塞避免接管。

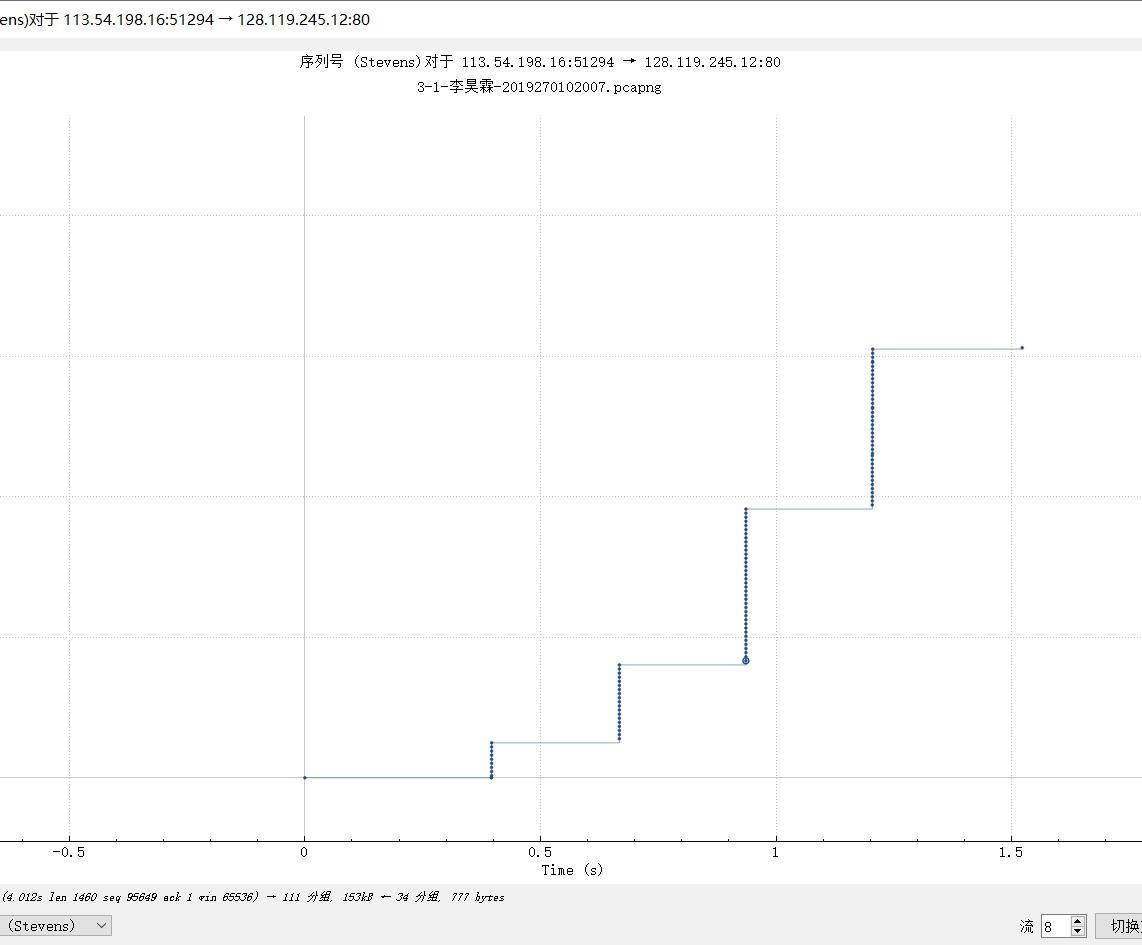




14. Answer each of two questions above for the trace that you have gathered when

you transferred a file from your computer to gaia.cs.umass.edu

答：时序图如下，由于数据量少，发送速度较快，拥塞控制，慢启动的情况不明显；但可看出后两次发送的报文段数量一致，拥塞控制应是在t=0.9s左右接管，在此之前为慢启动。



**其它情况：**

1.问题1：是否从慢启动进入到拥塞避免阶段？为什么？

答：同13题

2.问题2：是否呈现了明显的指数增长现象？为什么？

答：呈增长趋势且增速逐渐增大，但数据量较少， 指数增长现象不明显。

3.问题3：初始拥塞窗口值是否为1个MSS？为什么？

答：

根据RFC文档：

If SMSS(Sender’s MSS) > 2190 bytes:

IW = 2 \* SMSS bytes and MUST NOT be more than 2 segments

If (SMSS > 1095 bytes) and (SMSS <= 2190 bytes):

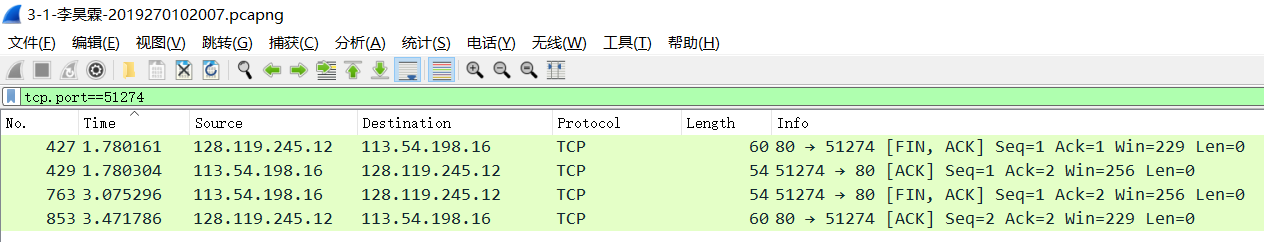
IW = 3 \* SMSS bytes and MUST NOT be more than 3 segments

if SMSS <= 1095 bytes:

IW = 4 \* SMSS bytes and MUST NOT be more than 4 segments

因为MSS=1460 bytes故初始拥塞窗口值应为3个MSS。

4．结束连接，下图显示了结束连接时四次握手的情况。



**三、总结及心得体会**

本次实验让我掌握了利用Wireshark分析TCP区段的方法，加深了我对TCP连接通过三次握手建立，以及根据序列号传送、确认以及四次握手结束连接等过程的理解 。

**四、对本实验过程及方法、手段的改进建议**

可以在问题设计上减少重复性的问题，对部分问题深入讨论，如EstimatedRTT，缓冲窗口等。