中国科学技术大学计算机学院

计算机网络实验报告

实验三

利用 Wireshark 观察 TCP 报文

学　　号：**PB15111604**

姓　　名：金泽文

专　　业：计算机科学与技术

指导老师：张信明

中国科学技术大学计算机学院

2017 年 12 月 8 日

一、 实验目的

1、 通过捕获观察并分析TCP报文，理解TCP的细节，包括：为了reliable传输的SEQ、ACK序号使用；TCP的拥塞控制算法-慢启动和拥塞避免；TCP的流控制机制；TCP连接的建立。

二、 实验原理

Wireshark 是一个 packet 分析工具，可以抓取 packet，并分析出详细信息。Wireshark 使用 wincap 作为接口，直接与网卡进行 packet 交换，监听共享网络上传送的 packet。

三、 实验条件

1、 硬件条件： 联想 Y700：

i5-6300HQ 2.30GHz

16G 内存

Intel(R) Dual Band Wireless-AC 3165

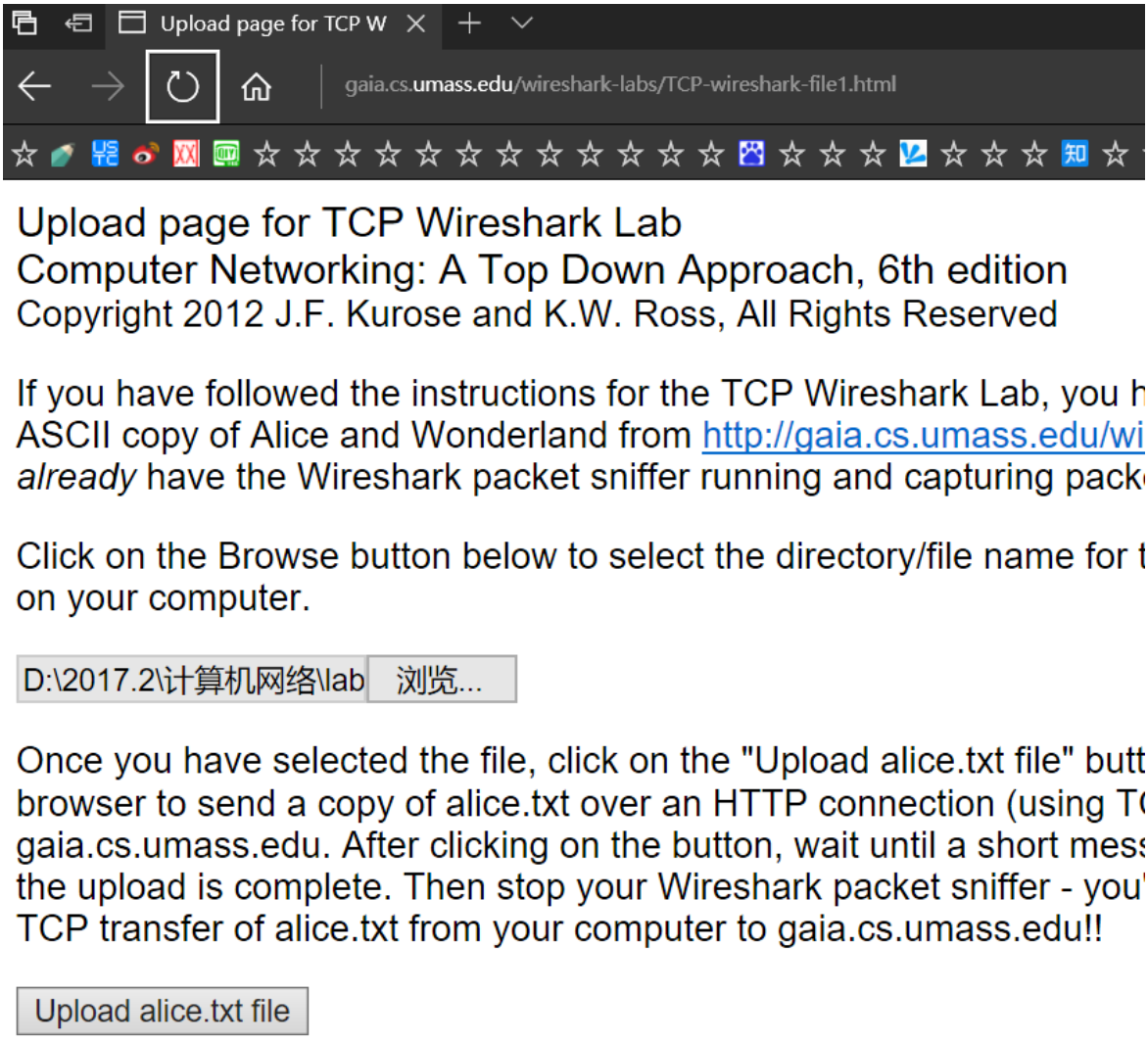
2、 软件条件： Win10 Professional 1703

Wireshark2.4.2

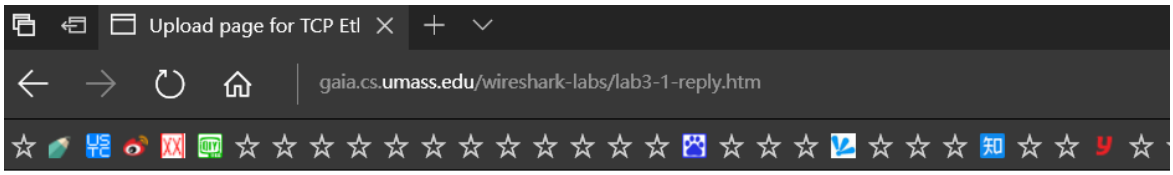
四、 实验过程

1、 向远程服务器发送一个 txt 文件，并捕获 TCP 报文。

- 首先下载 alice.txt
- 到指定页面选中要上传的文件。

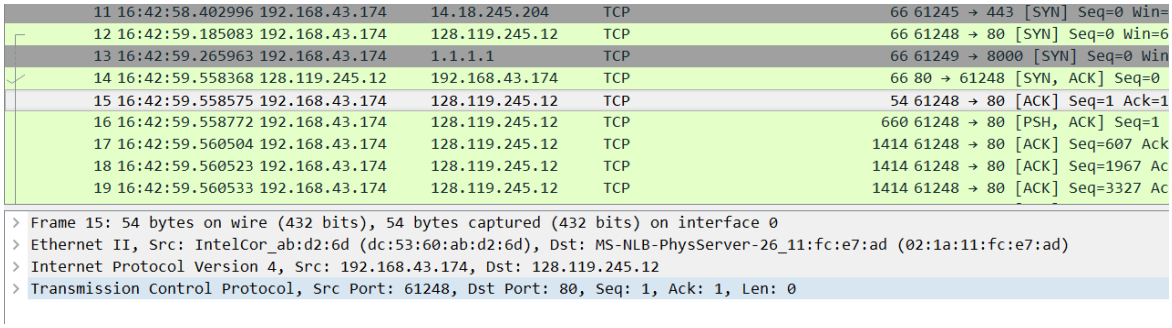


- 打开 wireshark 开始捕获。
- 开始上传文件。
- 成功上传之后，终止捕获。



Congratulations!

You've now transferred a copy of alice.txt ffrom your computer to gaia.cs.uma
stop Wireshark packet capture. It's time to start analyzing the captured Wiresl



2、 开始回答问题。

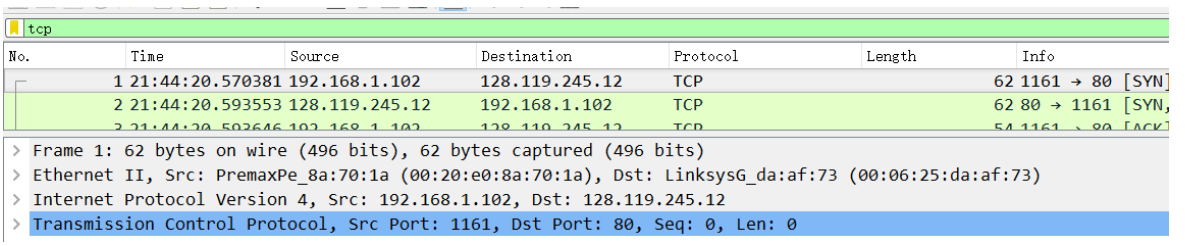
五、 结果分析

以下是 pdf 中 14 个问题对应的回答

（除 3、14 题，都是用的提供的 trace 文件。）

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：客户端电脑的 ip 地址：192.168.1.102；TCP 端口：1161。如下图：



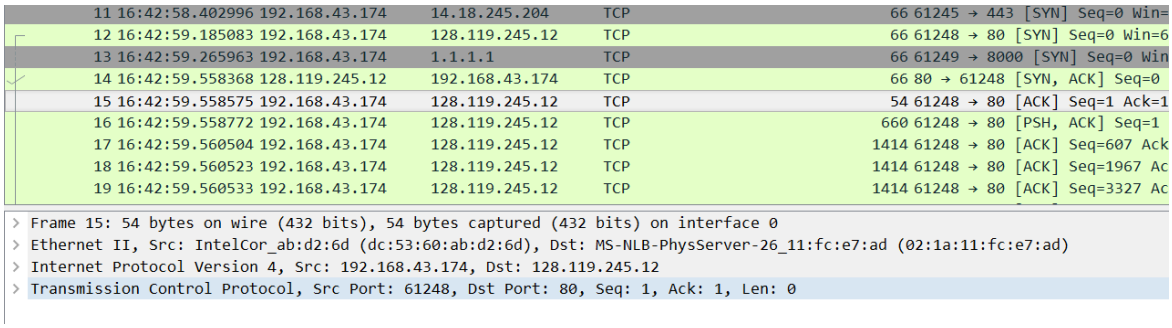
2. What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection?

答：如题1的图，IP地址为：128.119.245.12；端口号：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?

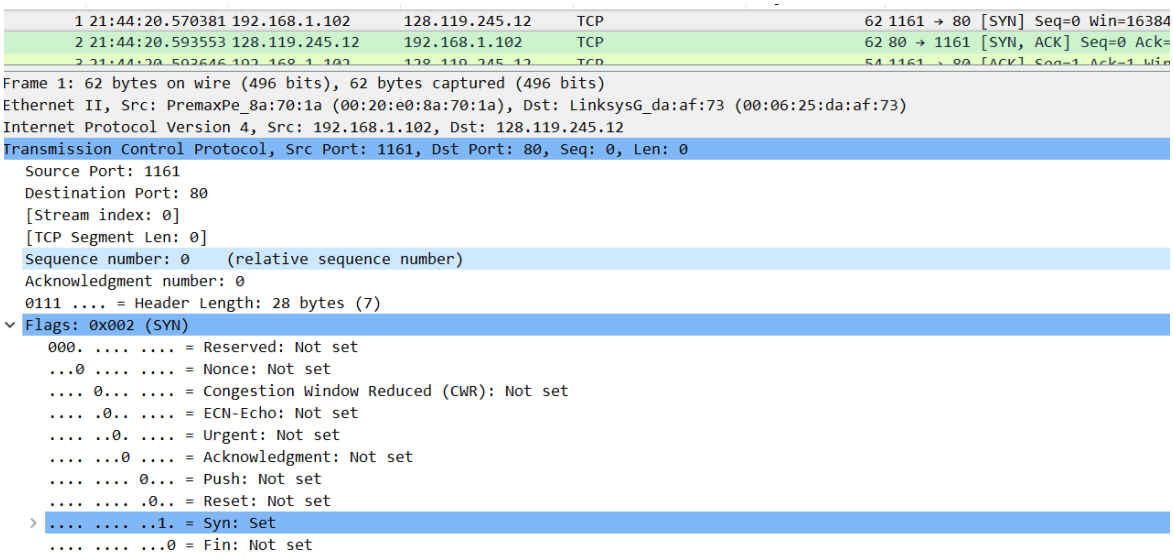
答：自己捕获一遍之后，得到下图。我自己电脑的 IP 地址为：

192.168.43.174，端口号：61248

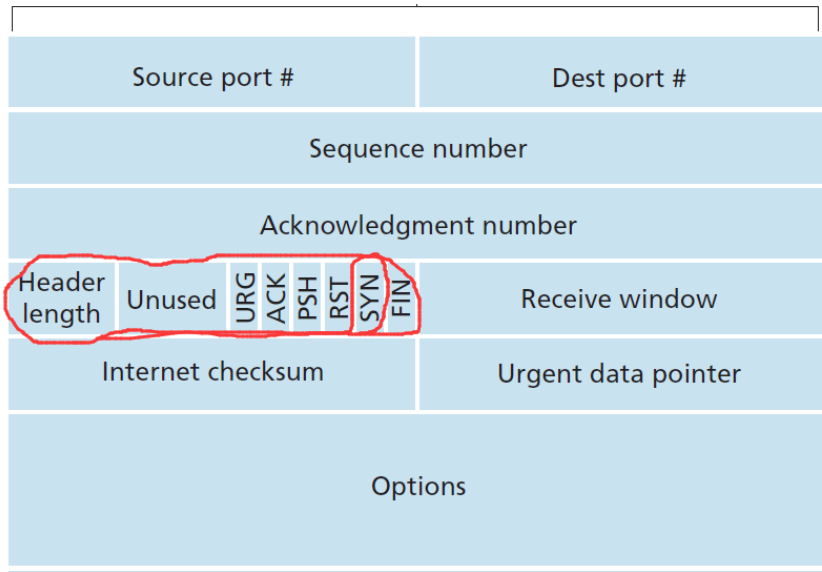


4. What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu? What is it in the segment that identifies the segment as a SYN segment?

答：如下图，SYN 的 Seq 序号为 0。



确认报文为 SYN 报文的标志是报文的 TCP HEADER 中的 flag field 中被置为 1 的 SYN 标志。



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?

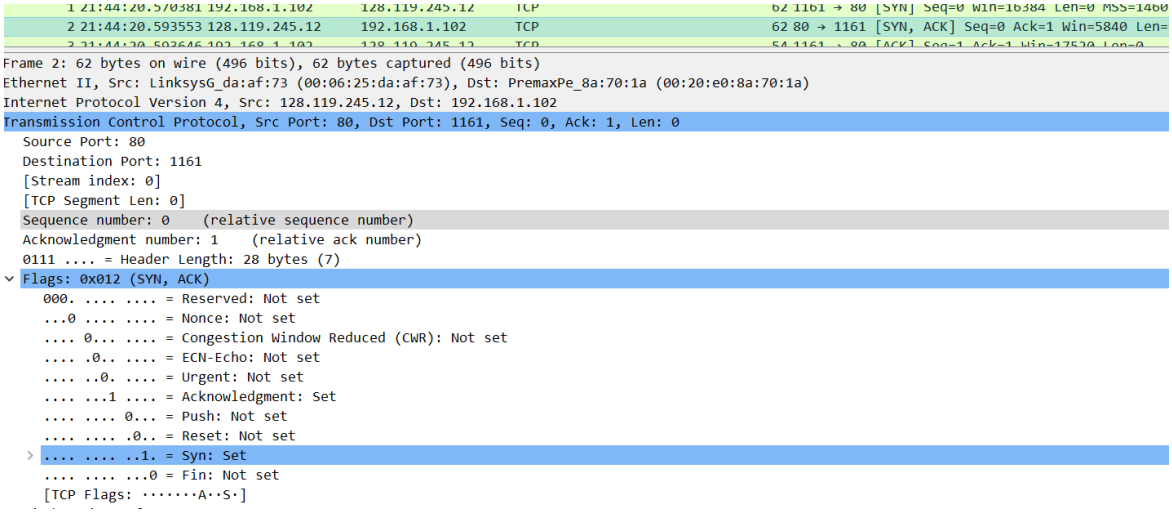
答：如下图。

序号为 0。

Acknowledgement field 为 1。

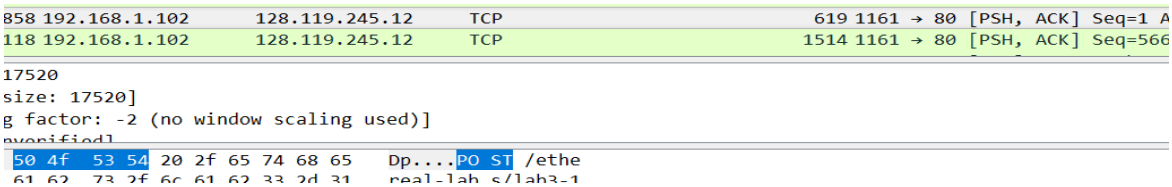
gaia.cs.umass.edu 将该值设置为所期望的下一个来自客户端的报文的 Sequence Number。

Flag field 中被置为 1 的 ACK 位和 SYN 位。



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：如下图，1。



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 page 249 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 249 for all subsequent segments.

答：根据下图，再根据等式：

$$EstimatedRTT = (1 - \alpha) \cdot EstimatedRTT + \alpha \cdot SampleRTT$$

```
21:44:20.596858 192.168.1.102   128.119.245.12   TCP    619 1161 → 80 [PSH, ACK] Seq=1 Ack=
21:44:20.612118 192.168.1.102   128.119.245.12   TCP   1514 1161 → 80 [PSH, ACK] Seq=566 Ac
21:44:20.624318 128.119.245.12  192.168.1.102   TCP     60 80 → 1161 [ACK] Seq=1 Ack=566 W
21:44:20.624407 192.168.1.102   128.119.245.12   TCP   1514 1161 → 80 [ACK] Seq=2026 Ack=1
21:44:20.625071 192.168.1.102   128.119.245.12   TCP   1514 1161 → 80 [ACK] Seq=3486 Ack=1
21:44:20.647675 128.119.245.12  192.168.1.102   TCP     60 80 → 1161 [ACK] Seq=1 Ack=2026
21:44:20.647786 192.168.1.102   128.119.245.12   TCP   1514 1161 → 80 [ACK] Seq=4946 Ack=1
21:44:20.648538 192.168.1.102   128.119.245.12   TCP   1514 1161 → 80 [ACK] Seq=6406 Ack=1
21:44:20.694466 128.119.245.12  192.168.1.102   TCP     60 80 → 1161 [ACK] Seq=1 Ack=3486
21:44:20.694566 192.168.1.102   128.119.245.12   TCP   1201 1161 → 80 [PSH, ACK] Seq=7866 A
21:44:20.739499 128.119.245.12  192.168.1.102   TCP     60 80 → 1161 [ACK] Seq=1 Ack=4946
21:44:20.787680 128.119.245.12  192.168.1.102   TCP     60 80 → 1161 [ACK] Seq=1 Ack=6406
21:44:20.838183 128.119.245.12  192.168.1.102   TCP     60 80 → 1161 [ACK] Seq=1 Ack=7866
```

（令α为 0.125,）得到下表：

i	Seq	发送时间	收到 ACK 时间	RTT(ms)	EstimatedRTT(ms)
1	1	21:44:20.596858	21:44:20.624318	27.5	27.5
2	566	21:44:20.612118	21:44:20.647675	35.6	28.5125
3	2026	21:44:20.624407	21:44:20.694466	70.1	33.7109375
4	3486	21:44:20.625071	21:44:20.737499	114.4	43.79707031
5	4946	21:44:20.647786	21:44:20.787680	139.9	55.80993652
6	6406	21:44:20.648538	21:44:20.838183	189.7	72.54619446

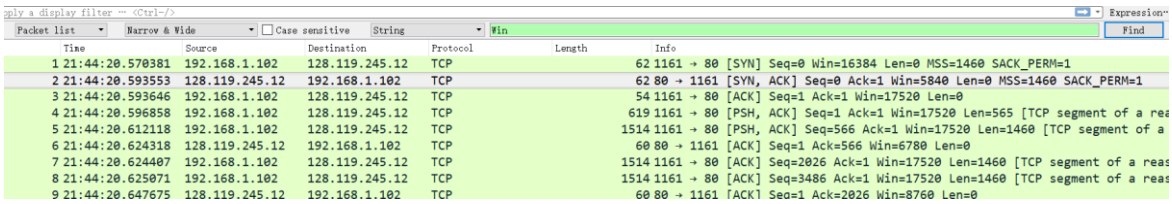
8. What is the length of each of the first six TCP segments?

答：分别是565, 1460, 1460, 1460, 1460, 1460。如下图：

```
L7520 Len=565 [TC
1=17520 Len=1460
) Len=0
520 Len=1460 [TCP
520 Len=1460 [TCP
50 Len=0
520 Len=1460 [TCP
520 Len=1460 [TCP
```

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：服务器最小的缓冲空间（Win）为 5840 字节。全程服务器端的 Win 在慢慢变大，最后变到 62780 字节，没有 throttle 发送端。



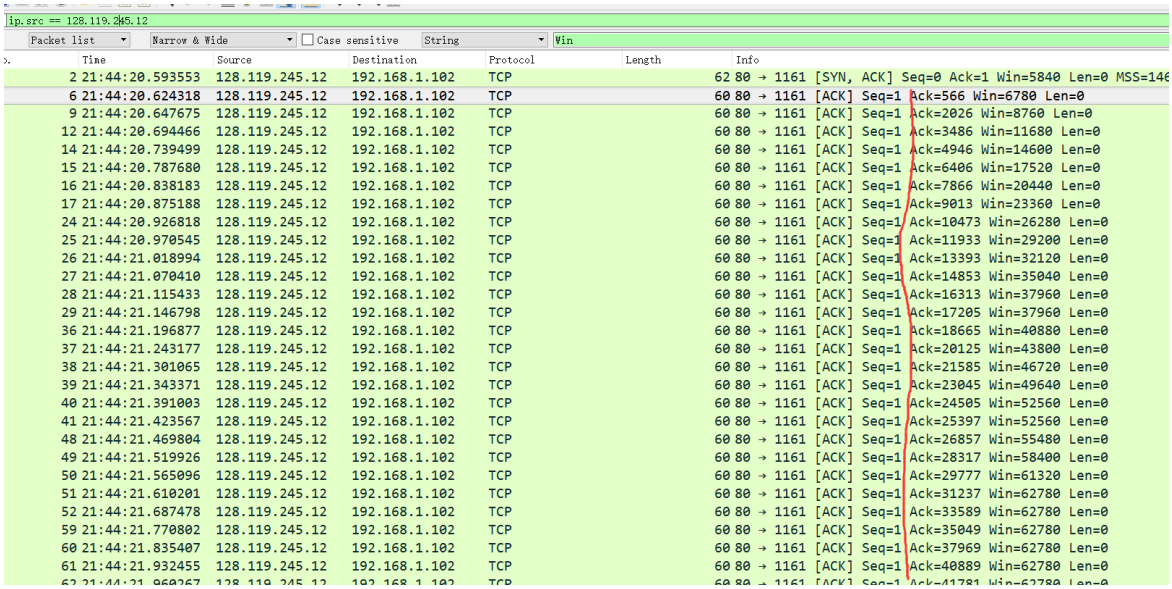
	Time	Source	Destination	Protocol	Length	Info
1	21:44:20.570381	192.168.1.102	128.119.245.12	TCP	62	1161 → 80 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM=1
2	21:44:20.593553	128.119.245.12	192.168.1.102	TCP	62	80 → 1161 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM=1
3	21:44:20.593646	192.168.1.102	128.119.245.12	TCP	54	1161 → 80 [ACK] Seq=1 Ack=1 Win=17520 Len=0
4	21:44:20.596858	192.168.1.102	128.119.245.12	TCP	619	1161 → 80 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=565 [TCP segment of a rea
5	21:44:20.612118	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [PSH, ACK] Seq=566 Ack=1 Win=17520 Len=1460 [TCP segment of a
6	21:44:20.624318	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=566 Win=6780 Len=0
7	21:44:20.624407	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=2026 Ack=1 Win=17520 Len=1460 [TCP segment of a reas
8	21:44:20.625071	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=3486 Ack=1 Win=17520 Len=1460 [TCP segment of a reas
9	21:44:20.647675	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=2026 Win=8760 Len=0

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：没有重传的报文。检查发送端是否发送过两个具有相同Seq序号的报文，Seq序号是否严格单调递增。

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 257 in the text).

答：典型的是1460和2920字节。此外还有566，1147，892，2352（892+1460）等。一次ACK所acknowledge的字节数根据相邻Ack的差值算出。一次确认两个报文的有确认2920和2352字节的。比如trace文件中第52个报文，就是确认了2352字节；比如第60个报文，就是确认了2920字节。



	Time	Source	Destination	Protocol	Length	Info
2	21:44:20.593553	128.119.245.12	192.168.1.102	TCP	62	80 → 1161 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=146
6	21:44:20.624318	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=566 Win=6780 Len=0
9	21:44:20.647675	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=2026 Win=8760 Len=0
12	21:44:20.694466	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=3486 Win=11680 Len=0
14	21:44:20.739499	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=4946 Win=14600 Len=0
15	21:44:20.787680	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=6406 Win=17520 Len=0
16	21:44:20.838183	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=7866 Win=20440 Len=0
17	21:44:20.875188	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=9013 Win=23360 Len=0
24	21:44:20.926818	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=10473 Win=26280 Len=0
25	21:44:20.970545	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=11933 Win=29200 Len=0
26	21:44:21.018994	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=13393 Win=32120 Len=0
27	21:44:21.070410	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=14853 Win=35040 Len=0
28	21:44:21.115433	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=16313 Win=37960 Len=0
29	21:44:21.146798	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=17205 Win=37960 Len=0
36	21:44:21.196877	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=18665 Win=40880 Len=0
37	21:44:21.243177	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=20125 Win=43800 Len=0
38	21:44:21.301065	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=21585 Win=46720 Len=0
39	21:44:21.343371	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=23045 Win=49640 Len=0
40	21:44:21.391003	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=24505 Win=52560 Len=0
41	21:44:21.423567	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=25397 Win=52560 Len=0
48	21:44:21.469804	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=26857 Win=55480 Len=0
49	21:44:21.519926	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=28317 Win=58400 Len=0
50	21:44:21.565096	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=29777 Win=61320 Len=0
51	21:44:21.610201	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=31237 Win=62780 Len=0
52	21:44:21.687478	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=33589 Win=62780 Len=0
59	21:44:21.770802	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=35049 Win=62780 Len=0
60	21:44:21.835407	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=37969 Win=62780 Len=0
61	21:44:21.932455	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=40889 Win=62780 Len=0

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

答：考虑第一次发送post到发送端收到最后一条Ack这一过程。

总的时长t = 21:44:26.026211-21:44:20.596858=5.429353（s）。

所传递的字节数w = 164091-1=164090（Bytes）。

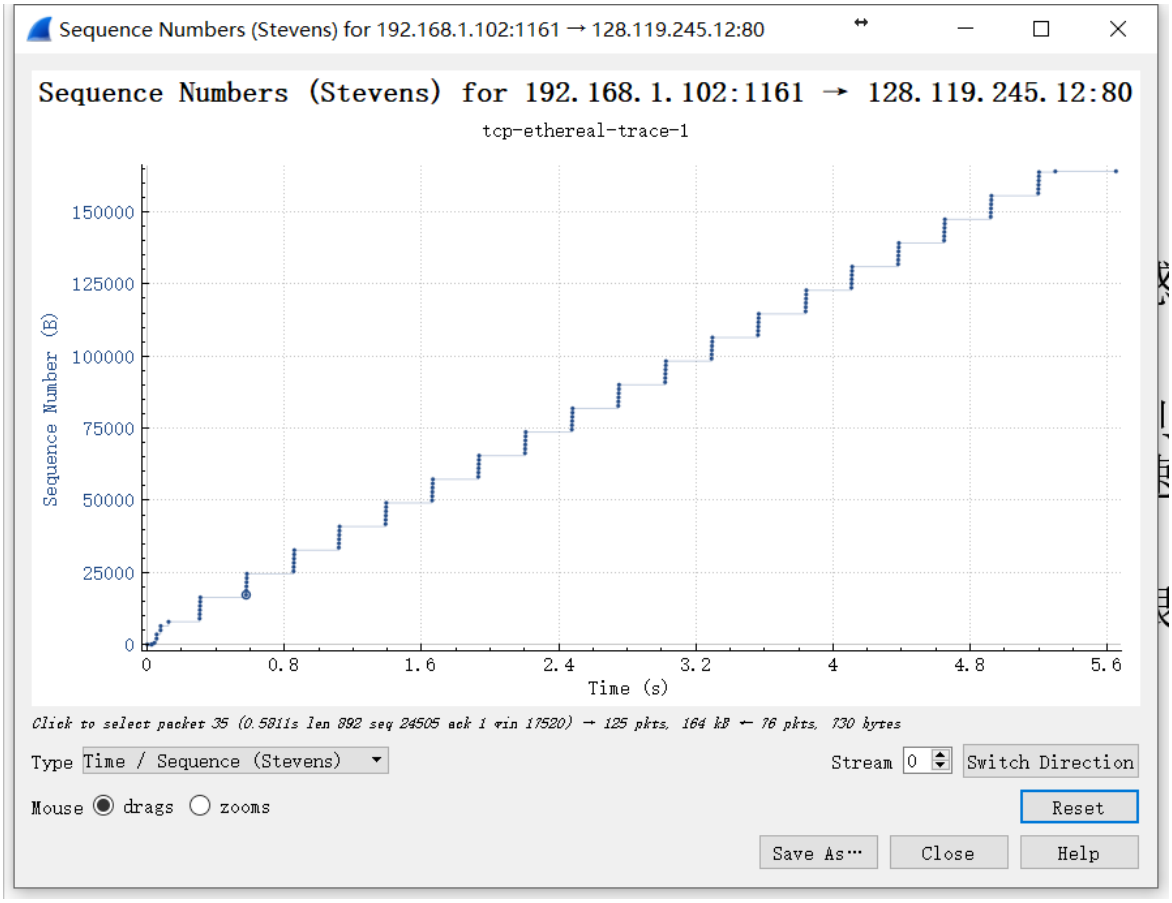
吞吐量为 w/t = 30222 Bps=30.222 KBps

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： 慢启动在post发送开始时开始，但是看不出何时结束，也看不出拥塞避免何时开始。因为从这个trace可以看出没有出现3次冗余ACK或者超时，也就没有丢包，没有拥塞发生。

不同点：

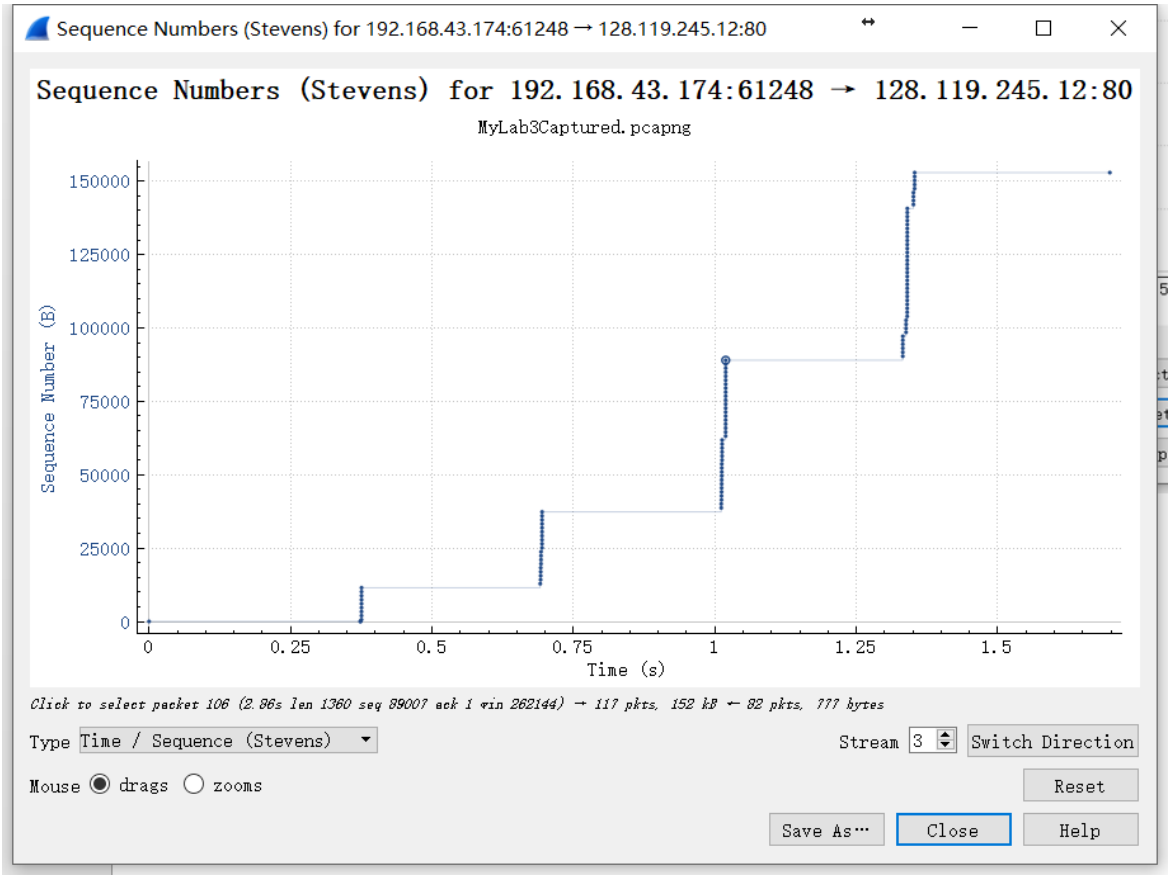
不同于课本上所说的先第一次慢启动，等到拥塞时再第二次进入慢启动同时调整ssthresh，cwnd值，等到cwnd大于等于ssthresh时进入拥塞避免这样的策略，我们的trace文件中没有冗余ACK，没有丢包，没有拥塞，没有以上的状态转换。而是：数据发送的速率被严格限制在了第一次慢启动结束前，并且到了后面一直重复这一过程：连续发送1460*5+892*1字节=8192字节的包就会暂停发送，直到这8192字节被全部ACK。这一数据传输过程，应该是被应用程序所严格的控制着，而不仅仅依赖于TCP自身的拥塞控制机制。



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：同样，慢启动在post发送开始时开始，但是看不出何时结束，因为从我下面的截图可以看出cwnd一直再指数增长；同时，我的trace文件中可以看出，除了两个一次的冗余ACK之外，没有三次冗余ACK，也就没有丢包，没有拥塞，所以慢启动没有结束，所以拥塞避免也没有开始。

不同点：我这里实际的情况是：在慢启动的开始，cwnd不是从1个MSS开始，而是以大约9个MSS（MSS=1360Bytes）开始，之后再指数递增。并且我这里还没有慢启动结束，就已发送完所有的数据。


Sequence Numbers (Stevens) for 192.168.43.174:61248 → 128.119.245.12:80
Sequence Numbers (Stevens) for 192.168.43.174:61248 → 128.119.245.12:80
MyLab3Captured.pcapng
Sequence Number (B)
150000
125000
100000
75000
50000
25000
0
0
0.25
0.5
0.75
1
1.25
1.5
Time (s)
Click to select packet 106 (2.86s len 1360 seq 89007 ack 1 win 262144) → 117 pkts, 152 kB ← 82 pkts, 777 bytes
Type Time / Sequence (Stevens)
Stream 3
Switch Direction
Mouse drags zooms
Reset
Save As…
Close
Help