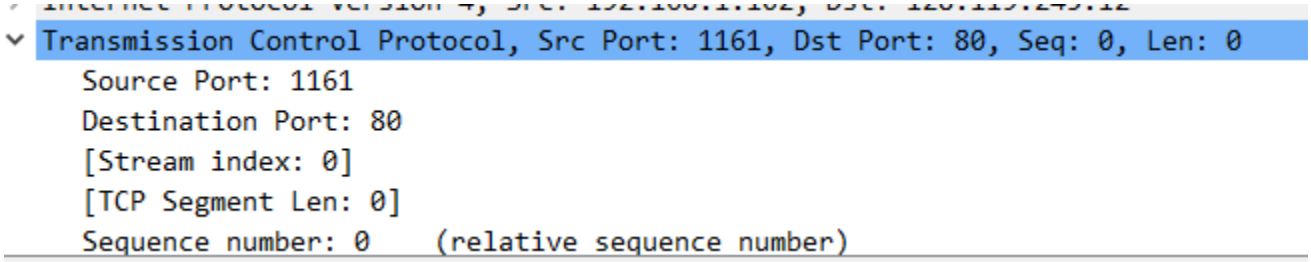
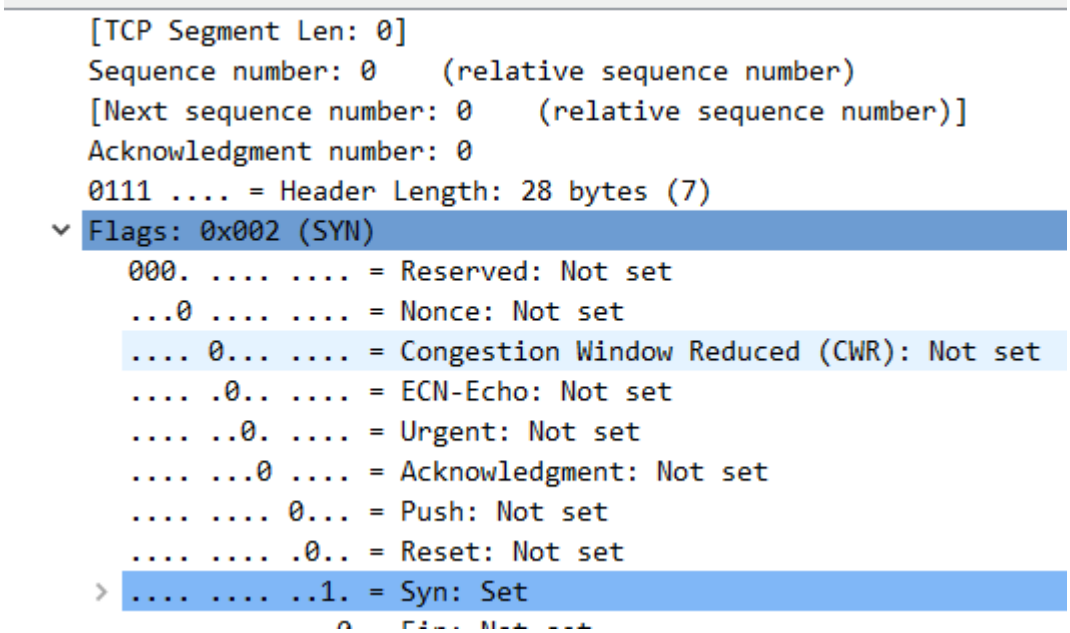
1. What is the IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu? To answer this question, it's probably easiest to select an HTTP message and explore the details of the TCP packet used to carry this HTTP message, using the "details 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.
   192.168.1.102；1161
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?
   128.119.245.12；80

If you have been able to create your own trace, answer the following question:

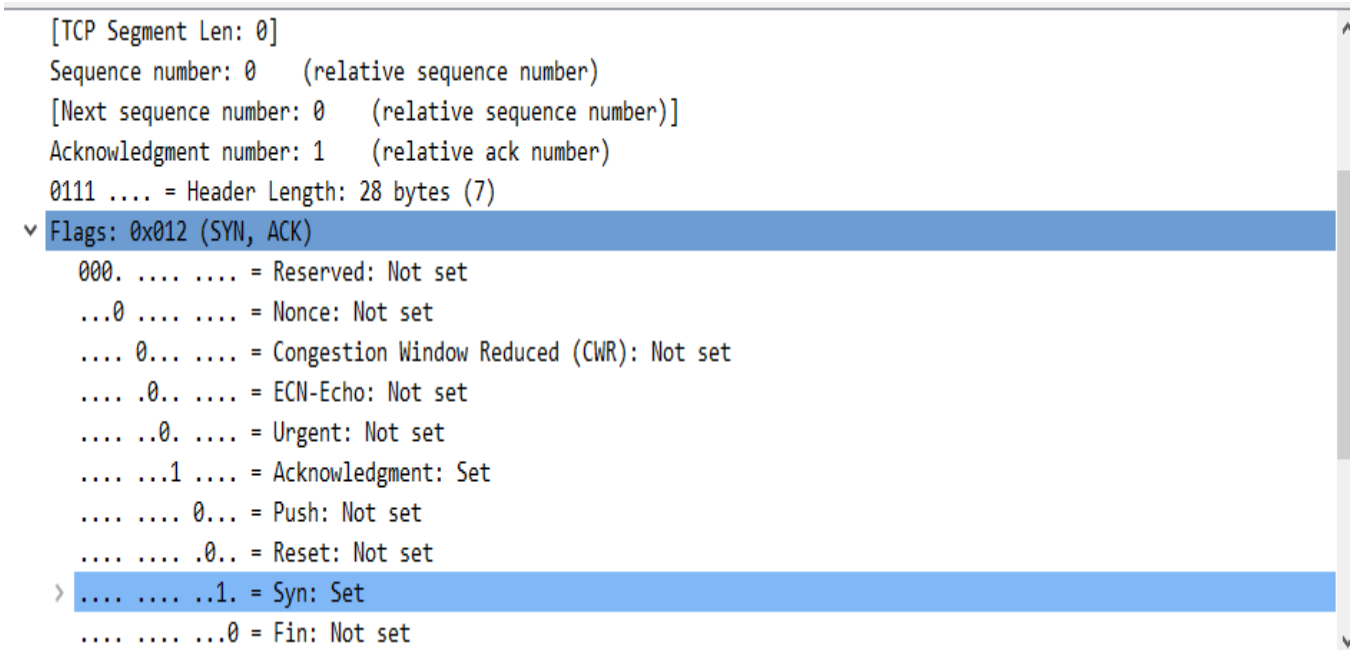
3. What is the IP address and TCP port number used by your client computer (source) to transfer the file to gaia.cs.umass.edu?
   用的是实验提供的抓包文件



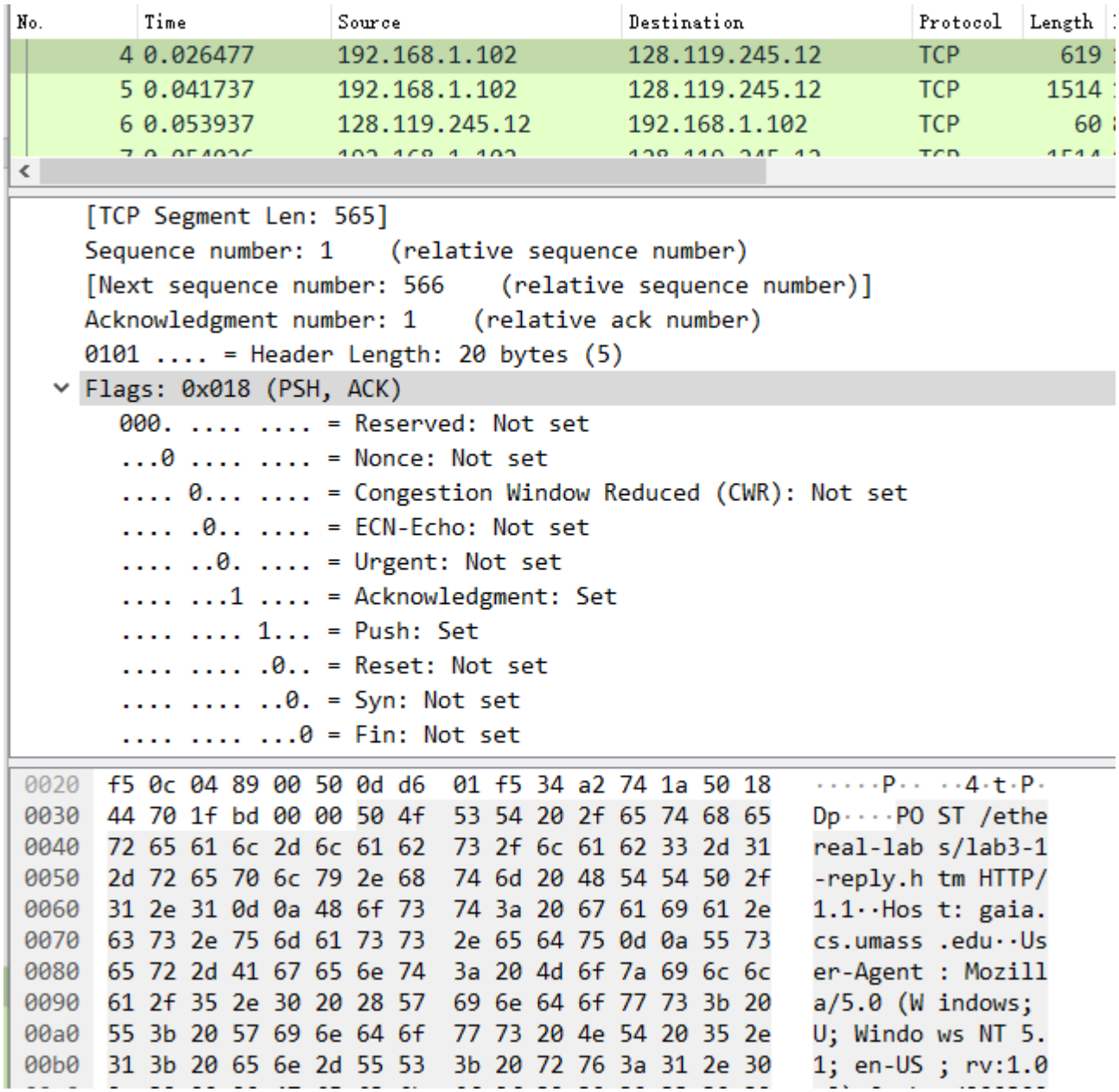
4. What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu? What is it in the segment that identifies the segment as a SYN segment?
   序列号为 0；设置为 1 的 SYN，flag 将标识段标识为 SYN 段



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?
   0；1；设置为 1 的 SYN；flag 将段标识为 SYNACK 段

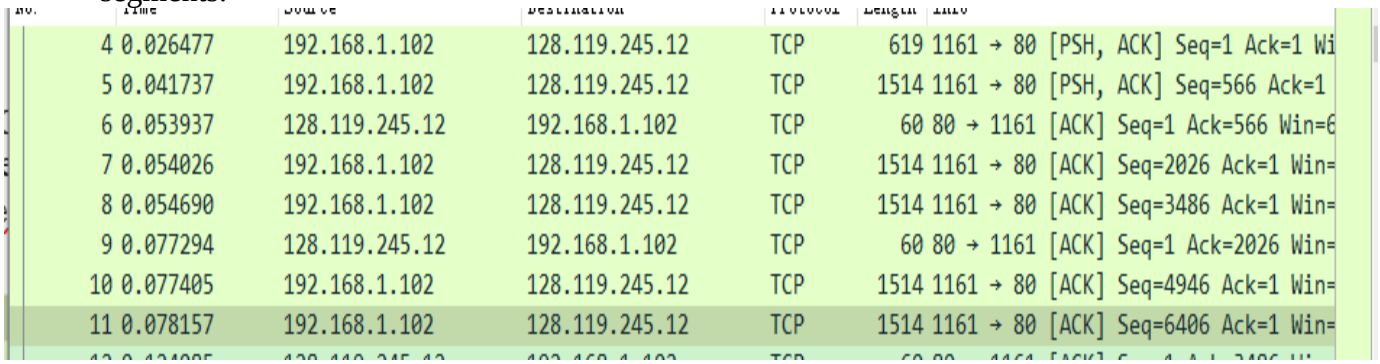


6. What is the sequence number of the TCP segment containing the HTTP POST command? Note that in order to find the POST command, you'll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with a "POST" within its DATA field.
   4



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

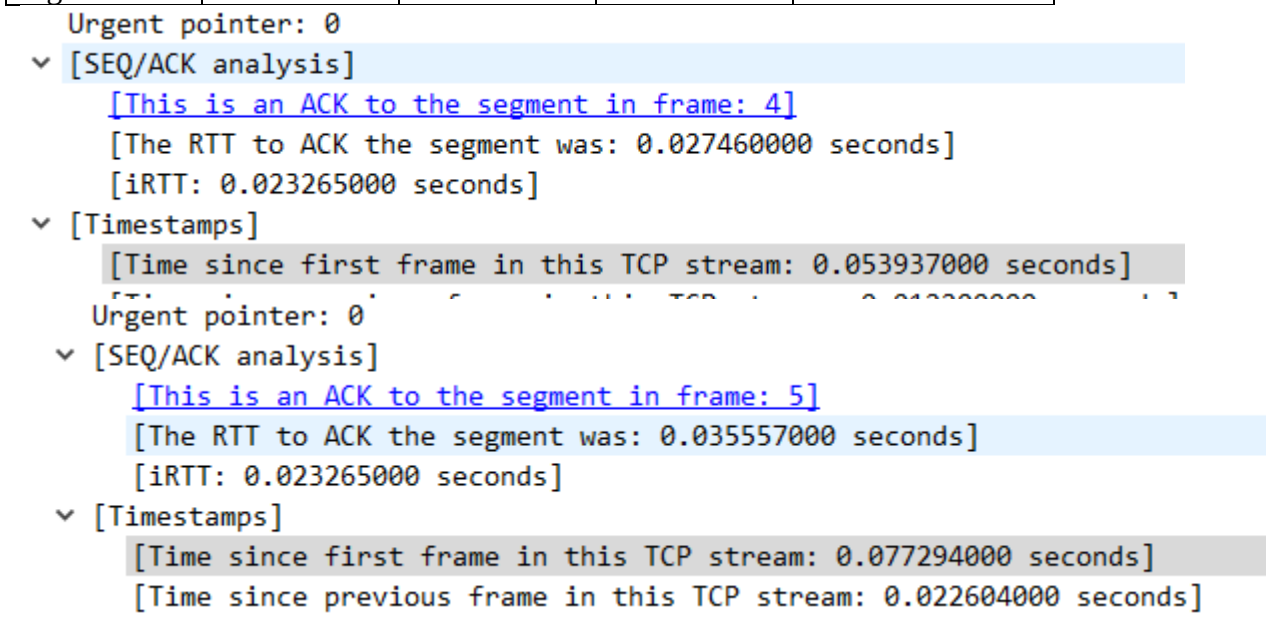
computed using the `EstimatedRTT` equation on page 242 for all subsequent segments.



No.	Time	Source	Destination	Protocol	Length	Info
4	0.026477	192.168.1.102	128.119.245.12	TCP	619	1161 → 80 [PSH, ACK] Seq=1 Ack=1 Wi
5	0.041737	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [PSH, ACK] Seq=566 Ack=1
6	0.053937	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=566 Win=6
7	0.054026	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=2026 Ack=1 Win=
8	0.054690	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=3486 Ack=1 Win=
9	0.077294	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=2026 Win=
10	0.077405	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=4946 Ack=1 Win=
11	0.078157	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=6406 Ack=1 Win=

前六段段号分别是 1、566、2026、3486、4906、6406

	sent time	ack time	RTT	estimated RTT
segment 1	0.026477	0.053937	0.02746	0.02746
segment 2	0.041737	0.077294	0.035557	0.028472125
segment 3	0.054026	0.124085	0.070059	0.033670484375
segment 4	0.05469	0.169118	0.114428	0.0437651738281
segment 5	0.077405	0.217299	0.139894	0.0557812770995
segment 6	0.078157	0.267802	0.189645	0.072514242462



```
Urgent pointer: 0
▾ [SEQ/ACK analysis]
    [This is an ACK to the segment in frame: 4]
    [The RTT to ACK the segment was: 0.027460000 seconds]
    [iRTT: 0.023265000 seconds]
▾ [Timestamps]
    [Time since first frame in this TCP stream: 0.053937000 seconds]
Urgent pointer: 0
▾ [SEQ/ACK analysis]
    [This is an ACK to the segment in frame: 5]
    [The RTT to ACK the segment was: 0.035557000 seconds]
    [iRTT: 0.023265000 seconds]
▾ [Timestamps]
    [Time since first frame in this TCP stream: 0.077294000 seconds]
    [Time since previous frame in this TCP stream: 0.022604000 seconds]
```

```
    Urgent pointer: 0
  ▾ [SEQ/ACK analysis]
      [This is an ACK to the segment in frame: 7]
      [The RTT to ACK the segment was: 0.070059000 seconds]
      [iRTT: 0.023265000 seconds]
  ▾ [Timestamps]
      [Time since first frame in this TCP stream: 0.124085000 seconds]
      [Time since previous frame in this TCP stream: 0.045928000 seconds]
```

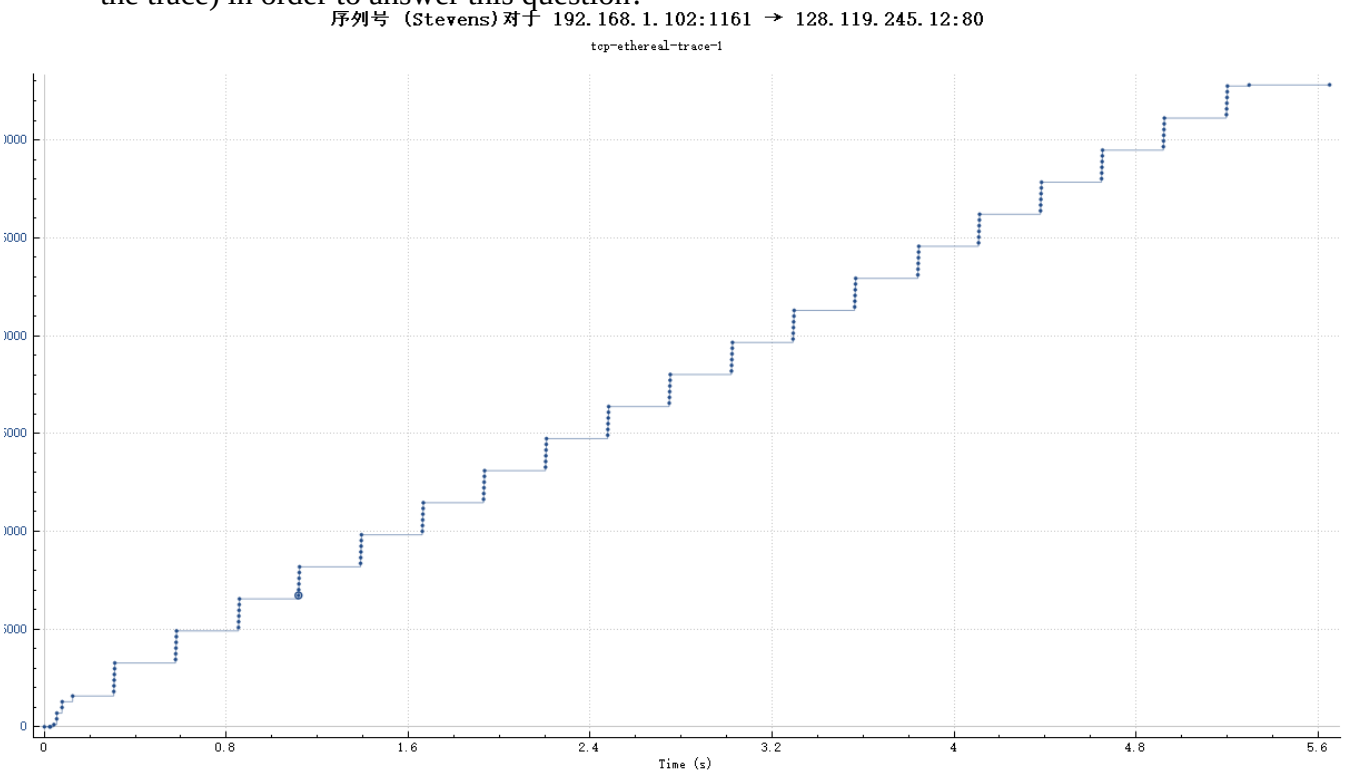
```
Urgent pointer: 0
▾ [SEQ/ACK analysis]
    [This is an ACK to the segment in frame: 8]
    [The RTT to ACK the segment was: 0.114428000 seconds]
    [iRTT: 0.023265000 seconds]
▾ [Timestamps]
    [Time since first frame in this TCP stream: 0.169118000 seconds]
    [Time since previous frame in this TCP stream: 0.044933000 seconds]
```

```
  Urgent pointer: 0
▾ [SEQ/ACK analysis]
    [This is an ACK to the segment in frame: 10]
    [The RTT to ACK the segment was: 0.139894000 seconds]
    [iRTT: 0.023265000 seconds]
▾ [Timestamps]
    [Time since first frame in this TCP stream: 0.217299000 seconds]
    [Time since previous frame in this TCP stream: 0.048181000 seconds]
    Urgent pointer: 0
▾ [SEQ/ACK analysis]
    [This is an ACK to the segment in frame: 11]
    [The RTT to ACK the segment was: 0.189645000 seconds]
    [iRTT: 0.023265000 seconds]
▾ [Timestamps]
    [Time since first frame in this TCP stream: 0.267802000 seconds]
    [Time since previous frame in this TCP stream: 0.050503000 seconds]
```

8. What is the length of each of the first six TCP segments?[1]
   619；1514；1514；1514；1514；1514
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?
   5840，并且稳步增长至窗口容量的最大值（62780）;没有受限制

```
Acknowledgment number: 1    (relative ack number
0111 .... = Header Length: 28 bytes (7)
Flags: 0x012 (SYN, ACK)
Window size value: 5840
[Calculated window size: 5840]
Checksum: 0x774d [unverified]
[Checksum Status: Unverified]
```

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



从这个序列号中可看出，没有重传报文段

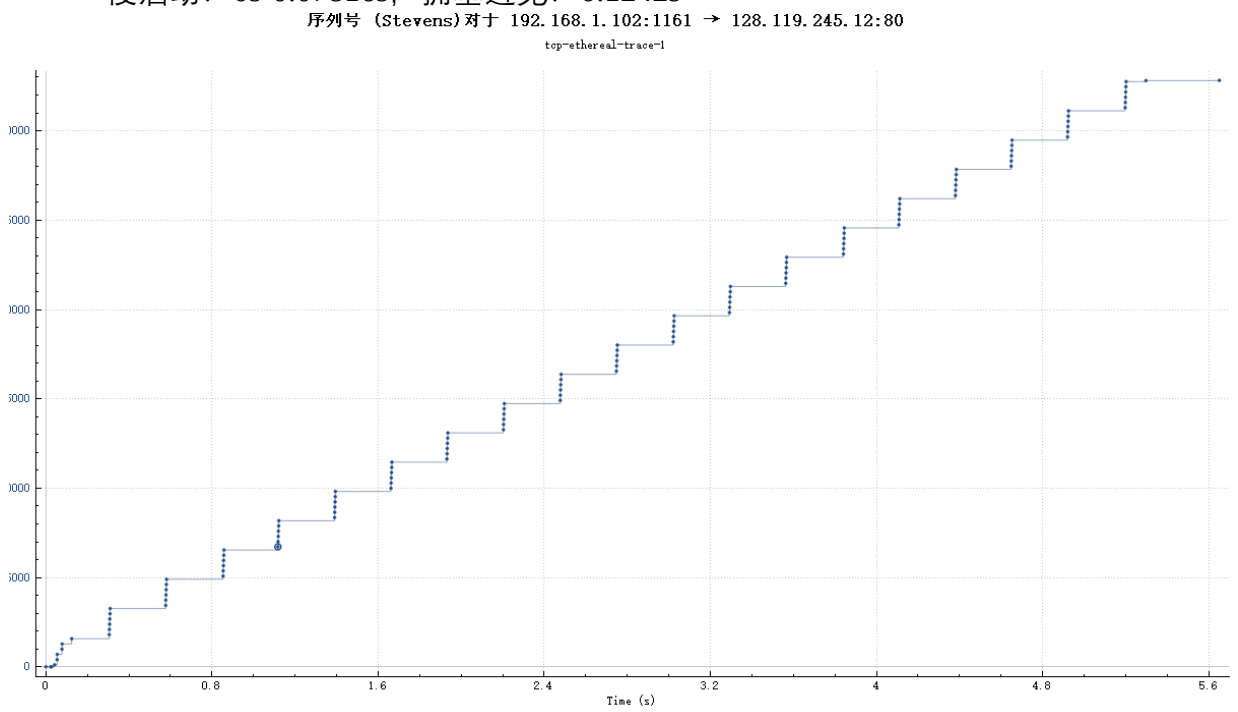
11. How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment (see Table 3.2 on page 250 in the text).
    ACK1 接收 566 字节，ACK2-ACK6=1460，接收 1460 字节，通过进一步计算，可发现间隔刚好为 2 倍的 1460，说明每一个 ack 一个
12. What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.

传输的总数据量为 149450bytes，总时间为 2.189526 s， 故吞吐量为：68.256 KBytes/sec

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

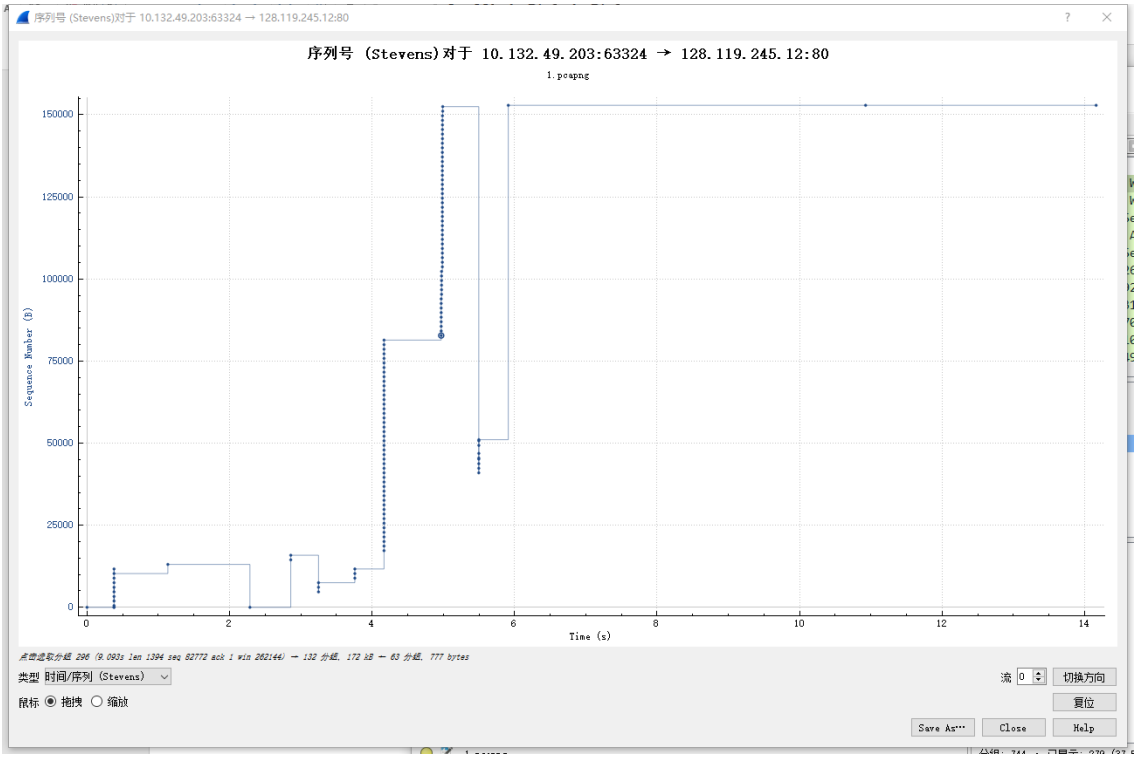
which the measured data differs from the idealized behavior of TCP that we've studied in the text.

慢启动：0s-0.07816s，拥塞避免：0.1242s



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

自己抓包的结果如下，从图上看不出有慢启动阶段，只有拥塞避免和快速恢复阶段。


序列号 (Stevens)对于 10.132.49.203:63324 → 128.119.245.12:80
序列号 (Stevens)对于 10.132.49.203:63324 → 128.119.245.12:80
1.pcapng
Sequence Number (B)
Time (s)
类型 时间/序列 (Stevens)
流 0
切换方向
鼠标 拖拽 缩放
复位
Save As···
Close
Help