- 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

[TCP Segment Len: 0]

```
Transmission Control Protocol, Src Port: 1161, Dst Port: 80, Seq: 0, Len: 0
Source Port: 1161
Destination Port: 80
[Stream index: 0]
[TCP Segment Len: 0]
Sequence number: 0 (relative sequence number)
```

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

```
(relative sequence number)
  Sequence number: 0
  [Next sequence number: 0
                             (relative sequence number)]
  Acknowledgment number: 0
  0111 .... = Header Length: 28 bytes (7)
Flags: 0x002 (SYN)
    000. .... = Reserved: Not set
    ...0 .... = Nonce: Not set
    .... 0... = Congestion Window Reduced (CWR): Not set
    .... .0.. .... = ECN-Echo: Not set
    .... ..0. .... = Urgent: Not set
    .... ...0 .... = Acknowledgment: Not set
    .... .... 0... = Push: Not set
    .... .... .0.. = Reset: Not set
  > .... .... ..1. = Syn: Set
```

- 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

```
[TCP Segment Len: 0]
  Sequence number: 0
                       (relative sequence number)
  [Next sequence number: 0
                             (relative sequence number)]
  Acknowledgment number: 1
                             (relative ack number)
  0111 .... = Header Length: 28 bytes (7)
Flags: 0x012 (SYN, ACK)
    000. .... = Reserved: Not set
    ...0 .... = Nonce: Not set
    .... 0... = Congestion Window Reduced (CWR): Not set
    .... .0.. .... = ECN-Echo: Not set
    .... ..0. .... = Urgent: Not set
    .... ...1 .... = Acknowledgment: Set
    .... 0... = Push: Not set
    .... .... .0.. = Reset: Not set
  > .... .... ..1. = Syn: Set
    .... .... 0 = Fin: Not set
```

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.

```
No.
         Time
                       Source
                                            Destination
                                                                 Protocol
                                                                         Length :
       4 0.026477
                       192.168.1.102
                                            128.119.245.12
                                                                 TCP
                                                                            619
       5 0.041737
                       192.168.1.102
                                            128.119.245.12
                                                                 TCP
                                                                           1514
       6 0.053937
                       128.119.245.12
                                            192.168.1.102
                                                                             60
                                                                 TCP
       7 0 05/000
                       100 400 4 400
                                            100 110 015 10
                                                                 TCD
                                                                           4544
    [TCP Segment Len: 565]
    Sequence number: 1
                           (relative sequence number)
                                   (relative sequence number)]
    [Next sequence number: 566
    Acknowledgment number: 1
                                 (relative ack number)
    0101 .... = Header Length: 20 bytes (5)
  Flags: 0x018 (PSH, ACK)
       000. .... = Reserved: Not set
       ...0 .... = Nonce: Not set
       .... 0... = Congestion Window Reduced (CWR): Not set
       .... .0.. .... = ECN-Echo: Not set
       .... ..0. .... = Urgent: Not set
       .... - 1 .... = Acknowledgment: Set
       .... - 1... = Push: Set
       .... .... .0.. = Reset: Not set
       .... .... ..0. = Syn: Not set
       .... .... 0 = Fin: Not set
                                                          .....P.. ..4.t.P.
      f5 0c 04 89 00 50 0d d6
                               01 f5 34 a2 74 1a 50 18
0020
0030
     44 70 1f bd 00 00 50 4f
                               53 54 20 2f 65 74 68 65
                                                         Dp····PO ST /ethe
      72 65 61 6c 2d 6c 61 62
                               73 2f 6c 61 62 33 2d 31
                                                          real-lab s/lab3-1
0040
0050 2d 72 65 70 6c 79 2e 68 74 6d 20 48 54 54 50 2f
                                                          -reply.h tm HTTP/
0060
      31 2e 31 0d 0a 48 6f 73 74 3a 20 67 61 69 61 2e
                                                         1.1··Hos t: gaia.
0070
     63 73 2e 75 6d 61 73 73
                               2e 65 64 75 0d 0a 55 73
                                                          cs.umass .edu··Us
     65 72 2d 41 67 65 6e 74
                               3a 20 4d 6f 7a 69 6c 6c
0080
                                                          er-Agent : Mozill
0090 61 2f 35 2e 30 20 28 57 69 6e 64 6f 77 73 3b 20
                                                          a/5.0 (W indows;
     55 3b 20 57 69 6e 64 6f
                               77 73 20 4e 54 20 35 2e
                                                         U; Windo ws NT 5.
00a0
00b0
      31 3b 20 65 6e 2d 55 53
                               3b 20 72 76 3a 31 2e 30
                                                          1; en-US ; rv:1.0
```

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

	гло.	segments.	150ш се	Destination	11010001	Persin Into
- 1	210.	Trine	504100	Destination	11010001	-
		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
	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=
1		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=
		42.0.424000	400 440 045 40	400 400 4 400	TCD	CO DO 44C4 [ACK] C 4 A L 340C III

## 前六段段号分别是1、566、2026、3486、4906、6406

1137 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1								
	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]
   orgenic pointer: o
 「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?<sup>1</sup>
     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

ACKNOWIEGBMENT NUMBER: I (Lefacine ack unmber

0111 .... = Header Length: 28 bytes (7)

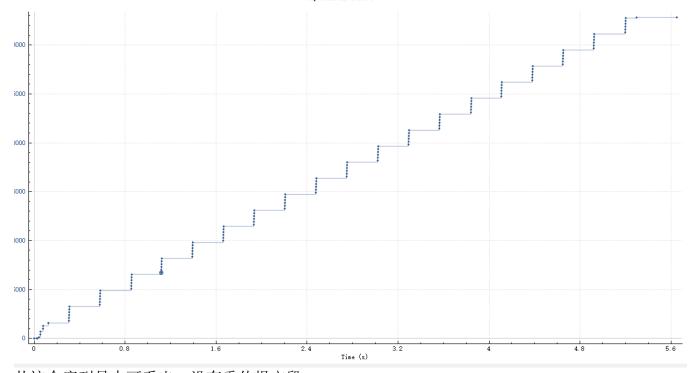
> Flags: 0x012 (SYN, ACK)

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?

序列号 (Stevens)对于 192.168.1.102:1161 → 128.119.245.12:80



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

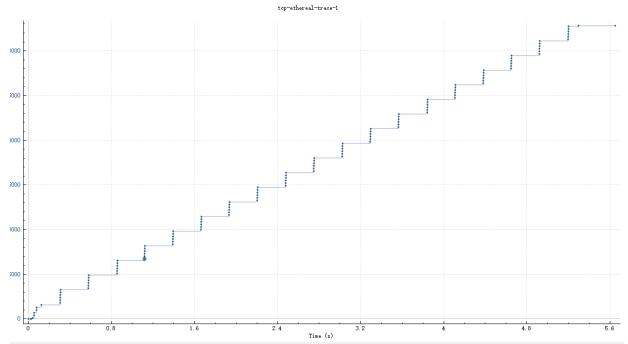
12. What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.

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 序列号 (Stevens)对+ 192.168.1.102:1161 → 128.119.245.12:80



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。

