计网实验 5

姓名：岳宇轩　学号：19020011038　指导老师：洪峰

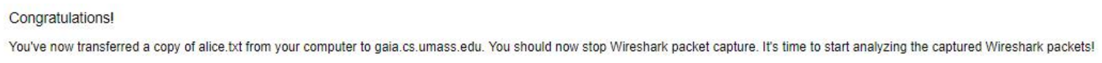
1. Capturing a bulk TCP transfer from your computer to a remote server

打开 http://gaia.cs.umass.edu/wiresharklabs/alice.txt，并将其中内容保存为 alice.txt。

打开 http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html，选择刚刚保存的 alice.txt 文件

打开 wireshark 进行抓包，随后点击页面中的 Upload alice.txt file

页面提示



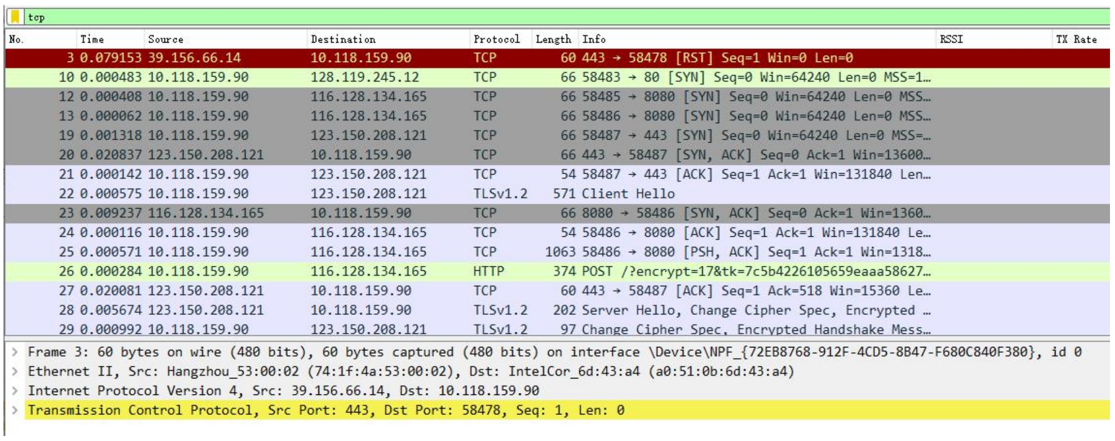
Congratulations!

You've now transferred a copy of alice.txt from your computer to gaia.cs.umass.edu. You should now stop Wireshark packet capture. It's time to start analyzing the captured Wireshark packets!

后停止 wireshark 抓包

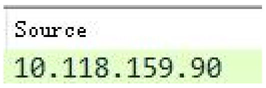
2. A first look at the captured trace

在筛选框中输入 tcp 得到下图所示结果

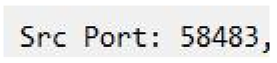


```
tcp
No.   Time     Source           Destination      Protocol Length Info                                                    RSSI  TX Rate
    3 0.079153 39.156.66.14     10.118.159.90    TCP        60 443 → 58478 [RST] Seq=1 Win=0 Len=0
   10 0.000483 10.118.159.90    128.119.245.12   TCP        66 58483 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1…
   12 0.000408 10.118.159.90    116.128.134.165  TCP        66 58485 → 8080 [SYN] Seq=0 Win=64240 Len=0 MSS…
   13 0.000062 10.118.159.90    116.128.134.165  TCP        66 58486 → 8080 [SYN] Seq=0 Win=64240 Len=0 MSS…
   19 0.001318 10.118.159.90    123.150.208.121  TCP        66 58487 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=…
   20 0.020837 123.150.208.121  10.118.159.90    TCP        66 443 → 58487 [SYN, ACK] Seq=0 Ack=1 Win=13600…
   21 0.000142 10.118.159.90    123.150.208.121  TCP        54 58487 → 443 [ACK] Seq=1 Ack=1 Win=131840 Len…
   22 0.000575 10.118.159.90    123.150.208.121  TLSv1.2   571 Client Hello
   23 0.009237 116.128.134.165  10.118.159.90    TCP        66 8080 → 58486 [SYN, ACK] Seq=0 Ack=1 Win=1360…
   24 0.000116 10.118.159.90    116.128.134.165  TCP        54 58486 → 8080 [ACK] Seq=1 Ack=1 Win=131840 Le…
   25 0.000571 10.118.159.90    116.128.134.165  TCP      1063 58486 → 8080 [PSH, ACK] Seq=1 Ack=1 Win=1318…
   26 0.000284 10.118.159.90    116.128.134.165  HTTP      374 POST /?encrypt=17&tk=7c5b4226105659eaaa58627…
   27 0.020081 123.150.208.121  10.118.159.90    TCP        60 443 → 58487 [ACK] Seq=1 Ack=518 Win=15360 Le…
   28 0.005674 123.150.208.121  10.118.159.90    TLSv1.2   202 Server Hello, Change Cipher Spec, Encrypted …
   29 0.000992 10.118.159.90    123.150.208.121  TLSv1.2    97 Change Cipher Spec, Encrypted Handshake Mess…

> Frame 3: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{72EB8768-912F-4CD5-8B47-F680C840F380}, id 0
> Ethernet II, Src: Hangzhou_53:00:02 (74:1f:4a:53:00:02), Dst: IntelCor_6d:43:a4 (a0:51:0b:6d:43:a4)
> Internet Protocol Version 4, Src: 39.156.66.14, Dst: 10.118.159.90
> Transmission Control Protocol, Src Port: 443, Dst Port: 58478, Seq: 1, Len: 0
```

1. What is the IP address and TCP port number used by the client computer (source) that is transferring the alice.txt file to gaia.cs.umass.edu?



Source
10.118.159.90

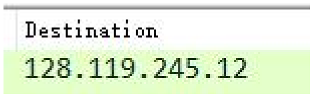


Src Port: 58483,

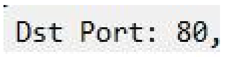
（1）IP address ： 10.118.159.90

（2）TCP port ： 58483

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?



```
Destination
128.119.245.12
```



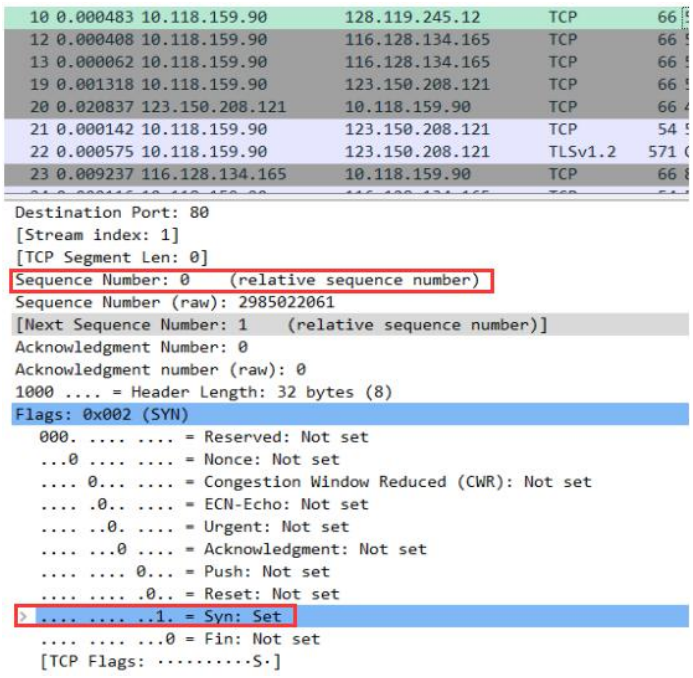
```
Dst Port: 80,
```

（1）IP address ： 128.119.245.12

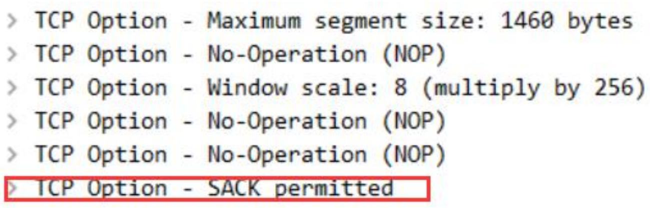
（2）port number ： 80

3. TCP Basics

3. 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 this TCP segment that identifies the segment as aSYN segment? Will the TCP receiver in this session be able to use SelectiveAcknowledgments (allowing TCP to function a bit more like a "selective repeat”'receiver, see section 3.4.5 in the text)?



```
10 0.000483 10.118.159.90      128.119.245.12     TCP      66
12 0.000408 10.118.159.90      116.128.134.165    TCP      66
13 0.000062 10.118.159.90      116.128.134.165    TCP      66
19 0.001318 10.118.159.90      123.150.208.121    TCP      66
20 0.020837 123.150.208.121    10.118.159.90      TCP      66
21 0.000142 10.118.159.90      123.150.208.121    TCP      54
22 0.000575 10.118.159.90      123.150.208.121    TLSv1.2  571
23 0.009237 116.128.134.165    10.118.159.90      TCP      66

Destination Port: 80
[Stream index: 1]
[TCP Segment Len: 0]
Sequence Number: 0    (relative sequence number)
Sequence Number (raw): 2985022061
[Next Sequence Number: 1    (relative sequence number)]
Acknowledgment Number: 0
Acknowledgment number (raw): 0
1000 .... = Header Length: 32 bytes (8)
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
   .... .... ...0 = Fin: Not set
   [TCP Flags: ··········S·]
```



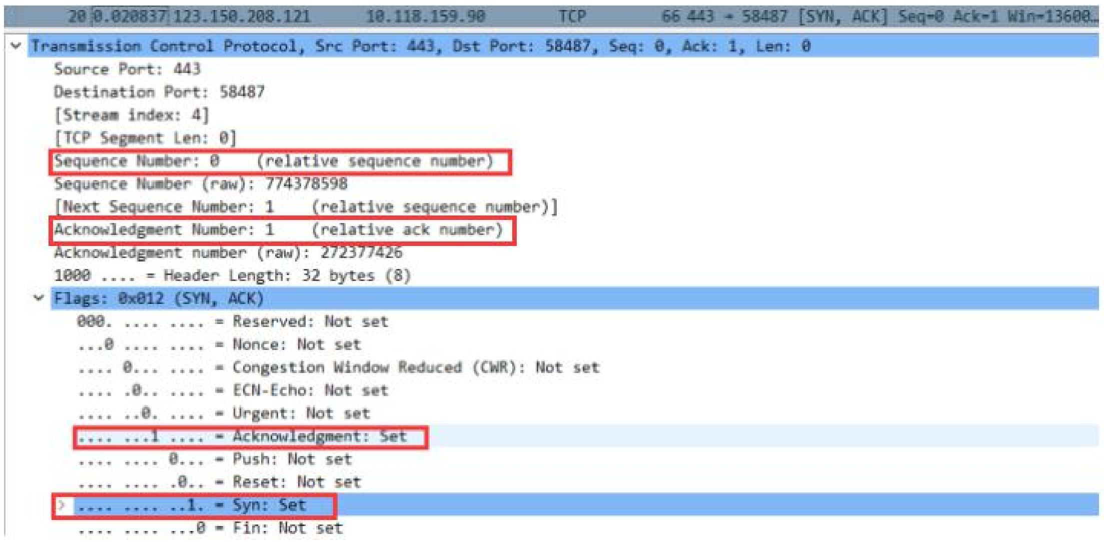
```
> TCP Option - Maximum segment size: 1460 bytes
> TCP Option - No-Operation (NOP)
> TCP Option - Window scale: 8 (multiply by 256)
> TCP Option - No-Operation (NOP)
> TCP Option - No-Operation (NOP)
> TCP Option - SACK permitted
```

（1）sequence number： 0

（2）Syn 标志被置为 1，表示这是一个 Syn segment

（3）允许 SelectiveAcknowledgments

4.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 it in the segment that identifies the segment as a SYNACK segment? What is the value of the Acknowledgement field in the SYNACK segment? How did gaia.cs.umass.edu determine that value?



```
20 0.020837 123.150.208.121      10.118.159.90        TCP        66 443 → 58487 [SYN, ACK] Seq=0 Ack=1 Win=13600…
Transmission Control Protocol, Src Port: 443, Dst Port: 58487, Seq: 0, Ack: 1, Len: 0
    Source Port: 443
    Destination Port: 58487
    [Stream index: 4]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 774378598
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 272377426
    1000 .... = Header Length: 32 bytes (8)
  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
```

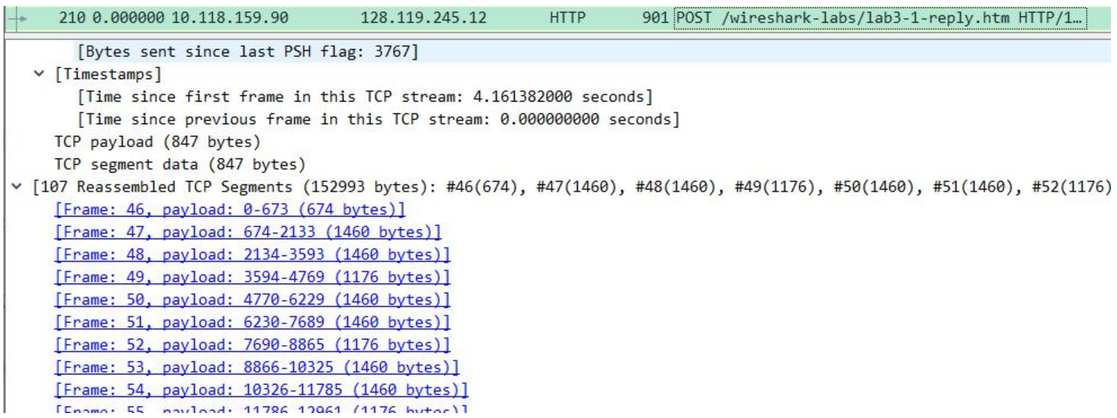
（1）sequence number：0

（2）Acknowledgement 和 Syn 标志都被置为 1

（3）the value of the Acknowledgement field in the SYNACK segment ： 1

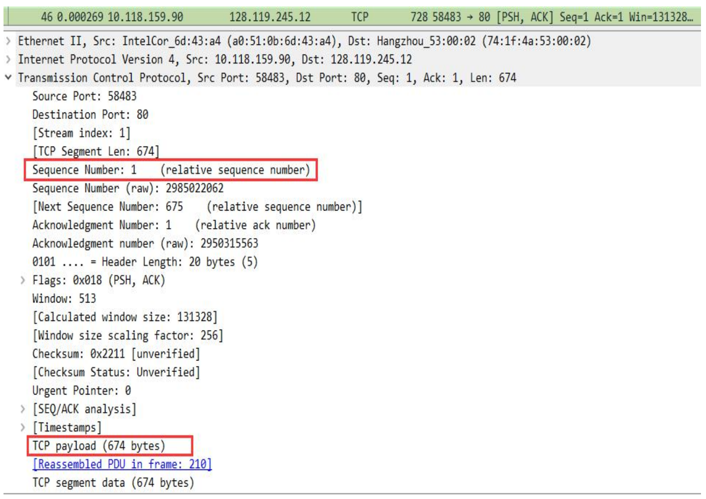
（4）对来自 client computer 的 initial sequence number of SYN 加一

5. What is the sequence number of the TCP segment containing the header of the HTTP POST command? Note that in order to find the POST message header,you'll need to dig into the packet content field at the bottom of the Wiresharkwindow, looking for a segment with the ASCl text "POST"within its DATAfield3+. How many bytes of data are contained in the payload (data) field of thisTCP segment? Did all of the data in the transferred file alice.txt fit into this single segment?



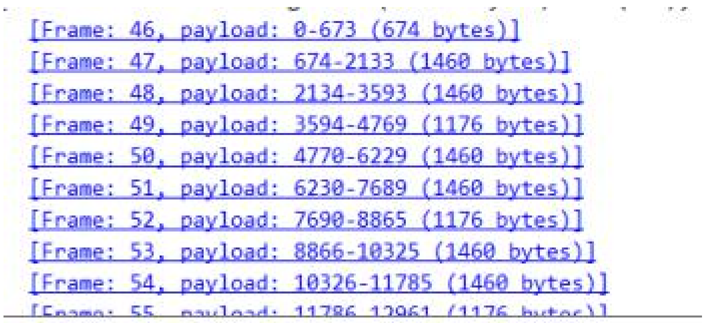
```
210 0.000000 10.118.159.90       128.119.245.12       HTTP       901 POST /wireshark-labs/lab3-1-reply.htm HTTP/1…
        [Bytes sent since last PSH flag: 3767]
    [Timestamps]
        [Time since first frame in this TCP stream: 4.161382000 seconds]
        [Time since previous frame in this TCP stream: 0.000000000 seconds]
    TCP payload (847 bytes)
    TCP segment data (847 bytes)
[107 Reassembled TCP Segments (152993 bytes): #46(674), #47(1460), #48(1460), #49(1176), #50(1460), #51(1460), #52(1176)
    [Frame: 46, payload: 0-673 (674 bytes)]
    [Frame: 47, payload: 674-2133 (1460 bytes)]
    [Frame: 48, payload: 2134-3593 (1460 bytes)]
    [Frame: 49, payload: 3594-4769 (1176 bytes)]
    [Frame: 50, payload: 4770-6229 (1460 bytes)]
    [Frame: 51, payload: 6230-7689 (1460 bytes)]
    [Frame: 52, payload: 7690-8865 (1176 bytes)]
    [Frame: 53, payload: 8866-10325 (1460 bytes)]
    [Frame: 54, payload: 10326-11785 (1460 bytes)]
    [Frame: 55, payload: 11786-12961 (1176 bytes)]
```

找到 http post 请求，frame:46 包含 http post 的 tcp segment



```
46 0.000269 10.118.159.90        128.119.245.12        TCP        728 58483 → 80 [PSH, ACK] Seq=1 Ack=1 Win=131328…
> Ethernet II, Src: IntelCor_6d:43:a4 (a0:51:0b:6d:43:a4), Dst: Hangzhou_53:00:02 (74:1f:4a:53:00:02)
> Internet Protocol Version 4, Src: 10.118.159.90, Dst: 128.119.245.12
v Transmission Control Protocol, Src Port: 58483, Dst Port: 80, Seq: 1, Ack: 1, Len: 674
    Source Port: 58483
    Destination Port: 80
    [Stream index: 1]
    [TCP Segment Len: 674]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 2985022062
    [Next Sequence Number: 675    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 2950315563
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x018 (PSH, ACK)
    Window: 513
    [Calculated window size: 131328]
    [Window size scaling factor: 256]
    Checksum: 0x2211 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > [SEQ/ACK analysis]
  > [Timestamps]
    TCP payload (674 bytes)
    [Reassembled PDU in frame: 210]
    TCP segment data (674 bytes)
```

根据下图可知



```
[Frame: 46, payload: 0-673 (674 bytes)]
[Frame: 47, payload: 674-2133 (1460 bytes)]
[Frame: 48, payload: 2134-3593 (1460 bytes)]
[Frame: 49, payload: 3594-4769 (1176 bytes)]
[Frame: 50, payload: 4770-6229 (1460 bytes)]
[Frame: 51, payload: 6230-7689 (1460 bytes)]
[Frame: 52, payload: 7690-8865 (1176 bytes)]
[Frame: 53, payload: 8866-10325 (1460 bytes)]
[Frame: 54, payload: 10326-11785 (1460 bytes)]
[Frame: 55, payload: 11786-12961 (1176 bytes)]
```

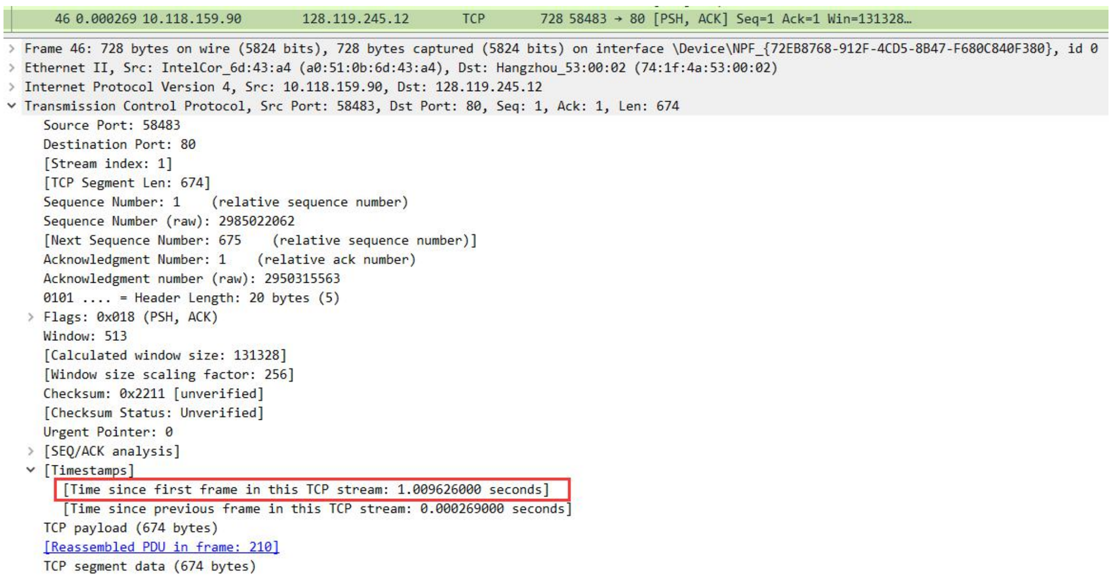
第一个 TCP segment 只携带了部分数据，并没有将 alice.txt 中的全部内容放入。

（1）sequence number : 1
（2）674bytes
（3）NO ，第一个 TCP segment 只携带了部分数据，并没有将 alice.txt 中的全部内容放入。

6. Consider the TCP segment containing the HTTP "POST"as the first segment in the data transfer part of the TCP connection.

根据第五问找到 first segment 编号是 46，second segment 编号是 47

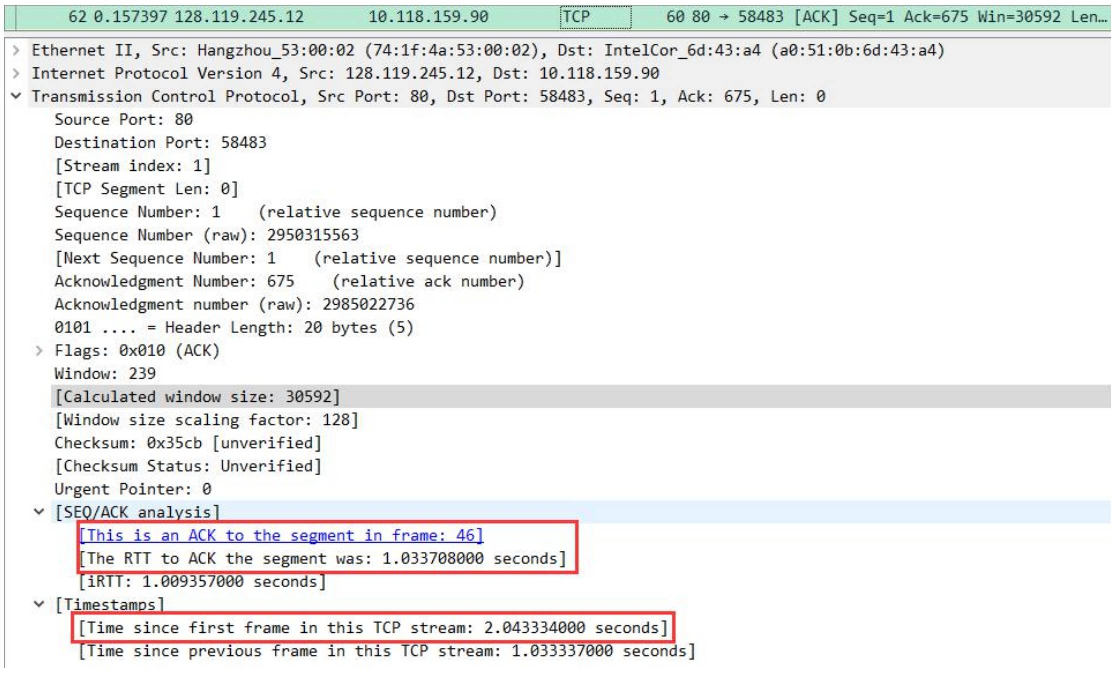
. At what time was the first segment (the one containing the HTTP POST) in the data-transfer part of the TCP connection sent?



```
46 0.000269 10.118.159.90        128.119.245.12      TCP      728 58483 → 80 [PSH, ACK] Seq=1 Ack=1 Win=131328…

> Frame 46: 728 bytes on wire (5824 bits), 728 bytes captured (5824 bits) on interface \Device\NPF_{72EB8768-912F-4CD5-8B47-F680C840F380}, id 0
> Ethernet II, Src: IntelCor_6d:43:a4 (a0:51:0b:6d:43:a4), Dst: Hangzhou_53:00:02 (74:1f:4a:53:00:02)
> Internet Protocol Version 4, Src: 10.118.159.90, Dst: 128.119.245.12
v Transmission Control Protocol, Src Port: 58483, Dst Port: 80, Seq: 1, Ack: 1, Len: 674
    Source Port: 58483
    Destination Port: 80
    [Stream index: 1]
    [TCP Segment Len: 674]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 2985022062
    [Next Sequence Number: 675    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 2950315563
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x018 (PSH, ACK)
    Window: 513
    [Calculated window size: 131328]
    [Window size scaling factor: 256]
    Checksum: 0x2211 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > [SEQ/ACK analysis]
  v [Timestamps]
      [Time since first frame in this TCP stream: 1.009626000 seconds]
      [Time since previous frame in this TCP stream: 0.000269000 seconds]
    TCP payload (674 bytes)
    [Reassembled PDU in frame: 210]
    TCP segment data (674 bytes)
```

46 号包：**Time since first frame in this TCP stream : 1.009626seconds**

. At what time was the ACK for this first data-containing segment received?. What is the RTT for this first data-containing segment?



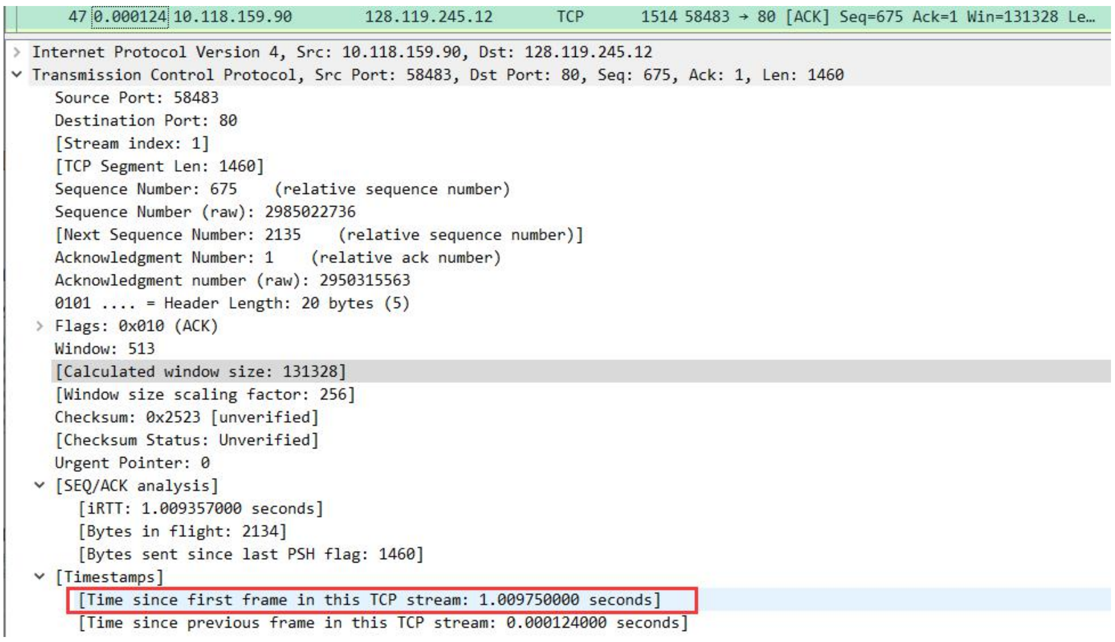
```
62 0.157397 128.119.245.12       10.118.159.90       TCP      60 80 → 58483 [ACK] Seq=1 Ack=675 Win=30592 Len…

> Ethernet II, Src: Hangzhou_53:00:02 (74:1f:4a:53:00:02), Dst: IntelCor_6d:43:a4 (a0:51:0b:6d:43:a4)
> Internet Protocol Version 4, Src: 128.119.245.12, Dst: 10.118.159.90
v Transmission Control Protocol, Src Port: 80, Dst Port: 58483, Seq: 1, Ack: 675, Len: 0
    Source Port: 80
    Destination Port: 58483
    [Stream index: 1]
    [TCP Segment Len: 0]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 2950315563
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 675    (relative ack number)
    Acknowledgment number (raw): 2985022736
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x010 (ACK)
    Window: 239
    [Calculated window size: 30592]
    [Window size scaling factor: 128]
    Checksum: 0x35cb [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  v [SEQ/ACK analysis]
      [This is an ACK to the segment in frame: 46]
      [The RTT to ACK the segment was: 1.033708000 seconds]
      [iRTT: 1.009357000 seconds]
  v [Timestamps]
      [Time since first frame in this TCP stream: 2.043334000 seconds]
      [Time since previous frame in this TCP stream: 1.033337000 seconds]
```

找到 62 号包是对 46 号包的 ACK

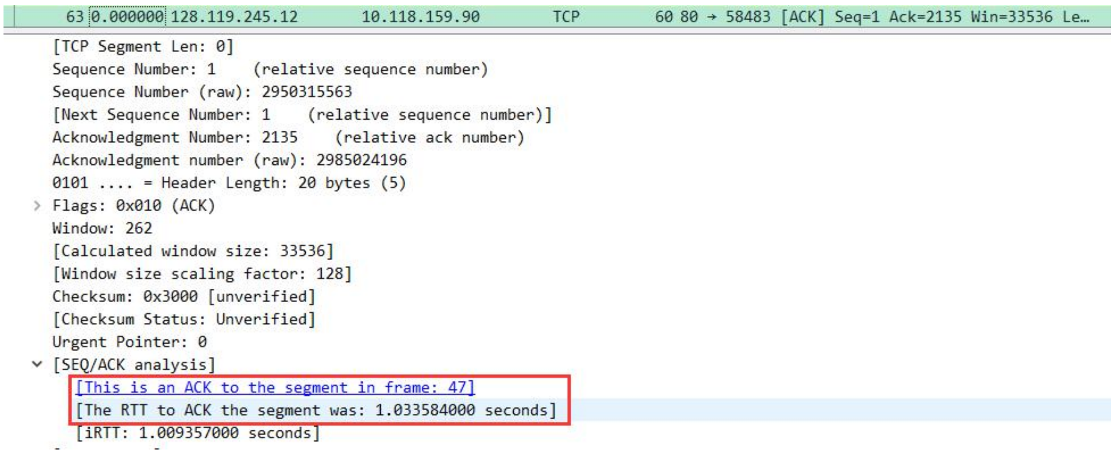
62 号包：**Time since first frame in this TCP stream : 2.043334seconds**

RTT：1.033708seconds

. What is the RTT value the second data-carrying TCP segment and its ACK?


```
47 0.000124 10.118.159.90         128.119.245.12        TCP       1514 58483 → 80 [ACK] Seq=675 Ack=1 Win=131328 Le…
> Internet Protocol Version 4, Src: 10.118.159.90, Dst: 128.119.245.12
v Transmission Control Protocol, Src Port: 58483, Dst Port: 80, Seq: 675, Ack: 1, Len: 1460
    Source Port: 58483
    Destination Port: 80
    [Stream index: 1]
    [TCP Segment Len: 1460]
    Sequence Number: 675    (relative sequence number)
    Sequence Number (raw): 2985022736
    [Next Sequence Number: 2135    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 2950315563
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x010 (ACK)
    Window: 513
    [Calculated window size: 131328]
    [Window size scaling factor: 256]
    Checksum: 0x2523 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  v [SEQ/ACK analysis]
      [iRTT: 1.009357000 seconds]
      [Bytes in flight: 2134]
      [Bytes sent since last PSH flag: 1460]
  v [Timestamps]
      [Time since first frame in this TCP stream: 1.009750000 seconds]
      [Time since previous frame in this TCP stream: 0.000124000 seconds]
```

47 号包是 the second data-carrying TCP segment


```
63 0.000000 128.119.245.12        10.118.159.90         TCP         60 80 → 58483 [ACK] Seq=1 Ack=2135 Win=33536 Le…
    [TCP Segment Len: 0]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 2950315563
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 2135    (relative ack number)
    Acknowledgment number (raw): 2985024196
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x010 (ACK)
    Window: 262
    [Calculated window size: 33536]
    [Window size scaling factor: 128]
    Checksum: 0x3000 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  v [SEQ/ACK analysis]
      [This is an ACK to the segment in frame: 47]
      [The RTT to ACK the segment was: 1.033584000 seconds]
      [iRTT: 1.009357000 seconds]
```

63 号包是对 47 号包的 ACK，**RTT 是 1.033584seconds。**

. What is the Estimated RTT value(see Section 3.5.3, in the text) after the ACK for the second data-carrying segment is received? Assume that in making this calculation after the received of the ACK for the second segment,that the initial value of Estimated RTT is equal to the measured RTT for the first segment, and then is computed using the EstimatedRTT equation onpage 242, and a value of a = 0.125.

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.

EstimatedRTT = 0.875 * EstimatedRTT + 0.125 * SampleRTT

After the first segment received:

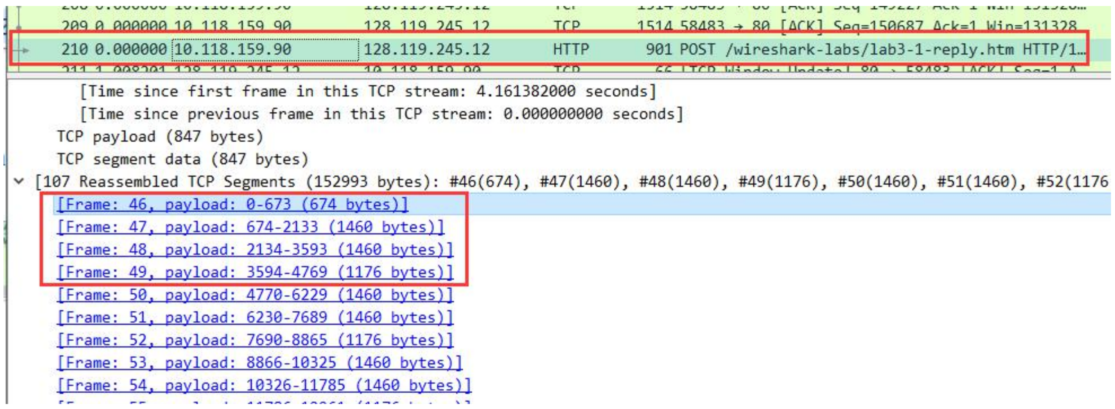
EstimatedRTT = 1.033708s

After the second segment received:

EstimatedRTT = 0.875 * 1.033708 + 0.125 * 1.033584

= 1.0336925s

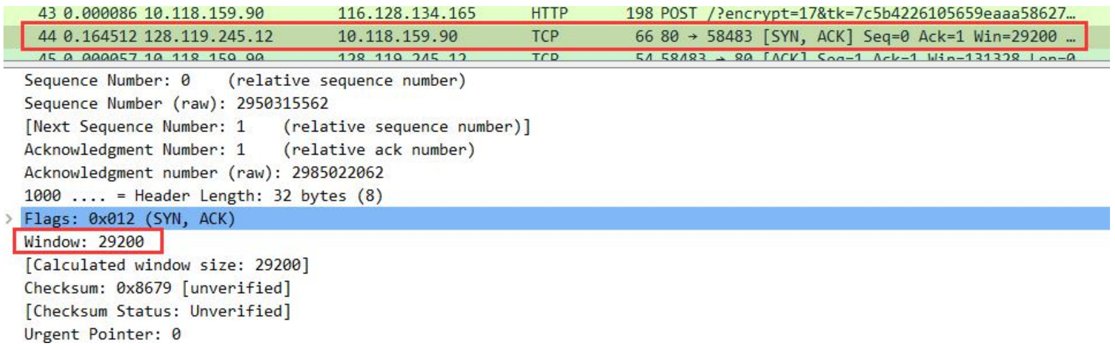
7. What is the length (header plus payload) of each of the first four data-carrying TCP segments?



```
209 0.000000 10.118.159.90    128.119.245.12    TCP     1514 58483 → 80 [ACK] Seq=150687 Ack=1 Win=131328
210 0.000000 10.118.159.90    128.119.245.12    HTTP     901 POST /wireshark-labs/lab3-1-reply.htm HTTP/1…

      [Time since first frame in this TCP stream: 4.161382000 seconds]
      [Time since previous frame in this TCP stream: 0.000000000 seconds]
    TCP payload (847 bytes)
    TCP segment data (847 bytes)
  [107 Reassembled TCP Segments (152993 bytes): #46(674), #47(1460), #48(1460), #49(1176), #50(1460), #51(1460), #52(1176
    [Frame: 46, payload: 0-673 (674 bytes)]
    [Frame: 47, payload: 674-2133 (1460 bytes)]
    [Frame: 48, payload: 2134-3593 (1460 bytes)]
    [Frame: 49, payload: 3594-4769 (1176 bytes)]
    [Frame: 50, payload: 4770-6229 (1460 bytes)]
    [Frame: 51, payload: 6230-7689 (1460 bytes)]
    [Frame: 52, payload: 7690-8865 (1176 bytes)]
    [Frame: 53, payload: 8866-10325 (1460 bytes)]
    [Frame: 54, payload: 10326-11785 (1460 bytes)]
```

前四个段长度分别为:**674bytes, 1460bytes, 1460bytes, 1176bytes**

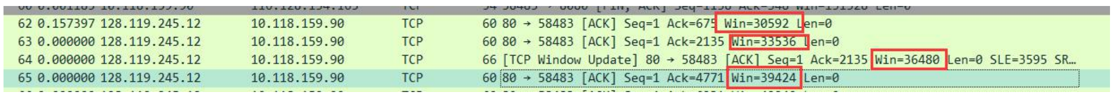
8. What is the minimum amount of available buffer space advertised to the client by gaia.cs.umass.edu among these first four data-carrying TCP segments? Does the lack of receiver buffer space ever throttle the sender for these first four data-carrying segments?



```
43 0.000086 10.118.159.90    116.128.134.165    HTTP     198 POST /?encrypt=17&tk=7c5b4226105659eaaa58627…
44 0.164512 128.119.245.12   10.118.159.90      TCP       66 80 → 58483 [SYN, ACK] Seq=0 Ack=1 Win=29200 …

  Sequence Number: 0    (relative sequence number)
  Sequence Number (raw): 2950315562
  [Next Sequence Number: 1    (relative sequence number)]
  Acknowledgment Number: 1    (relative ack number)
  Acknowledgment number (raw): 2985022062
  1000 .... = Header Length: 32 bytes (8)
> Flags: 0x012 (SYN, ACK)
  Window: 29200
  [Calculated window size: 29200]
  Checksum: 0x8679 [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
```

（1）在 44 号分组传输完成后，46-47 便是前四个 tcp segment

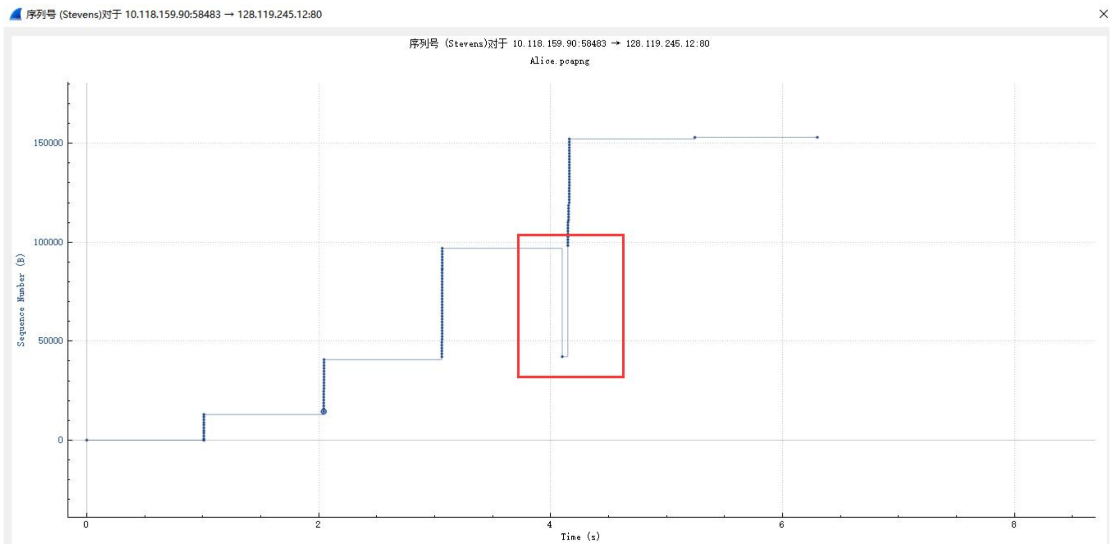
在传送前四个 TCP segment 之前服务器传回的的**窗口大小为 29200**



```
62 0.157397 128.119.245.12   10.118.159.90   TCP    60 80 → 58483 [ACK] Seq=1 Ack=675 Win=30592 Len=0
63 0.000000 128.119.245.12   10.118.159.90   TCP    60 80 → 58483 [ACK] Seq=1 Ack=2135 Win=33536 Len=0
64 0.000000 128.119.245.12   10.118.159.90   TCP    66 [TCP Window Update] 80 → 58483 [ACK] Seq=1 Ack=2135 Win=36480 Len=0 SLE=3595 SR…
65 0.000000 128.119.245.12   10.118.159.90   TCP    60 80 → 58483 [ACK] Seq=1 Ack=4771 Win=39424 Len=0
```

（2）62 是对第一个 tcp segment 的 ACK，65 是对第四个 tcp segment 的 ACK，可以看到在这期间窗口值稳步增长，说明**没有出现接收方窗口大小抑制发放传输速率**

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



存在重传的分组，发送到服务器的 **sequence number** 并不是一直递增的，比如在图中 **t=4s** 左右发送的分组序号小于上一个的序号。

10.How much data does the receiver typically acknowledge in an ACK among the first ten data-carrying segments sent from the client to gaia.cs.umass.edu?Can you identify cases where the receiver is ACKing every other received segment(see Table 3.2 in the text) among these first ten data-carrying segments?

```
[Frame: 46, payload: 0-673 (674 bytes)]
[Frame: 47, payload: 674-2133 (1460 bytes)]
[Frame: 48, payload: 2134-3593 (1460 bytes)]
[Frame: 49, payload: 3594-4769 (1176 bytes)]
[Frame: 50, payload: 4770-6229 (1460 bytes)]
[Frame: 51, payload: 6230-7689 (1460 bytes)]
[Frame: 52, payload: 7690-8865 (1176 bytes)]
[Frame: 53, payload: 8866-10325 (1460 bytes)]
[Frame: 54, payload: 10326-11785 (1460 bytes)]
[Frame: 55, payload: 11786-12961 (1176 bytes)]
```

（1）在前十个 TCP segment 中接收方在一个 ACK 确定的最多的是 1460bytes (6 次)，其次是 1176 字节 （3 次），674bytes （1 次）。

观察前十个 tcp segment 的 ACK 的 sequence number 和 payload 之间的关系，得出下表

ACK 的 Frame	ACK 的 sequence number	receiver acking data
46	1	674
47	675	1460
48	2135	1460
49	3595	1176
50	4771	1460
51	6231	1460
52	7691	1176

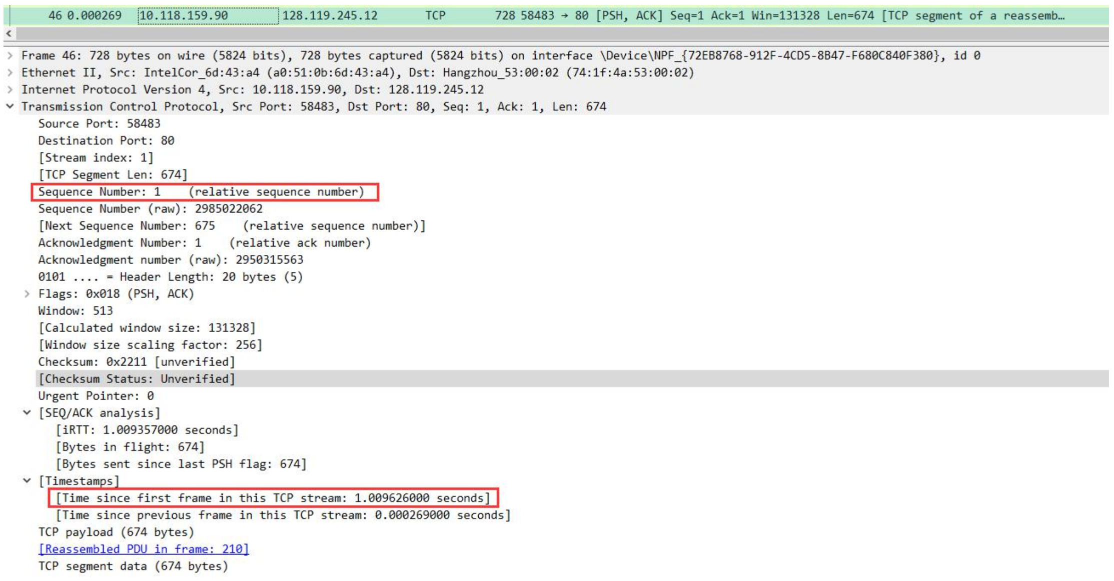
53	8867	1460
54	10327	1460
55	11787	1176

（2）观察上表可以看出：每次服务器 ACK 的 tcp segment 的 sequence num 增长的间隔就是前一个 tcp segment 的 payload

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

throughput = ammount of data transmitted / time

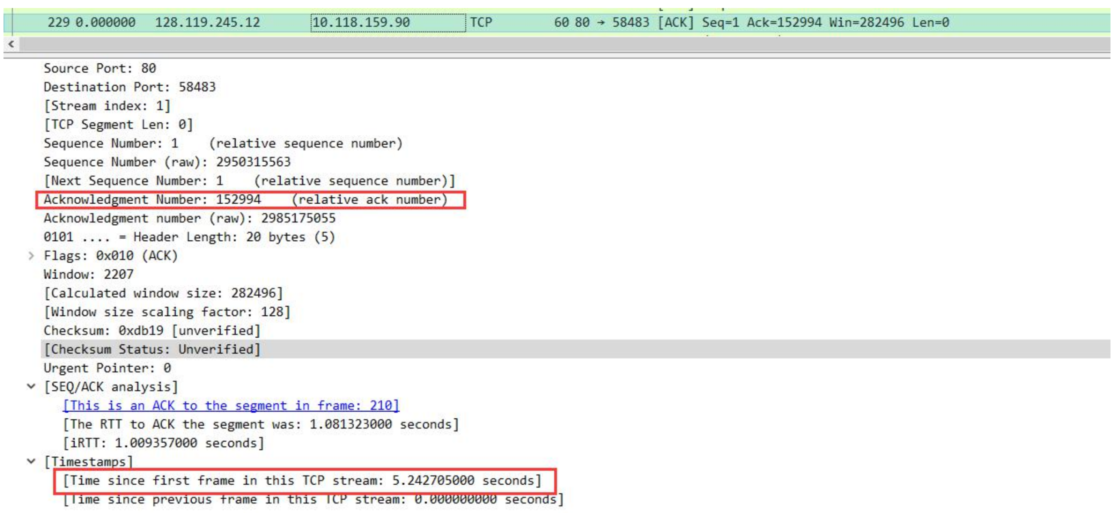
总传输数据量：首先寻找第一个 tcp segment



```
   46 0.000269  10.118.159.90     128.119.245.12      TCP       728 58483 → 80 [PSH, ACK] Seq=1 Ack=1 Win=131328 Len=674 [TCP segment of a reassemb…
<
> Frame 46: 728 bytes on wire (5824 bits), 728 bytes captured (5824 bits) on interface \Device\NPF_{72EB8768-912F-4CD5-8B47-F680C840F380}, id 0
> Ethernet II, Src: IntelCor_6d:43:a4 (a0:51:0b:6d:43:a4), Dst: Hangzhou_53:00:02 (74:1f:4a:53:00:02)
> Internet Protocol Version 4, Src: 10.118.159.90, Dst: 128.119.245.12
v Transmission Control Protocol, Src Port: 58483, Dst Port: 80, Seq: 1, Ack: 1, Len: 674
    Source Port: 58483
    Destination Port: 80
    [Stream index: 1]
    [TCP Segment Len: 674]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 2985022062
    [Next Sequence Number: 675    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 2950315563
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x018 (PSH, ACK)
    Window: 513
    [Calculated window size: 131328]
    [Window size scaling factor: 256]
    Checksum: 0x2211 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  v [SEQ/ACK analysis]
      [iRTT: 1.009357000 seconds]
      [Bytes in flight: 674]
      [Bytes sent since last PSH flag: 674]
  v [Timestamps]
      [Time since first frame in this TCP stream: 1.009626000 seconds]
      [Time since previous frame in this TCP stream: 0.000269000 seconds]
    TCP payload (674 bytes)
    [Reassembled PDU in frame: 210]
    TCP segment data (674 bytes)
```

seq 为 1，time 为 1.009626s

然后寻找最后一个 ACK



```
   229 0.000000   128.119.245.12    10.118.159.90      TCP        60 80 → 58483 [ACK] Seq=1 Ack=152994 Win=282496 Len=0
<
     Source Port: 80
     Destination Port: 58483
     [Stream index: 1]
     [TCP Segment Len: 0]
     Sequence Number: 1    (relative sequence number)
     Sequence Number (raw): 2950315563
     [Next Sequence Number: 1    (relative sequence number)]
     Acknowledgment Number: 152994    (relative ack number)
     Acknowledgment number (raw): 2985175055
     0101 .... = Header Length: 20 bytes (5)
   > Flags: 0x010 (ACK)
     Window: 2207
     [Calculated window size: 282496]
     [Window size scaling factor: 128]
     Checksum: 0xdb19 [unverified]
     [Checksum Status: Unverified]
     Urgent Pointer: 0
   v [SEQ/ACK analysis]
       [This is an ACK to the segment in frame: 210]
       [The RTT to ACK the segment was: 1.081323000 seconds]
       [iRTT: 1.009357000 seconds]
   v [Timestamps]
       [Time since first frame in this TCP stream: 5.242705000 seconds]
       [Time since previous frame in this TCP stream: 0.000000000 seconds]
```

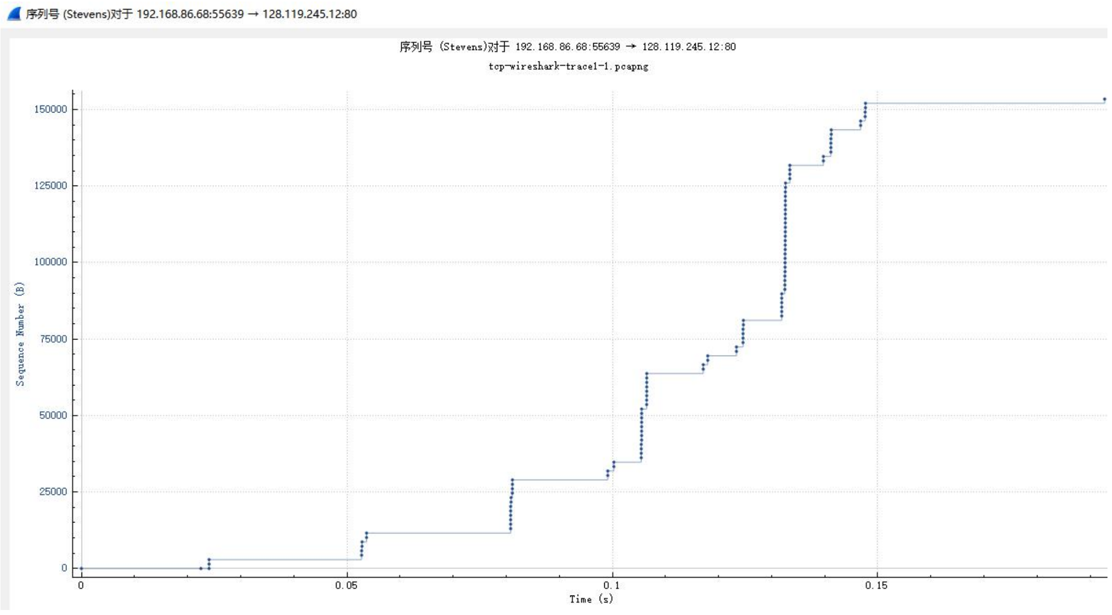
ACK 的 tcp segment 的 sequence number 是 152994，time 是 5.242705

故 throughput=(152994-1) / (5.242705-1.009626)=36142.25bytes/s

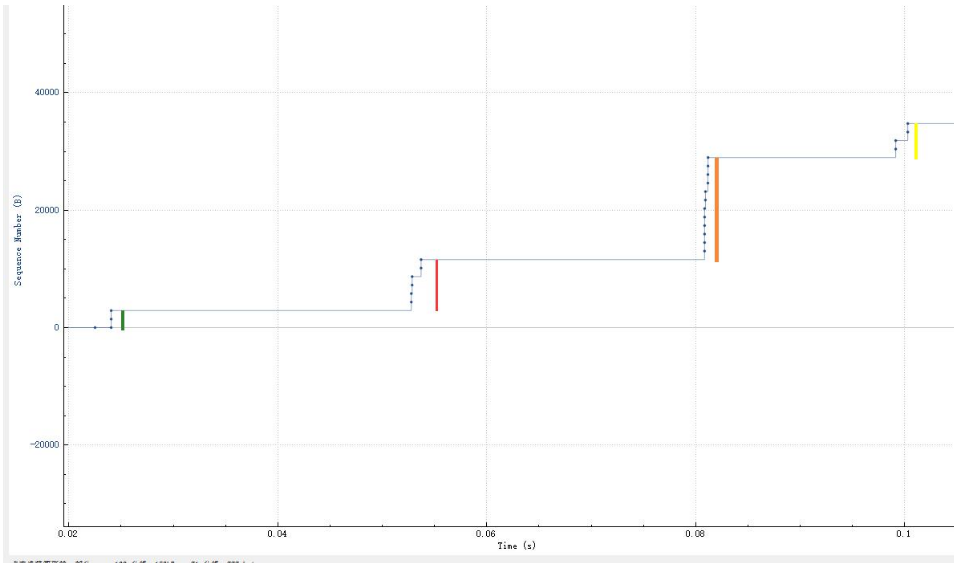
4.TCP congestion control in action

12.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. Consider the "fleets”of packets sent around t= 0.025,t=0.053,t= 0.082 and t=0.1.Comment on whether this looks as if TCP is in its slow start phase, congestion avoidance phase or some other phase.Figure 6 showsa slightly different view of this data.

首先访问
http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces-8.1.zip
下载 tcp-wireshark-trace1-1
打开统计->TCP 流图形->时间序列，得到下图



分析其中部分：



根据图中标出的线段：绿线，红线，橙线长度可知，在 t=0.025，0.053,0.082 时刻附近，sequence number 成倍增长（红线长度差=2 倍绿线长度，橙线长度=2 倍红线长度）可知，**t=0.025，0.053,0.082 时刻左右均处于慢开始阶段。**而 t=1s 时刻左右标出的黄线长度与开始时相近，说明在 t=0.082-到 t=0.1 之间发生了超时，**所以在 t=1 时刻又回到了慢开始阶段。**

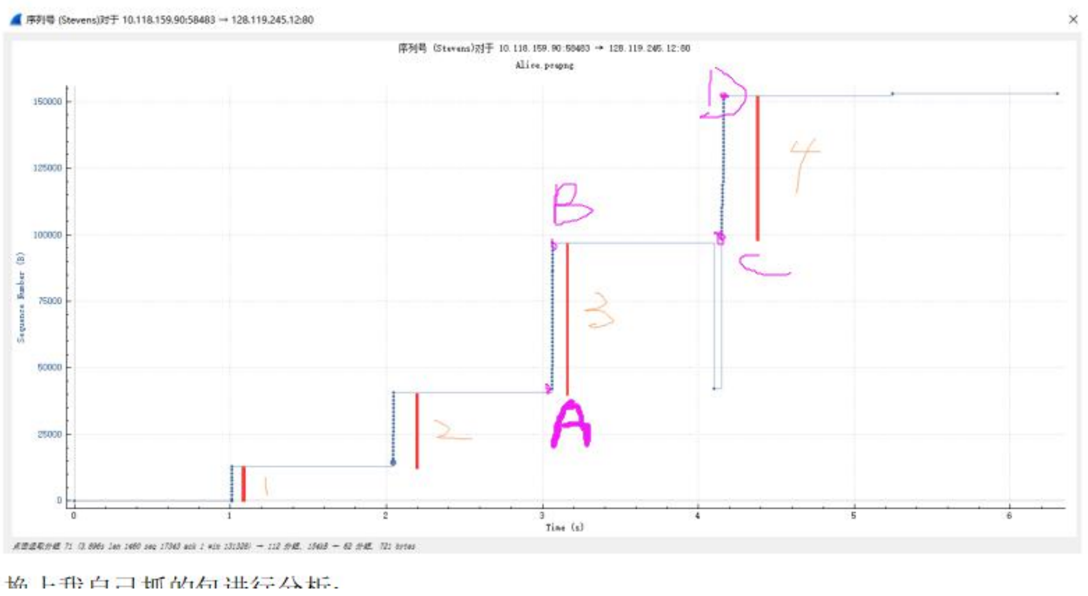
13.These "fleets of segments appear to have some periodicity. What can you say about the period?

总体来说：
tcp 在开始发送报文段时先发送一个，试探网络的拥塞程度，然后成倍增加；
当发送数量超过了慢开始门限之后，就要执行拥塞避免，发送报文段数线性增加；
一旦发生超时，就要重新执行慢开始阶段；
连续收到 3 个重复确认，则乘法减小；

在发送数小于拥塞窗口时有特定发送报文段数的的加倍增大和加法增大，发送报文段数逐渐增多；而在超过拥塞窗口后又要退回去重新执行发送报文段数的增大；故呈现出一定的周期性。

在这张图中：
周期性主要表现在 TCP 报文段发送数小于慢开始门限时的倍增和发生超时后的退回到慢开始所表现出的周期性规律中。

14. Answer each of two questions above for the trace that you have gatheredwhen you transferred a file from your computer to gaia.cs.umass.edu



换上我自己抓的包进行分析：
（1）
1、2、3 表示的线段长度逐渐倍增，可以判断在 t=1s,t=2.05s,t=3.08s 左右的时刻，均处于慢开始的阶段。计算 A、B 两点直接的 sequence number 之差为 96911-49463=47488，C、D 两

点的 sequence number 之差为 152147-99831=52316 可以看出 t=4.1s 时刻左右，TCP 发送报文段数执行加法增长，故处于拥塞避免阶段。

（2）

对周期性的分析同上题，不过我自己抓的包似乎没有周期性，只有从慢开始的倍增向拥塞避免的加法增大的变化阶段。