一．实验要求和操作：

在 http://gaia.cs.umass.edu/wireshark-labs/alice.txt 下载文本内容并保存到本地

在 http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html 选择该文件并上传

上传前后用 wireshark 捕获并查看批量 TCP 传输

（注：Q1、Q2 及 Q7 中与 RTT 有关的部分、Q10、Q12、Q13 是用 zip 文件中的 TCP 捕获结果分析的，其它是用自己的捕获结果分析的）

二．回答问题：

1. 将文件传输到 gaia.cs.umass.edu 的客户计算机（源）使用的 IP 地址和 TCP 端口号码是什么？

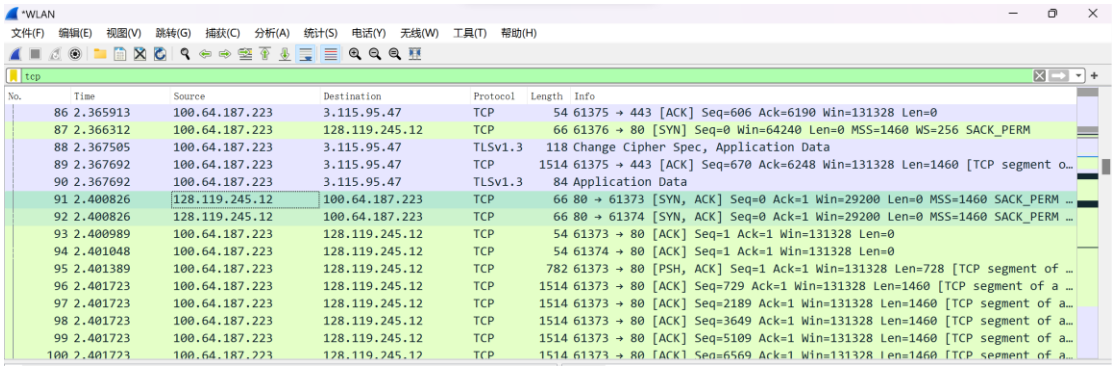
```
   1 0.000000      192.168.1.102         128.119.245.12        TCP        62 1161 → 80 [SYN] Seq=0 Win=163
   2 0.023172      128.119.245.12        192.168.1.102         TCP        62 80 → 1161 [SYN, ACK] Seq=0 Ac
   3 0.023265      192.168.1.102         128.119.245.12        TCP        54 1161 → 80 [ACK] Seq=1 Ack=1 W
   4 0.026477      192.168.1.102         128.119.245.12        TCP       619 1161 → 80 [PSH, ACK] Seq=1 Ac
```

主机 IP 地址为：192.168.1.102　　TCP 端口号为：1161

2. gaia.cs.umass.edu 的 IP 地址是什么？在此连接中，它会发送和接收哪个端口号码的 TCP 段？

IP 地址为：　128.119.245.12　发送和接收端口号为 80 的 TCP 段

3. 客户端计算机（源）将文件传输到 gaia.cs.umass.edu 所使用的 IP 地址和 TCP 端口号是多少？



主机 IP 地址为：100.64.187.223　　TCP 端口号为：61373

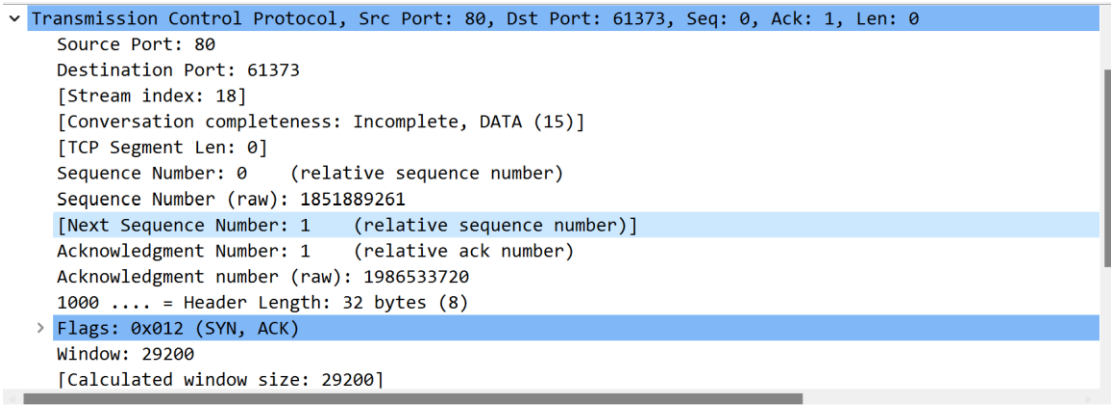
4. 用于启动客户端计算机和 gaia.cs.umass.edu 之间的 TCP 连接的 TCP SYN 段的序列号是多少？将区段标识为 SYN 区段的区段有什么功能？

```
87 2.366312      100.64.187.223        128.119.245.12        TCP        66 61376 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
v Transmission Control Protocol, Src Port: 61376, Dst Port: 80, Seq: 0, Len: 0
    Source Port: 61376
    Destination Port: 80
    [Stream index: 21]
    [Conversation completeness: Incomplete, ESTABLISHED (7)]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 2493924383
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 0
    Acknowledgment number (raw): 0
    1000 .... = Header Length: 32 bytes (8)
```

SYN 相对序列号为 0，代表客户端请求建立连接。

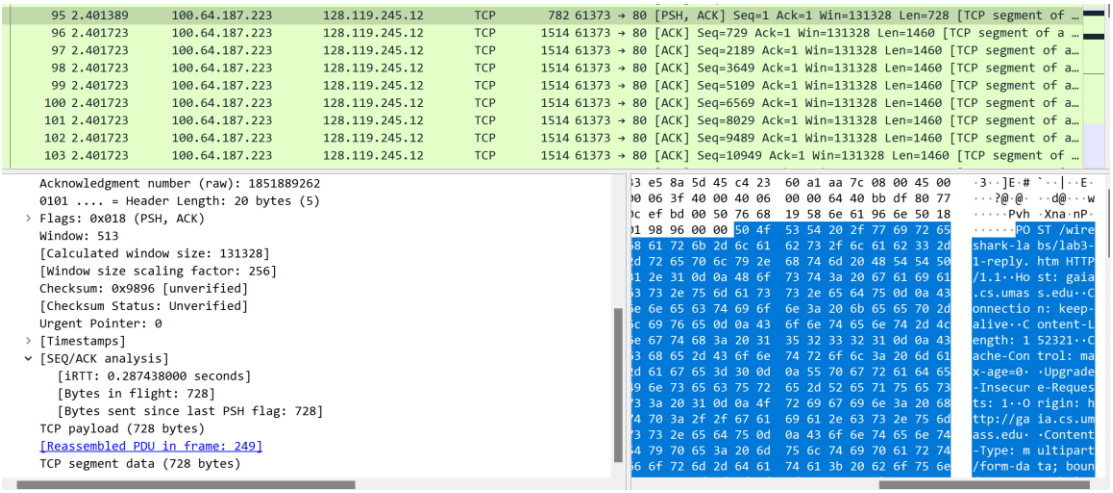
5. gaia.cs.umass.edu 发送到客户端计算机以响应 SYN 的 SYNACK 段的序列号是多少？

SYNACK 中确认字段的值是多少？ gaia.cs.umass.edu 如何确定该值？这一部分的作用有什么？



SYN 序列号为 0，ACK 为 1，ACK 是按照 SYN+1 得到（即 0+1=1）作用是表示服务器接收到我的连接请求并且发 SYN-ACK 确认。

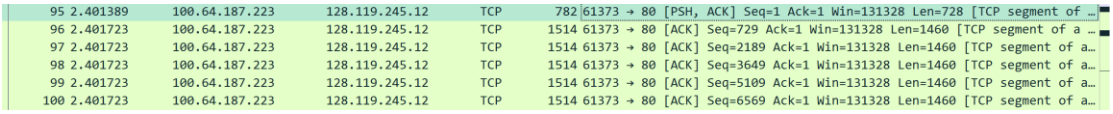
6. 包含 HTTP POST 命令的 TCP 段的序列号是多少？

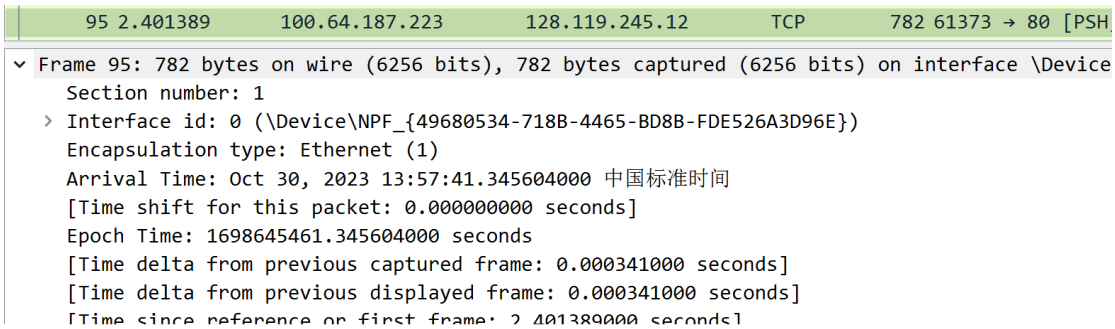


序列号为 1

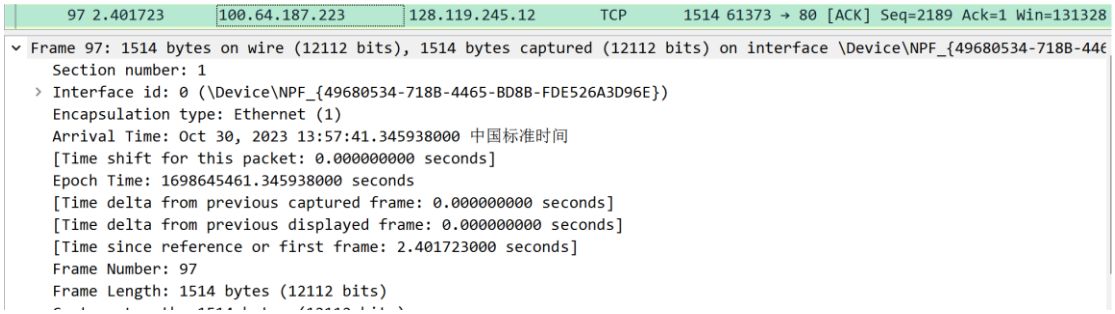
7. 将包含 HTTP POST 的 TCP 区段视为 TCP 连接中的第一个区段。在这个 TCP 连接中前六个 TCP 区段的序列号是什么（包括包含 HTTP POST 的段）？ 每区段发送的时间是什么时候？ 收到的每个区段的 ACK 是什么时候？ 鉴于发送每个 TCP 区段的时间与收到确认的时间之间的差异，六个区段中每个区段的 RTT 值是多少？ 收到每个 ACK 后，EstimatedRTT 值是什么？ 假设第一个 EstimatedRTT 的值等于第一个区段的测量 RTT，然后使用课本第 242 页的 EstimatedRTT 公式计算所有后续区段。

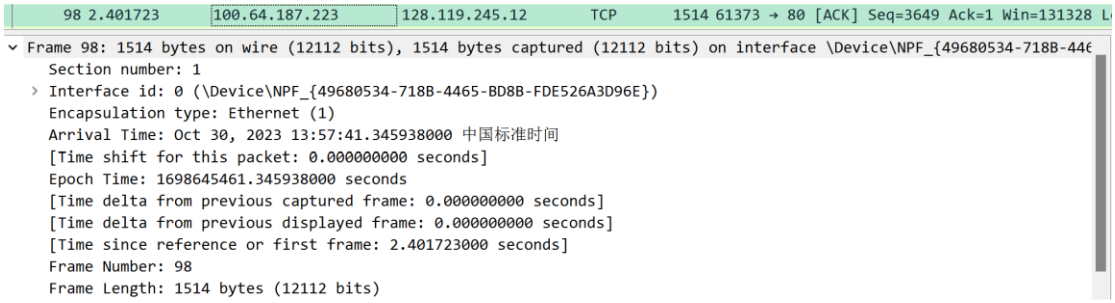
以下是前六个 TCP 区段及其对应发送时间的信息：




95 2.401389 100.64.187.223 128.119.245.12 TCP 782 61373 → 80 [PSH
Frame 95: 782 bytes on wire (6256 bits), 782 bytes captured (6256 bits) on interface \Device
Section number: 1
Interface id: 0 (\Device\NPF_{49680534-718B-4465-BD8B-FDE526A3D96E})
Encapsulation type: Ethernet (1)
Arrival Time: Oct 30, 2023 13:57:41.345604000 中国标准时间
[Time shift for this packet: 0.000000000 seconds]
Epoch Time: 1698645461.345604000 seconds
[Time delta from previous captured frame: 0.000341000 seconds]
[Time delta from previous displayed frame: 0.000341000 seconds]

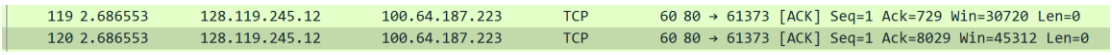

96 2.401723 100.64.187.223 128.119.245.12 TCP 1514 61373 → 80 [ACK] Seq=729 Ack=1 Win=13
Frame 96: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface \Device\NPF_{49680534-718
Section number: 1
Interface id: 0 (\Device\NPF_{49680534-718B-4465-BD8B-FDE526A3D96E})
Encapsulation type: Ethernet (1)
Arrival Time: Oct 30, 2023 13:57:41.345938000 中国标准时间
[Time shift for this packet: 0.000000000 seconds]
Epoch Time: 1698645461.345938000 seconds
[Time delta from previous captured frame: 0.000334000 seconds]
[Time delta from previous displayed frame: 0.000334000 seconds]
[Time since reference or first frame: 2.401723000 seconds]
Frame Number: 96

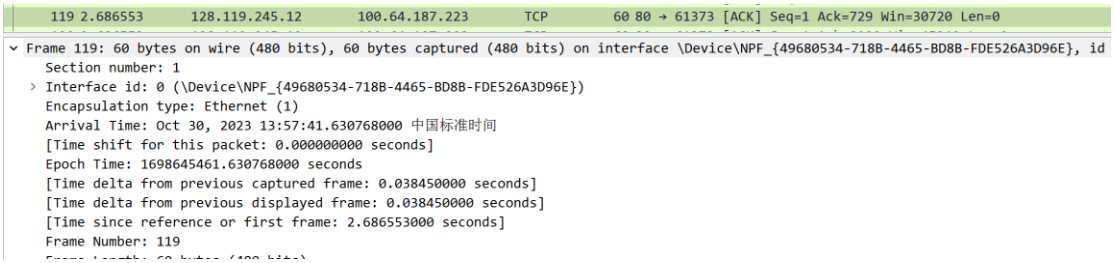

97 2.401723 100.64.187.223 128.119.245.12 TCP 1514 61373 → 80 [ACK] Seq=2189 Ack=1 Win=131328
Frame 97: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface \Device\NPF_{49680534-718B-446
Section number: 1
Interface id: 0 (\Device\NPF_{49680534-718B-4465-BD8B-FDE526A3D96E})
Encapsulation type: Ethernet (1)
Arrival Time: Oct 30, 2023 13:57:41.345938000 中国标准时间
[Time shift for this packet: 0.000000000 seconds]
Epoch Time: 1698645461.345938000 seconds
[Time delta from previous captured frame: 0.000000000 seconds]
[Time delta from previous displayed frame: 0.000000000 seconds]
[Time since reference or first frame: 2.401723000 seconds]
Frame Number: 97
Frame Length: 1514 bytes (12112 bits)

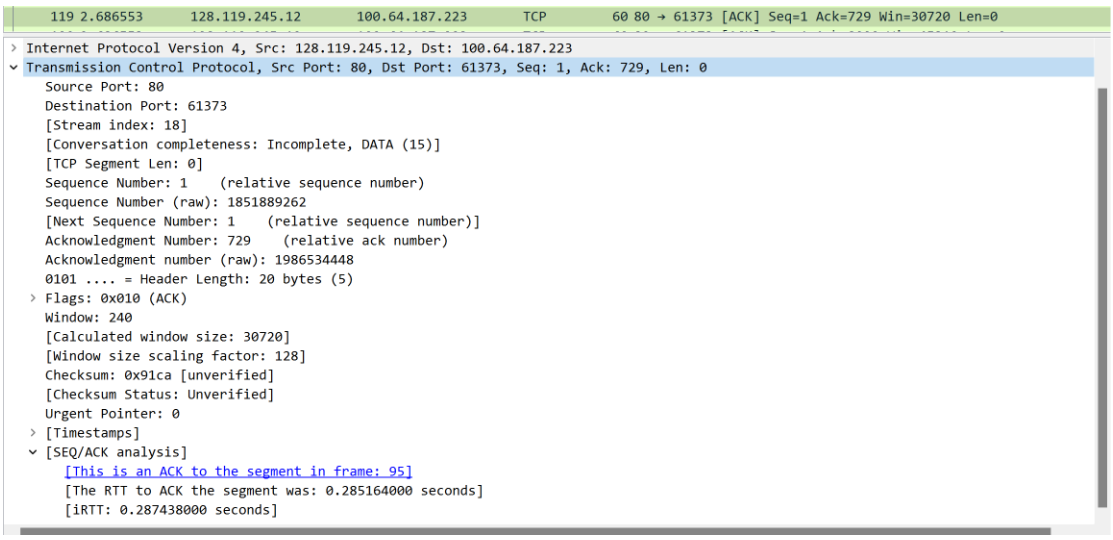

98 2.401723 100.64.187.223 128.119.245.12 TCP 1514 61373 → 80 [ACK] Seq=3649 Ack=1 Win=131328 L
Frame 98: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface \Device\NPF_{49680534-718B-446
Section number: 1
Interface id: 0 (\Device\NPF_{49680534-718B-4465-BD8B-FDE526A3D96E})
Encapsulation type: Ethernet (1)
Arrival Time: Oct 30, 2023 13:57:41.345938000 中国标准时间
[Time shift for this packet: 0.000000000 seconds]
Epoch Time: 1698645461.345938000 seconds
[Time delta from previous captured frame: 0.000000000 seconds]
[Time delta from previous displayed frame: 0.000000000 seconds]
[Time since reference or first frame: 2.401723000 seconds]
Frame Number: 98
Frame Length: 1514 bytes (12112 bits)

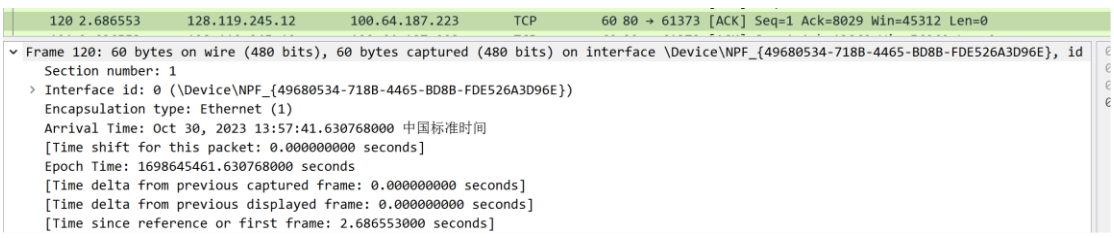

99 2.401723 100.64.187.223 128.119.245.12 TCP 1514 61373 → 80 [ACK] Seq=5109 Ack=1 Win=13
Frame 99: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface \Device\NPF_{49680534-718E
Section number: 1
Interface id: 0 (\Device\NPF_{49680534-718B-4465-BD8B-FDE526A3D96E})
Encapsulation type: Ethernet (1)
Arrival Time: Oct 30, 2023 13:57:41.345938000 中国标准时间
[Time shift for this packet: 0.000000000 seconds]
Epoch Time: 1698645461.345938000 seconds
[Time delta from previous captured frame: 0.000000000 seconds]
[Time delta from previous displayed frame: 0.000000000 seconds]
[Time since reference or first frame: 2.401723000 seconds]
Frame Number: 99
Frame Length: 1514 bytes (12112 bits)
Capture Length: 1514 bytes (12112 bits)
[Frame is marked: False]

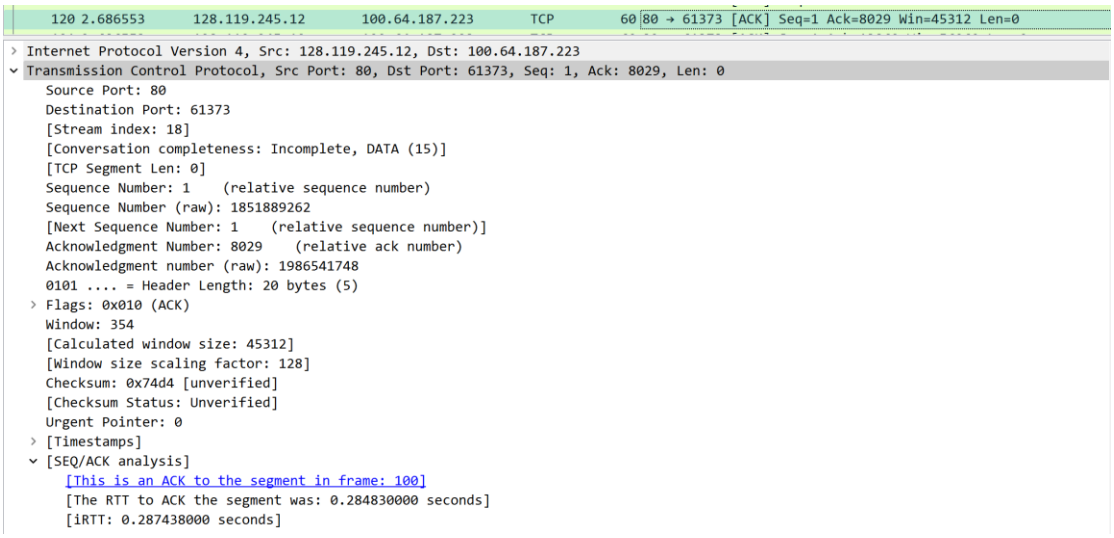
以下为收到 ACK 的接收时间和 RTT 信息：(其中只收到了第一个 TCP 区段和第六个 TCP 区段的 ACK)







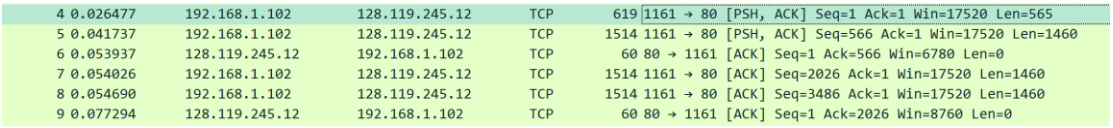


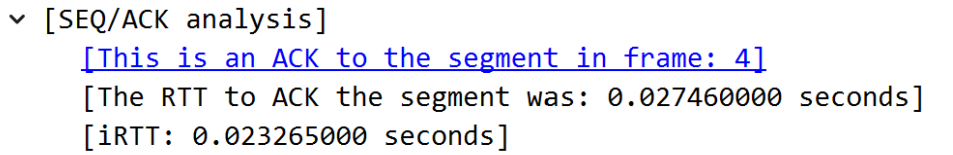


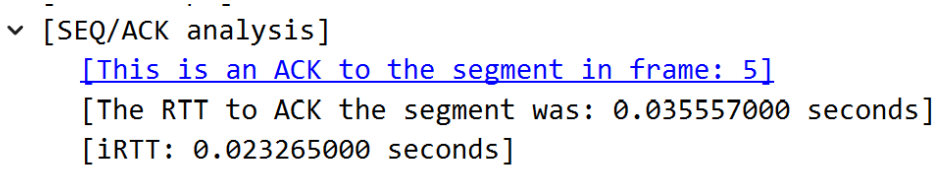
这是统计出的不同区段的序列号、发送时间和 ack 到达时间。（由于 ack 的数据缺失，在统计 RTT 和 EstimatedRTT 时，我是用的作者给的包中的信息来做的，RTT 的内容写在这个表格后面）

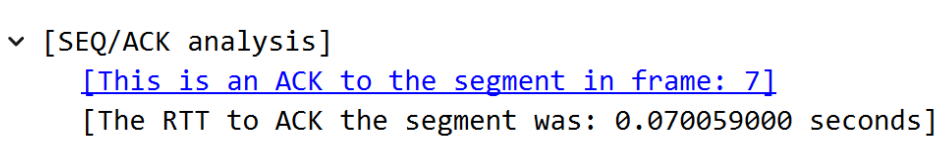
	序列号	发送时间	到达时间
区段1	1	Oct 30,2023 13:57:41.34560400	Oct 30,2023 13:57:41.63076800
区段2	729	Oct 30,2023 13:57:41.34593800	
区段3	2189	Oct 30,2023 13:57:41.34593800	
区段4	3649	Oct 30,2023 13:57:41.34593800	
区段5	5109	Oct 30,2023 13:57:41.34593800	
区段6	6569	Oct 30,2023 13:57:41.34593800	Oct 30,2023 13:57:41.63076800

以 zip 文件中提供的捕获结果：









```
▾ [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]

▾ [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]

▾ [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]
```

RTT 分别为 0.027460 、 0.035557 、 0.070059 、 0.114428 、 0.139894 、 0.189645

根据课本上的公式 EstimatedRTT = (1 - a) × EstimatedRTT + a × SampleRTT

a 取为推荐系数 0.125，EstimatedRTT 分别为：

0.027460

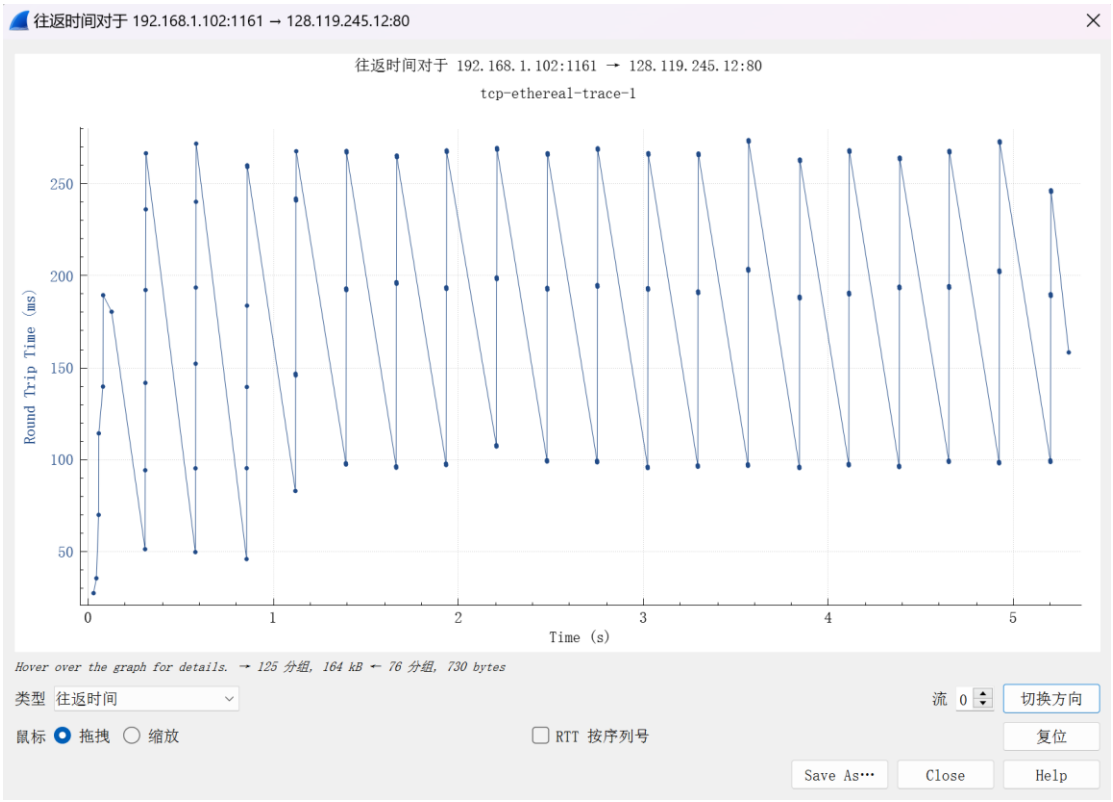
0.028472125=0.875*0.027460+0.125*0.035557

0.033670484=0.875*0.028472125+0.125*0.070059

0.043765174=0.875*0.033670484+0.125*0.114428

0.055781277=0.875*0.043765174+0.125*0.139894

0.072514242=0.875*0.055781277+0.125*0.189645



8. 前 6 个 TCP 报文段的长度为

第一个：729-1=728

第二个：2189-729=1460

第三个：3649-2189=1460

第四个：5109-3649=1460

第五个：6569-5109=1460

第六个：8029-6569=1460

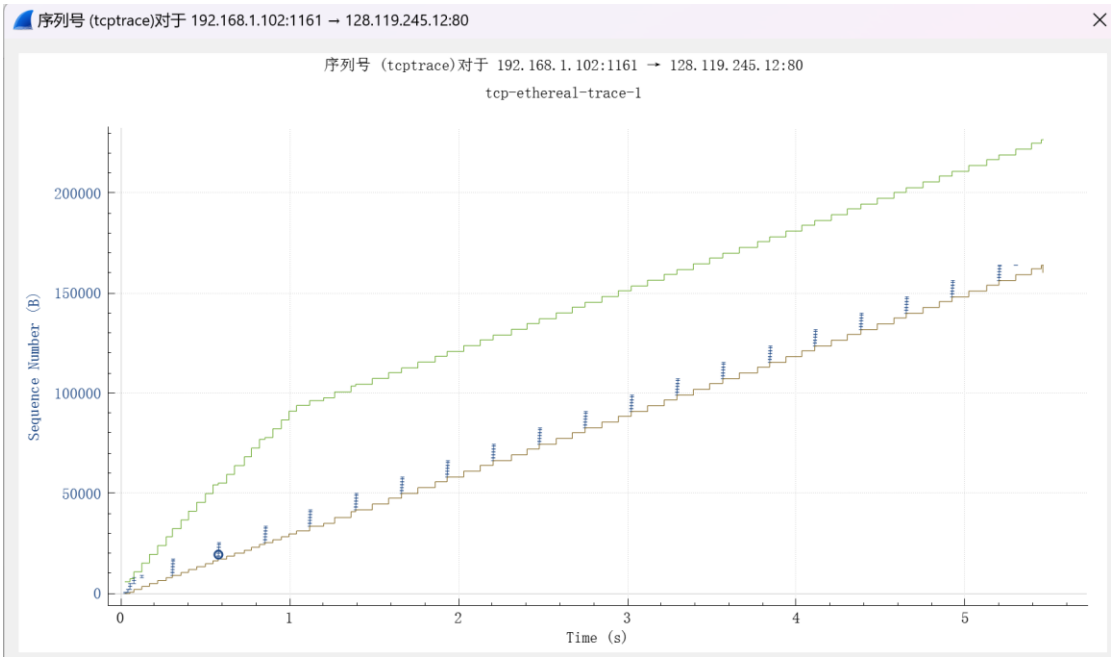
9. 对于整个跟踪包，收到的最小可用缓冲区空间量是多少？缺少接收器缓冲区空间是否会限制发送方传送 TCP 区段？

对于服务器而言，收到的最小可用缓冲区空间量为 30720

```
60 80 → 61373 [ACK] Seq=1 Ack=729 Win=30720 Len=0
60 80 → 61373 [ACK] Seq=1 Ack=8029 Win=45312 Len=0
```

缺少接收器缓冲区空间会限制发送方传送 TCP 区段

10. 在跟踪文件中是否有任何重传的报文段？为了回答这个问题，你（在跟踪中）检查了什么？



检查数据包的时间序列，因为序列号呈增大趋势，因此判定为没有重传

11. 接收方通常在一次 ACK 中确认多少数据？ 你能找出接收方每收到一个报文段就确认一次的情况吗？

接收方一次通常确认 1460bits 的数据。TCP 是累计确认的，一般接收方每隔一个接收到的区段才发送确认，故没有找到符合要求的例子。

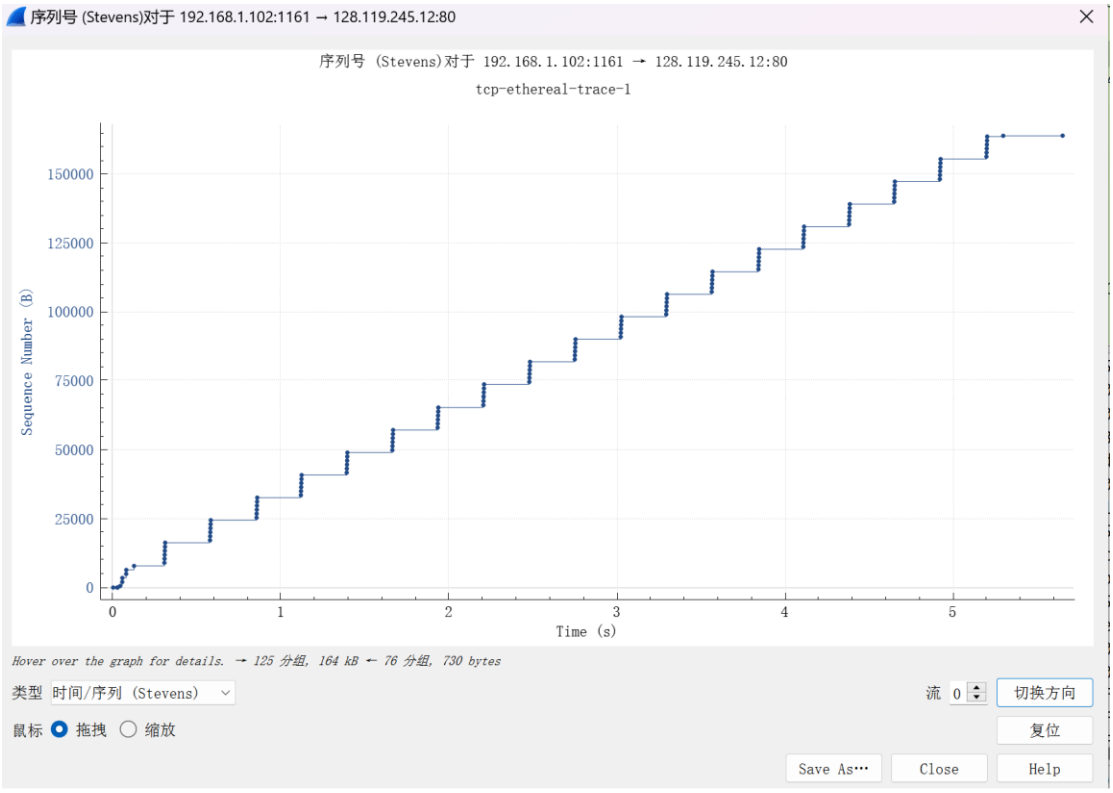
12. TCP 连接的吞吐量（每单位时间传输的字节数）是多少？ 解释一下是如何计算这个数值的。**（对于 zip 文件中的捕获结果）**

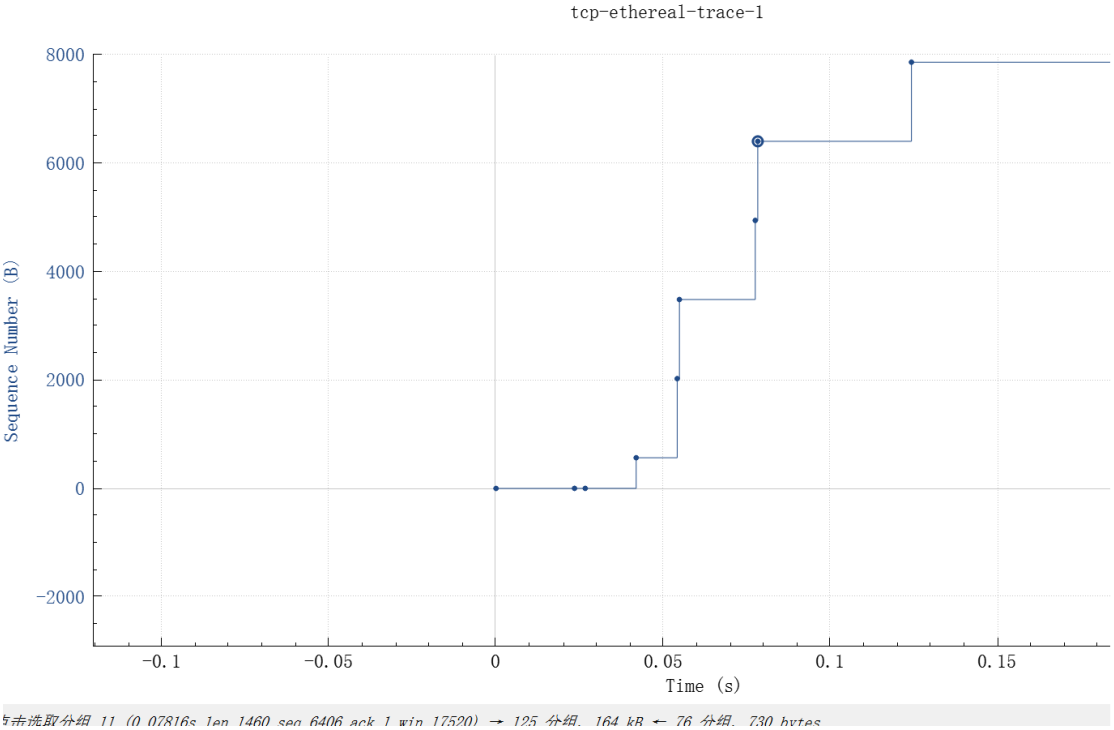
3	0.023265	192.168.1.102	128.119.245.12	TCP	54 1161 → 80 [ACK] Seq=1 Ack=1 Win=17520 Len=0
4	0.026477	192.168.1.102	128.119.245.12	TCP	619 1161 → 80 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=565
5	0.041737	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [PSH, ACK] Seq=566 Ack=1 Win=17520 Len=1460
6	0.053937	128.119.245.12	192.168.1.102	TCP	60 80 → 1161 [ACK] Seq=1 Ack=566 Win=6780 Len=0
206	5.651141	192.168.1.102	128.119.245.12	TCP	54 1161 → 80 [ACK] Seq=164091 Ack=731 Win=16790 Len=0
213	7.595557	192.168.1.102	199.2.53.206	TCP	62 1162 → 631 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM

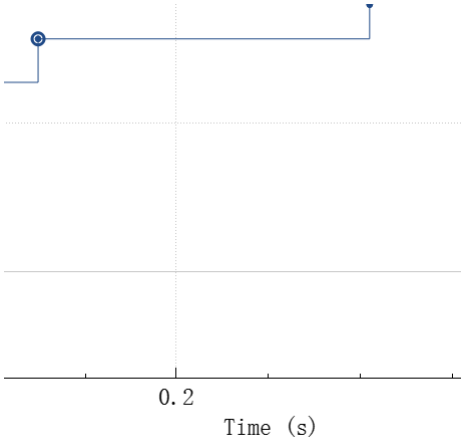
平均吞吐量 = 传输数据的比特数/接收方接收所有数据所用时间=164091*8/（5.651141-

0.026477）=233387.81bps

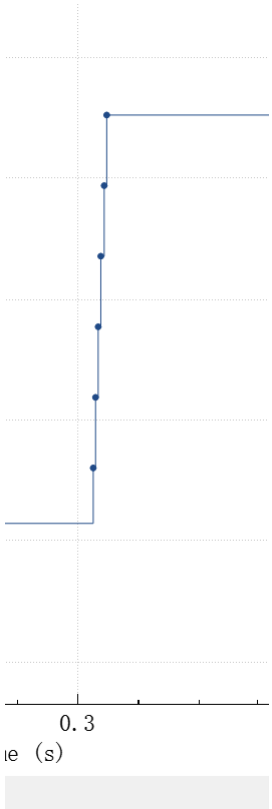
13. 能否确定 TCP 的慢启动阶段的开始和结束位置，以及拥塞控制的发生位置？讨论一下测量的数据与我们在课本中研究的 TCP 的理想化行为的不同之处。



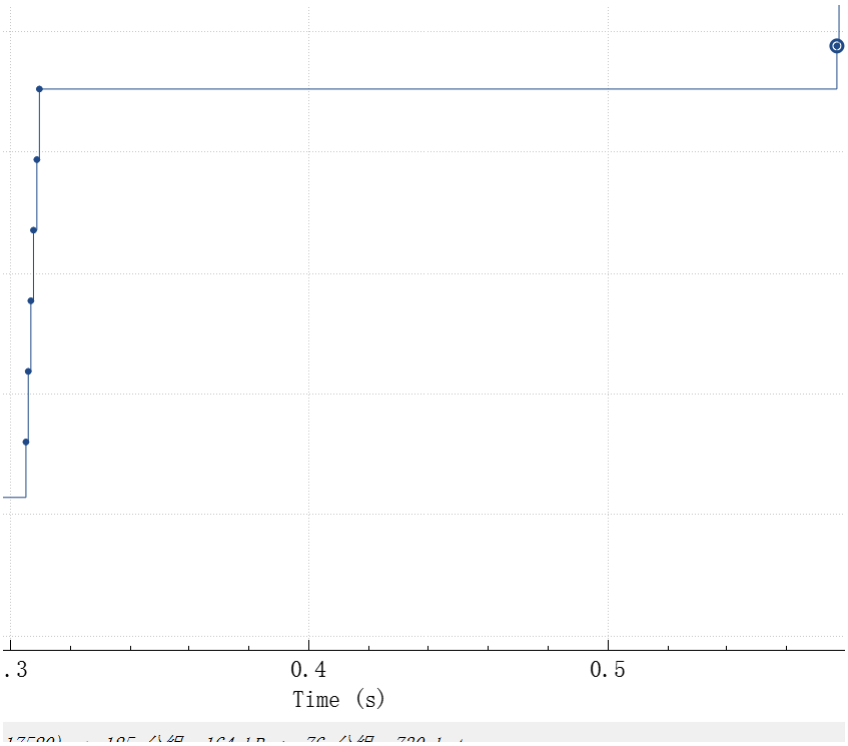




(1) 慢启动从发送了第一个 TCP 报文段后开始， 结束在观察到指数型增长的速率卡壳了，说明这个时候发生了拥塞，进入拥塞避免阶段。(如下图)

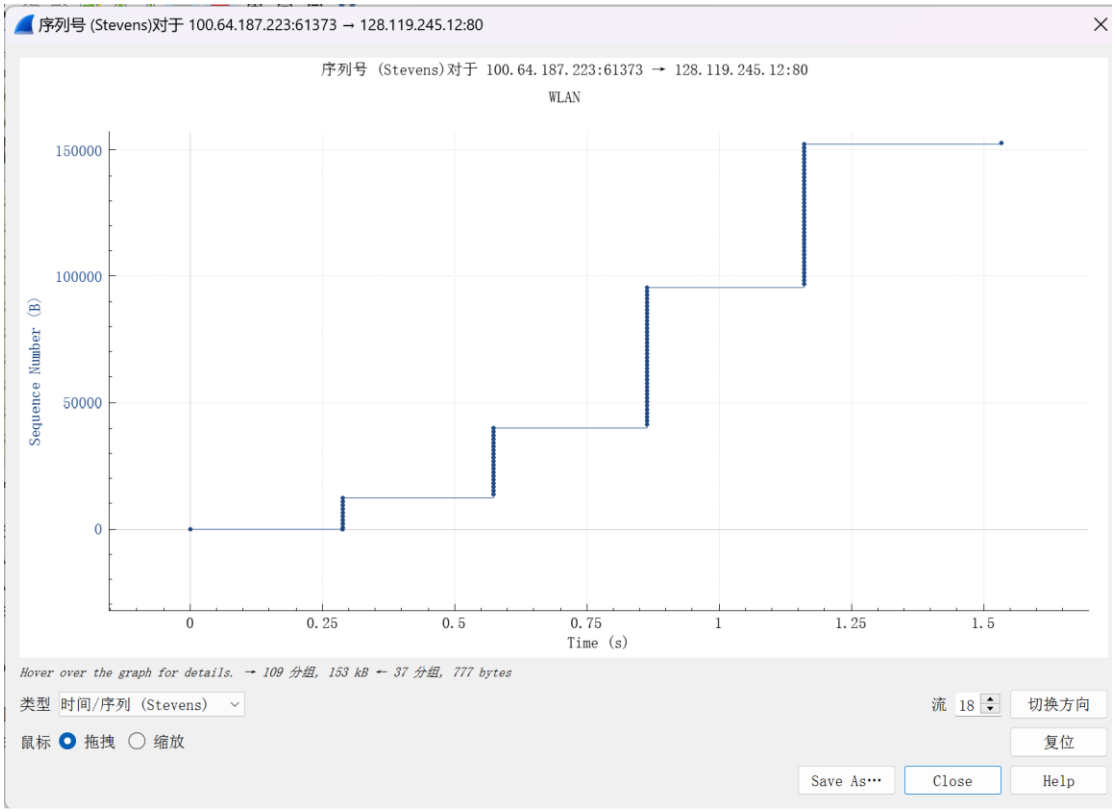


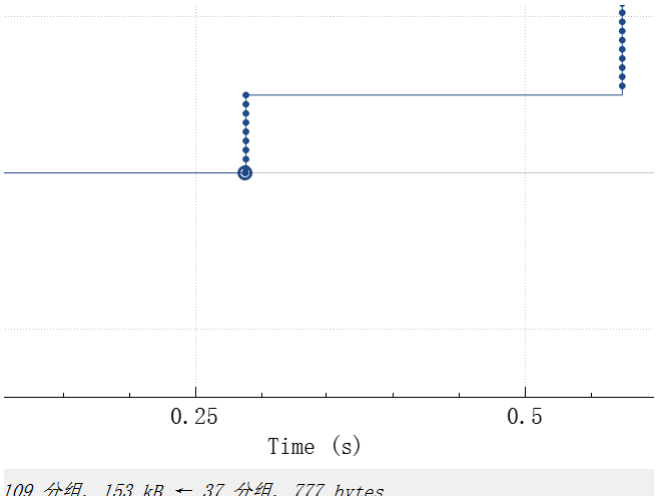
如下图这个区域就是拥塞避免区域：



（2）慢启动并不是永远都是高效的，在一些情况下效率不会达到最好。

14. 如上题





109 分组, 153 kB ← 37 分组, 777 bytes

慢启动从发送了第一个 TCP 报文段后开始， 结束在观察到指数型增长的速率卡壳了，说明这个时候发生了拥塞，进入拥塞避免阶段

94 2.401048	100.64.187.223	128.119.245.12	TCP	54 61374 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0
95 2.401389	100.64.187.223	128.119.245.12	TCP	782 61373 → 80 [PSH, ACK] Seq=1 Ack=1 Win=131328 Len=728
96 2.401723	100.64.187.223	128.119.245.12	TCP	1514 61373 → 80 [ACK] Seq=729 Ack=1 Win=131328 Len=1460

3.591540	128.119.245.12	100.64.187.223	TCP	831 80 → 61373 [PSH, ACK] Seq=1 Ack=153050 Win=242560 Len=777
3.646836	100.64.187.223	128.119.245.12	TCP	54 61373 → 80 [ACK] Seq=153050 Ack=778 Win=130560 Len=0

平均吞吐量 = 传输数据的比特数/接收方接收所有数据所用时间=153050*8/（3.646836-2.401389）=983100.85bps