

## TCP 实验

学号：PB21000224 姓名：陈鸿绪 日期：10.28

1. 文件传输到 gaia.cs.umass.edu 时，客户端计算机（源）使用的 IP 地址和 TCP 端口号是什么？要回答这个问题，最简单的方法可能是选择一个 HTTP 消息，并使用“选定数据包详细信息窗口”来探索用于传输此 HTTP 消息的 TCP 数据包的详细信息。

192.168.1.102	128.119.245.12	TCP	62 1161 → 80 [SYN] Seq=0 Win=16384 Len=0 M
128.119.245.12	192.168.1.102	TCP	62 80 → 1161 [SYN, ACK] Seq=0 Ack=1 Win=58

客户端 IP 为 192.168.1.102，TCP 端口号为 1161。

2. gaia.cs.umass.edu 的 IP 地址是什么？在此连接上发送和接收 TCP 段的端口号是多少？

192.168.1.102	128.119.245.12	TCP	62 1161 → 80 [SYN] Seq=0 Win=16384 Len=0 M
128.119.245.12	192.168.1.102	TCP	62 80 → 1161 [SYN, ACK] Seq=0 Ack=1 Win=58

gaia.cs.umass.edu 的 IP 地址为 128.119.245.12，端口号为 80。

3. 您的客户端计算机（源）向 gaia.cs.umass.edu 传输文件时使用的 IP 地址和 TCP 端口号是什么？

2130	2023-10-30 11:03:51.772346	100.64.178.188	128.119.245.12	TCP	1514 63468 → 80 [ACK] Seq=13888 Ack=1 Win=131328
------	----------------------------	----------------	----------------	-----	--

我的客户端计算机 IP 地址：100.64.178.188，TCP 端口号为 63468。

4. 用于在客户端计算机和 gaia.cs.umass.edu 之间建立 TCP 连接的 TCP SYN 段的序列号是什么？是什么标识该段作为 SYN 段？

1984	2023-10-30 11:03:49.610766	100.64.178.188	128.119.245.12	TCP	66 63468 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=z
------	----------------------------	----------------	----------------	-----	---

[Stream index: 155]  
[Conversation completeness: Complete, WITH\_DATA (31)]  
[TCP Segment Len: 0]  
Sequence Number: 0 (relative sequence number)  
Sequence Number (raw): 3481616143  
[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 .... = Accurate ECN: Not set  
....0... .... = Congestion Window Reduced: Not set  
....0... .... = ECN-Echo: Not set  
....0... .... = Urgent: Not set  
....0... .... = Acknowledgment: Not set  
....0... .... = Push: Not set  
....0... .... = Reset: Not set  
....0... ..1. = Syn: Set  
....0... .... = Fin: Not set  
[TCP Flags: .....S.]

0000 c8 33 e5 8a 5d 45 c4 23 60 f4 d9 95 08 00  
0010 00 34 d6 56 40 00 80 06 00 00 64 40 b2 bc  
0020 f5 0c f7 ec 00 50 cf 85 3f 0f 00 00 00 00  
0030 fa f0 8c a7 00 00 02 04 05 b4 01 03 03 08  
0040 04 02

和 gaia.cs.umass.edu 之间建立 TCP 连接的 TCP SYN 段的序列号是 0，可以看见报文中 Flag 可以确定 SYN 段。

5. gaia.cs.umass.edu 发送给客户端计算机作为对 SYN 的响应的 SYNACK 段的序列号是什么？SYNACK 段的确认字段的值是多少？gaia.cs.umass.edu 是如何确定该值的？是什么标识该段作为 SYNACK 段？

128.119.245.12	100.64.178.188	TCP	66 80 → 63468 [SYN, ACK] Seq=0 Ack=1 Win=29200
----------------	----------------	-----	--

gaia.cs.umass.edu 发送给客户端计算机作为对 SYN 的响应的 SYNACK 段的序列号是 0；

```

▼ Transmission Control Protocol, Src Port: 80, Dst Port: 63468, Seq: 0, Ack: 1, Len: 0
  Source Port: 80
  Destination Port: 63468
  [Stream index: 155]
  [Conversation completeness: Complete, WITH_DATA (31)]
  [TCP Segment Len: 0]
  Sequence Number: 0 (relative sequence number)
  Sequence Number (raw): 366480044
  [Next Sequence Number: 1 (relative sequence number)]
  Acknowledgment Number: 1 (relative ack number)
  Acknowledgment number (raw): 3481616144
  1000 .... = Header Length: 32 bytes (8)
▼ Flags: 0x012 (SYN, ACK)

```

SYNACK 段的确认字段的值由上为 1。ACK 用于指示确认字段中的值是有效的。  
Flags: 0x012 指示了该段为 SYNACK 段。

6. 包含 HTTP POST 命令的 TCP 段的序列号是什么？请注意，为了找到 POST 命令，您需要挖掘 Wireshark 窗口底部的数据包内容字段，寻找在其 DATA 字段中包含“POST”的段。

```

▼ Transmission Control Protocol, Src Port: 63468, Dst Port: 80, Seq: 1, Ack: 1, Len: 747
  Source Port: 63468
  Destination Port: 80
  [Stream index: 155]
  [Conversation completeness: Complete, WITH_DATA (31)]
  [TCP Segment Len: 747]
  Sequence Number: 1 (relative sequence number)
  Sequence Number (raw): 3481616144
  [Next Sequence Number: 748 (relative sequence number)]
  Acknowledgment Number: 1 (relative ack number)
  Acknowledgment number (raw): 366480045
  0101 .... = Header Length: 20 bytes (5)
▼ Flags: 0x018 (PSH, ACK)

```

包含 HTTP POST 命令的 TCP 段的序列号是 1，因为该段是 PSHACK 段。

7. 考虑包含 HTTP POST 的 TCP 段作为 TCP 连接中的第一个段。第一个六个 TCP 连接段（包括包含 HTTP POST 的段）的序列号是什么？每个段发送的时是什么时候？每个段的 ACK 何时收到？鉴于每个 TCP 段的发送时间和其确认的接收时间之间的差异，这六个段的 RTT 值是什么？每个 ACK 收到后的 EstimatedRTT 值（请参阅第 242 页文本中的第 3.5.3 节）是什么？假设 EstimatedRTT 的值等于第一个测量的 RTT 值，然后使用第 242 页上的 EstimatedRTT 方程计算所有后续段的值。

**本题采用实验中提供的抓包数据分析：**

第一段：

```
2004-08-21 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
```

序号：1，长度：565，时间：2004-08-21 21: 44: 20.596858。

2004-08-21 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
2004-08-21 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
2004-08-21 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
2004-08-21 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
2004-08-21 21:44:20.647786	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [ACK] Seq=4946 Ack=1 Win=17520 Len=1460
2004-08-21 21:44:20.648538	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [ACK] Seq=6406 Ack=1 Win=17520 Len=1460
2004-08-21 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
2004-08-21 21:44:20.694566	192.168.1.102	128.119.245.12	TCP	1201 1161 → 80 [PSH, ACK] Seq=7866 Ack=1 Win=17520 Len=

Sequence Number (raw): 883061786	0000 00 20 e0 8a 70 1a 00
[Next Sequence Number: 1 (relative sequence number)]	0010 00 28 58 72 40 00 37
Acknowledgment Number: 566 (relative ack number)	0020 01 66 00 50 04 89 34
Acknowledgment number (raw): 232129578	0030 1a 7c 9e 30 00 00 da
0101 ... = Header Length: 20 bytes (5)	
> Flags: 0x010 (ACK)	
Window: 6780	
[Calculated window size: 6780]	
[Window size scaling factor: -2 (no window scaling used)]	
Checksum: 0x9e30 [unverified]	
[Checksum Status: Unverified]	
Urgent Pointer: 0	
> [Timestamps]	
✓ [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]	

ACK 在 2004-08-21 21: 44: 20.624318 时刻收到。

EstimatedRTT=RTT=0.027460000 s

第二段:

5 2004-08-21 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
------------------------------	---------------	----------------	-----	--

9 2004-08-21 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 2004-08-21 21:44:20.647786	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [ACK] Seq=4946 Ack=1 Win=17520 Len=1460
11 2004-08-21 21:44:20.648538	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [ACK] Seq=6406 Ack=1 Win=17520 Len=1460
12 2004-08-21 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
13 2004-08-21 21:44:20.694566	192.168.1.102	128.119.245.12	TCP	1201 1161 → 80 [PSH, ACK] Seq=7866 Ack=1 Win=17520 Len=1147

Sequence Number (raw): 883061786	0000 00 20 e0 8a 70 1a 00 06 25 da af
[Next Sequence Number: 1 (relative sequence number)]	0010 00 28 58 73 40 00 37 06 b3 ca 80
Acknowledgment Number: 2026 (relative ack number)	0020 01 66 00 50 04 89 34 a2 74 1a 0d
Acknowledgment number (raw): 232131038	0030 22 38 90 c0 00 00 87 9e 00 00 3a
0101 ... = Header Length: 20 bytes (5)	
> Flags: 0x010 (ACK)	
Window: 8760	
[Calculated window size: 8760]	
[Window size scaling factor: -2 (no window scaling used)]	
Checksum: 0x90c0 [unverified]	
[Checksum Status: Unverified]	
Urgent Pointer: 0	
> [Timestamps]	
✓ [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]	

ACK 在 2004-08-21 21: 44: 20.647675 时刻收到。

RTT=0.035557000 s

EstimatedRTT=0.125\*RTT+0.875\* EstimatedRTT=0.028472125 s

第三段:

7 2004-08-21 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
------------------------------	---------------	----------------	-----	--

序号: 2026, 长度: 1460, 时间: 2004-08-21 21: 44: 20.624407。

12	2004-08-21 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
13	2004-08-21 21:44:20.694566	192.168.1.102	128.119.245.12	TCP	1201	1161 → 80 [PSH, ACK] Seq=7866 Ack=1 Win=17520 Len=1

Sequence Number (raw): 883061786	0000	00 20 e0 8a 70 1a 00 06 2!
[Next Sequence Number: 1 (relative sequence number)]	0010	00 28 58 74 40 00 37 06 b:
Acknowledgment Number: 3486 (relative ack number)	0020	01 66 00 50 04 89 34 a2 7:
Acknowledgment number (raw): 232132498	0030	2d a0 7f a4 00 00 7b ec 0k
0101 .... = Header Length: 20 bytes (5)		
> Flags: 0x010 (ACK)		
Window: 11680		
[Calculated window size: 11680]		
[Window size scaling factor: -2 (no window scaling used)]		
Checksum: 0x7fa4 [unverified]		
[Checksum Status: Unverified]		
Urgent Pointer: 0		
> [Timestamps]		
▼ [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]		

ACK 在 2004-08-21 21: 44: 20.694466 时刻收到。

RTT=0.70059000 s

EstimatedRTT=0.125\*RTT+0.875\* EstimatedRTT=0.0336705 s

第四段:

8	2004-08-21 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
---	----------------------------	---------------	----------------	-----	------	---

14	2004-08-21 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	2004-08-21 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

Sequence Number (raw): 883061786	0000	00 20 e0 8a 70 1a 00 06 25
[Next Sequence Number: 1 (relative sequence number)]	0010	00 28 58 75 40 00 37 06 b3
Acknowledgment Number: 4946 (relative ack number)	0020	01 66 00 50 04 89 34 a2 74
Acknowledgment number (raw): 232133958	0030	39 08 6e 88 00 00 d4 3c 00
0101 .... = Header Length: 20 bytes (5)		
> Flags: 0x010 (ACK)		
Window: 14600		
[Calculated window size: 14600]		
[Window size scaling factor: -2 (no window scaling used)]		
Checksum: 0x6e88 [unverified]		
[Checksum Status: Unverified]		
Urgent Pointer: 0		
> [Timestamps]		
▼ [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]		

ACK 在 2004-08-21 21: 44: 20.739499 时刻收到。

RTT=0.114428000 s

EstimatedRTT=0.125\*RTT+0.875\* EstimatedRTT=0.0437652 s

第五段:

10	2004-08-21 21:44:20.647786	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=4946 Ack=1 Win=17520 Len=1460
----	----------------------------	---------------	----------------	-----	------	---

15	2004-08-21 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	2004-08-21 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

Sequence Number (raw): 883061786	0000	00 20 e0 8a 70 1a 00 06 25 da a:
[Next Sequence Number: 1 (relative sequence number)]	0010	00 28 58 76 40 00 37 06 b3 c7 8i
Acknowledgment Number: 6406 (relative ack number)	0020	01 66 00 50 04 89 34 a2 74 1a 0k
Acknowledgment number (raw): 232135418	0030	44 70 5d 6c 00 00 6a f3 00 00 b!
0101 .... = Header Length: 20 bytes (5)		
> Flags: 0x010 (ACK)		
Window: 17520		
[Calculated window size: 17520]		
[Window size scaling factor: -2 (no window scaling used)]		
Checksum: 0x5d6c [unverified]		
[Checksum Status: Unverified]		
Urgent Pointer: 0		
> [Timestamps]		
▼ [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]		

ACK 在 2004-08-21 21: 44: 20.787680 时刻收到。  
RTT=0.139894000 s  
 $EstimatedRTT=0.125*RTT+0.875* EstimatedRTT=0.0557813 s$

第六段:

11 2004-08-21 21:44:20.648538 192.168.1.102 128.119.245.12 TCP 1514 1161 → 80 [ACK] Seq=6406 Ack=1 Win=17520 Len=1460

序号: 6406, 长度: 1460, 时间: 2004-08-21 21: 44: 20.648538。

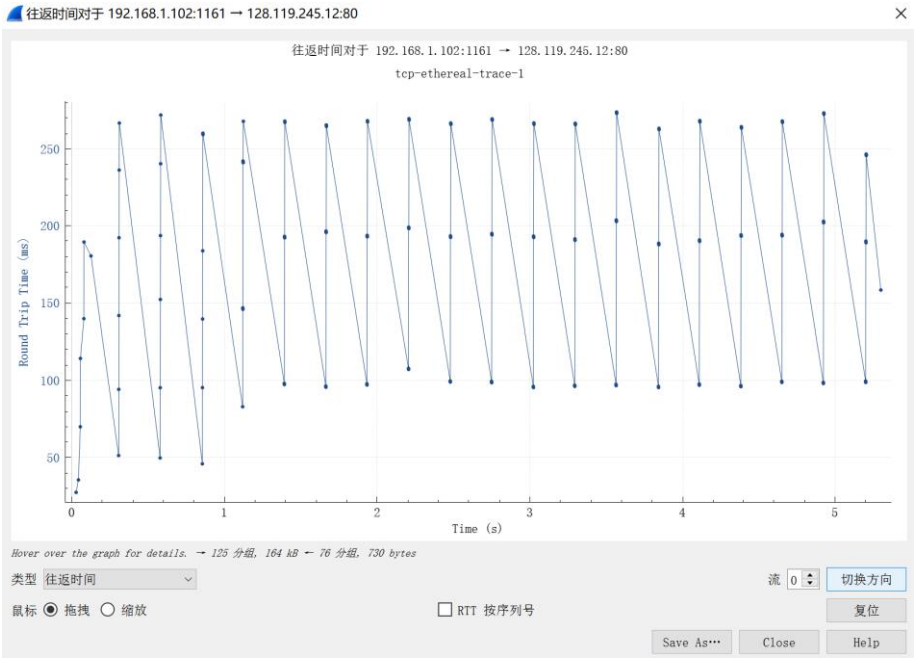
16 2004-08-21 21:44:20.838183 128.119.245.12 192.168.1.102 TCP 60 80 → 1161 [ACK] Seq=1 Ack=7866 Win=20440 Len=0  
17 2004-08-21 21:44:20.875188 128.119.245.12 192.168.1.102 TCP 60 80 → 1161 [ACK] Seq=1 Ack=9013 Win=23360 Len=0

Sequence Number (raw): 883061786  
[Next Sequence Number: 1 (relative sequence number)]  
Acknowledgment Number: 7866 (relative ack number)  
Acknowledgment number (raw): 232136878  
0101 .... = Header Length: 20 bytes (5)  
> Flags: 0x010 (ACK)  
Window: 20440  
[Calculated window size: 20440]  
[Window size scaling factor: -2 (no window scaling used)]  
Checksum: 0x4c50 [unverified]  
[Checksum Status: Unverified]  
Urgent Pointer: 0  
> [Timestamps]  
v [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]

0000 00 20 e0 8a 70 1a 00 06 25 da af 73 08 00  
0010 00 28 58 77 40 00 37 06 b3 c6 80 77 f5 0c  
0020 01 66 00 50 04 89 34 a2 74 1a 0d d6 20 ae  
0030 4f d8 4c 50 00 00 93 c0 00 00 63 ed

ACK 在 2004-08-21 21: 44: 20.838183 时刻收到。  
RTT=0.189645000 s  
 $EstimatedRTT=0.125*RTT+0.875* EstimatedRTT=0.0725142625 s$

绘图 TCP 流量:



8. 前六个 TCP 段中每个段的长度是多少？  
如第七题答案中所示。
9. 在整个跟踪中，接收端的最小可用缓冲空间是多少？接收端缓冲空间的不足是否会限制发送方？

1987	2023-10-30	11:03:49.984760	128.119.245.12	100.64.178.188	TCP	66 80 → 63468 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=:
1988	2023-10-30	11:03:49.984843	100.64.178.188	128.119.245.12	TCP	54 63468 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0
2039	2023-10-30	11:03:51.487510	100.64.178.188	128.119.245.12	TCP	801 63468 → 80 [PSH, ACK] Seq=1 Ack=1 Win=131328 Len=747
2040	2023-10-30	11:03:51.487747	100.64.178.188	128.119.245.12	TCP	1514 63468 → 80 [ACK] Seq=748 Ack=1 Win=131328 Len=1460
2041	2023-10-30	11:03:51.487747	100.64.178.188	128.119.245.12	TCP	1514 63468 → 80 [ACK] Seq=2208 Ack=1 Win=131328 Len=1460
2042	2023-10-30	11:03:51.487747	100.64.178.188	128.119.245.12	TCP	1514 63468 → 80 [ACK] Seq=3668 Ack=1 Win=131328 Len=1460

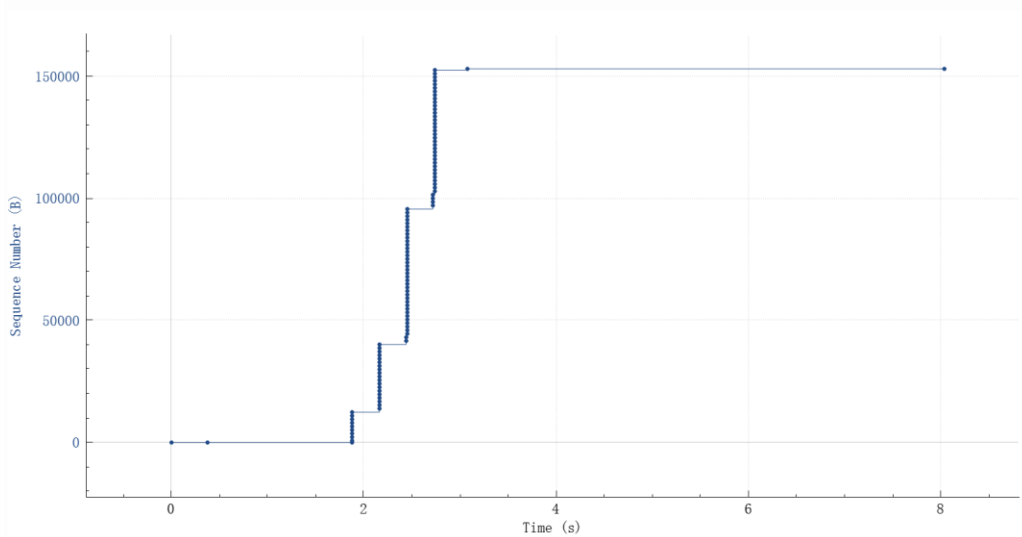
  

> 010. .... = Flags: 0x2, Don't fragment ...0 0000 0000 0000 = Fragment Offset: 0 Time to Live: 28 Protocol: TCP (6) Header Checksum: 0xd243 [validation disabled] [Header checksum status: Unverified] Source Address: 128.119.245.12 Destination Address: 100.64.178.188		0000 c4 23 60 f4 d9 95 c8 33 e5 8a 5d 45 08 00 0010 00 34 00 00 00 00 1c 06 d2 43 80 77 f5 0c 0020 b2 bc 00 50 f7 ec 15 d8 0a ac cf 85 3f 10 0030 72 10 49 19 00 00 02 04 05 b4 01 01 04 02 0040 03 07
Transmission Control Protocol, Src Port: 80, Dst Port: 63468, Seq: 0, Ack: 1, Len: 0 Source Port: 80 Destination Port: 63468 [Stream index: 155] [Conversation completeness: Complete, WITH_DATA (31)] [TCP Segment Len: 0] Sequence Number: 0 (relative sequence number) Sequence Number (raw): 366480044 [Next Sequence Number: 1 (relative sequence number)] Acknowledgment Number: 1 (relative ack number) Acknowledgment number (raw): 3481616144 1000 .... = Header Length: 32 bytes (8)		
Flags: 0x012 (SYN, ACK)		

根据本人实验的数据，最小可用缓冲空间是 win=29200。由 TCP 流量控制系统，接收端缓冲空间的不足会限制发送方。

10. 跟踪文件中是否有任何重传的段？为了回答这个问题，你检查了什么？

根据本人实验的数据，检查按照序列号关于时间的往返情况：



明显对于同一个序列并没有对应两个横坐标，所以没有重传的段。

11. 接收端通常在 ACK 中确认多少数据？是否可以找出来 ACK 确认两个或者多个 TCP 段的情况。

根据本人实验的数据，接收端通常在 ACK 中确认大小为 1460bit 的数据。

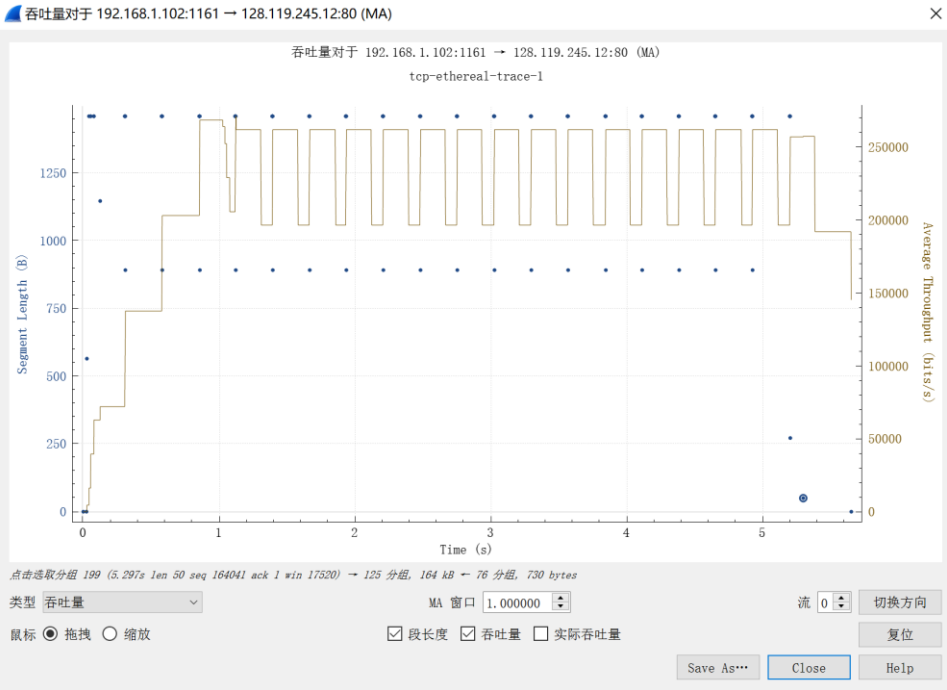


2041	2023-10-30	11:03:51.487747	100.64.178.188	128.119.245.12	TCP	1514	63468 → 80 [ACK]	Seq=2208 Ack=1 Win=131328 Len=1460
2042	2023-10-30	11:03:51.487747	100.64.178.188	128.119.245.12	TCP	1514	63468 → 80 [ACK]	Seq=3668 Ack=1 Win=131328 Len=1460
2043	2023-10-30	11:03:51.487747	100.64.178.188	128.119.245.12	TCP	1514	63468 → 80 [ACK]	Seq=5128 Ack=1 Win=131328 Len=1460
2044	2023-10-30	11:03:51.487747	100.64.178.188	128.119.245.12	TCP	1514	63468 → 80 [ACK]	Seq=6588 Ack=1 Win=131328 Len=1460
2045	2023-10-30	11:03:51.487747	100.64.178.188	128.119.245.12	TCP	1514	63468 → 80 [ACK]	Seq=8048 Ack=1 Win=131328 Len=1460
2046	2023-10-30	11:03:51.487747	100.64.178.188	128.119.245.12	TCP	1514	63468 → 80 [ACK]	Seq=9508 Ack=1 Win=131328 Len=1460
2047	2023-10-30	11:03:51.487747	100.64.178.188	128.119.245.12	TCP	1514	63468 → 80 [ACK]	Seq=10968 Ack=1 Win=131328 Len=1460
2048	2023-10-30	11:03:51.487747	100.64.178.188	128.119.245.12	TCP	1514	63468 → 80 [ACK]	Seq=12428 Ack=1 Win=131328 Len=1460
2129	2023-10-30	11:03:51.772272	128.119.245.12	100.64.178.188	TCP	60	80 → 63468 [ACK]	Seq=1 Ack=748 Win=30720 Len=0
2130	2023-10-30	11:03:51.772346	100.64.178.188	128.119.245.12	TCP	1514	63468 → 80 [ACK]	Seq=13888 Ack=1 Win=131328 Len=1460
2131	2023-10-30	11:03:51.772586	128.119.245.12	100.64.178.188	TCP	60	80 → 63468 [ACK]	Seq=1 Ack=8048 Win=45312 Len=0

由上图中左边序号为 2041、2042、2043、2044 是客户端发送给服务器段的 TCP 段，seq 为 2208、3668、5128 和 6588，然而 2208、3668、5128 并没有等来相应的 ACK，而是直接 ACK8048 将四个 TCP 段一起确认。

12. TCP 连接的吞吐量（每单位时间传输的字节数）是多少？解释一下你是如何计算这个值的。

本题采用实验中提供的抓包数据分析：

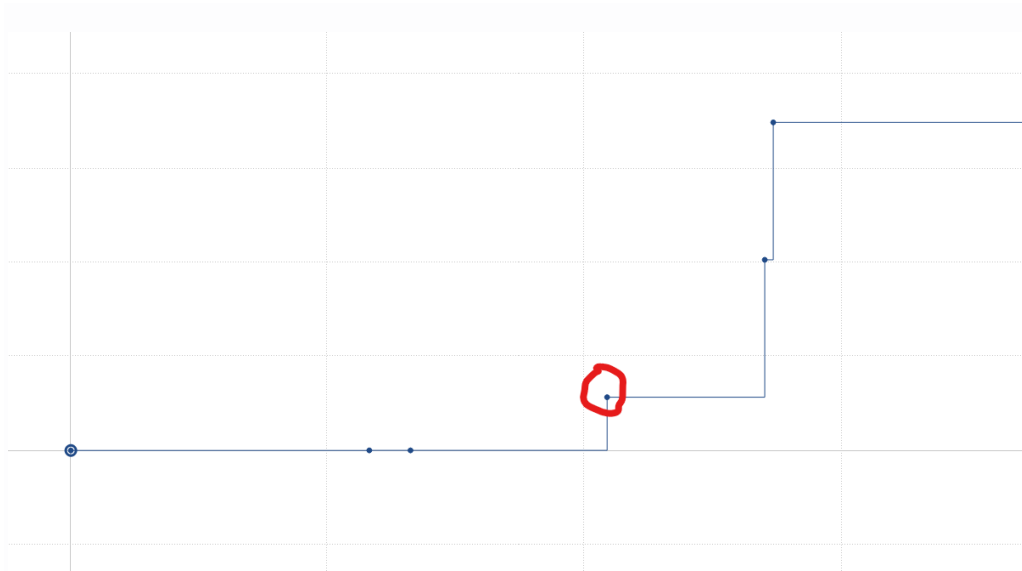


4	2004-08-21	21:44:20.596858	192.168.1.102	128.119.245.12	TCP	619	1161 → 80 [PSH, ACK]	Seq=1 Ack=1 Win=17520 Len=565
199	2004-08-21	21:44:25.867722	192.168.1.102	128.119.245.12	TCP	104	1161 → 80 [PSH, ACK]	Seq=164041 Ack=1 Win=17520 Len=50

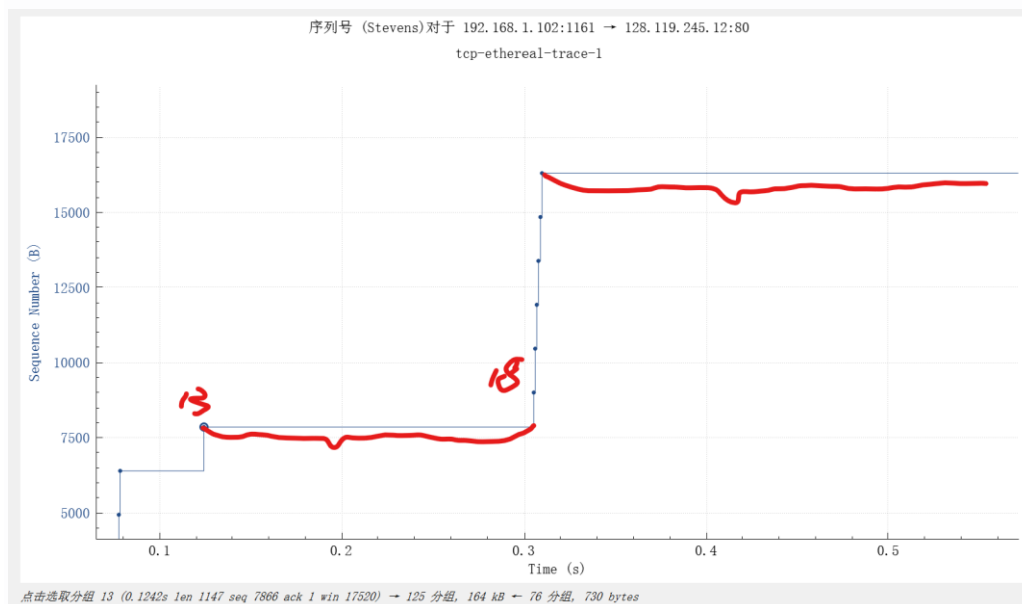
$T=25.867722s-20.596858s=5.4294\text{ s}$   
吞吐量= $(164041-1)\text{bit}/T=30,213.2832=30.213\text{ kb/s}$

13. 使用 Time-Sequence-Graph(Stevens)绘图工具查看从客户端到 gaia.cs.umass.edu 服务器发送的段的序列号与时间的图。您能否确定 TCP 慢启动阶段开始位置，以及拥塞避免接管的位置？请对我们文本中研究的 TCP 理想化行为与测量数据的差异进行评论。

本题采用实验中提供的抓包数据分析：



上图红色所圈位置 TCP 段第二段（编号为 5），同时也为慢启动开始的位置。



以上图中红线所标识的为拥塞避免机制起作用的地方。

评价：慢启动在数据较少的情况下，且网络畅通情况下，可能在慢启动结束之前就将报文已经传输完毕，也就是说慢启动可能在某些情况下影响发送效率。但是总体对于实际情况而言，该做法是有益于避免网络拥塞的。

14. 回答您从计算机传输文件到 [gaia.cs.umass.edu](http://gaia.cs.umass.edu) 时收集的跟踪文件中的两个问题。

1. 一开始发现有的 TCP 段并没有找到相应的 ACK，后面才知道可以一个 ACK 应答多个相连的 TCP 段。
2. 在本人的实验中感觉拥塞避免机制并没非常明显的体现，所以采用了实验中所提供的实验数据。