Lab4 TCP

PB21111723 王涵

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

使用实验文档中提供包tcp-ethereal-trace-1的截图，

	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.102	128.119.245.12	TCP	62	1161 → 80 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_
2	0.023172	128.119.245.12	192.168.1.102	TCP	62	80 → 1161 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=
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=146
6	0.053937	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=566 Win=6780 Len=0

ip: 192.168.1.102

TCP端口号：1161

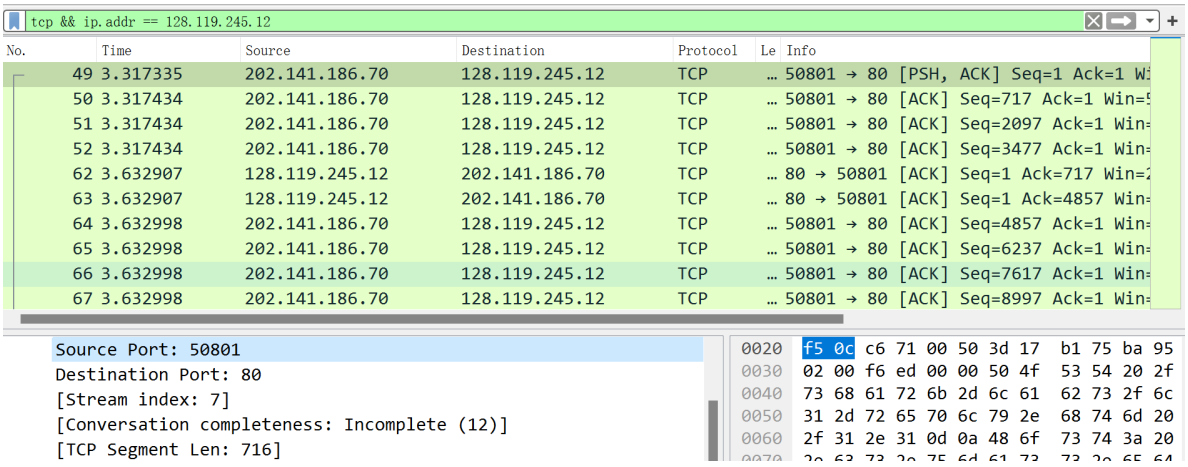
> 2.gaia.cs.umass.edu 的 IP 地址是什么? 在哪个端口号上发送和接收此连接的 TCP 区段?

IP 地址：128.119.245.12

接收连接的端口号：80

> 3.用于在客户端计算机和 gaia.cs.umass.edu 之间启动 TCP 连接的 TCP SYN 区段的序列号是什么? 将区段标识为 SYN 区段的区段有什么功能?

截图如下，



我的客户计算机使用的 IP 地址为 `202.141.186.70`，端口号为 `50801`.

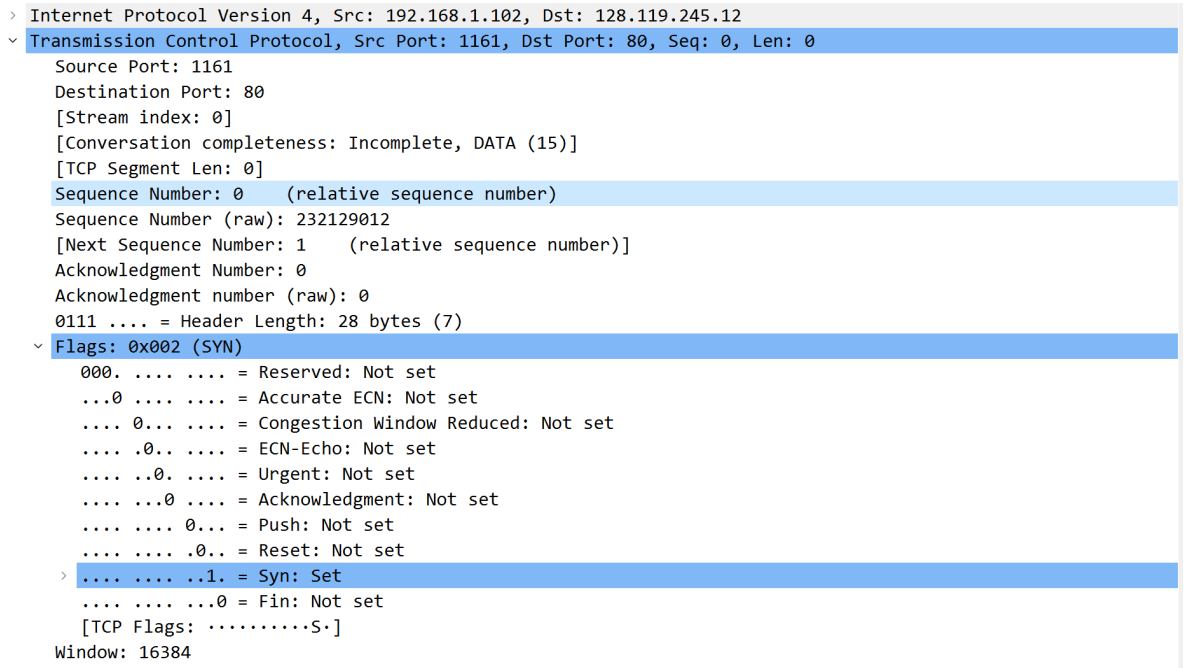
> 4.用于在客户端计算机和 gaia.cs.umass.edu 之间启动 TCP 连接的TCP SYN网段的

> 序号是多少？段中标识该段为 SYN 段的内容是什么？

使用实验文档中提供包tcp-ethereal-trace-1的截图，

可以看出，序列号为 `0` ；

Flags设置为 `0x002` ，Syn字段设置为1 (Syn:Set) ，Flags中只含Syn标志位表示它是 SYN 报文段.



> 5.gaia.cs.umass.edu 发送给客户端计算机以回复 SYN 的 SYNACK 网段的序列号是多少？SYNACK 网段中Acknowledgement字段的值是多少？gaia.cs.umass.edu 如何确定该值？该网段中标识该网段为 SYNACK 网段的内容是什么？

使用实验文档中提供包tcp-ethereal-trace-1的截图，

序列号为 `0` ；

Acknowledgement字段值为 `1` ；

Acknowledgement字段的值等于客户发出的SYN序列号加 1 ；

Flags 字段被设为 `0x012` ，Acknowledgement和Syn字段设置为1 (Acknowledgement:Set && Syn:Set) ,Flags中两个标志位都被设置，所以它是 SYNACK 报文;

```
> Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.1.102
v Transmission Control Protocol, Src Port: 80, Dst Port: 1161, Seq: 0, Ack: 1, Len: 0
    Source Port: 80
    Destination Port: 1161
    [Stream index: 0]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 883061785
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 232129013
    0111 .... = Header Length: 28 bytes (7)
  v Flags: 0x012 (SYN, ACK)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Accurate ECN: Not set
      .... 0... .... = Congestion Window Reduced: 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
      [TCP Flags: ·····A··S·]
    Window: 5840
```

> 6.包含 HTTP POST 命令的 TCP 网段的序列号是多少？请注意，为了找到POST 命令，您需要深入 Wireshark 窗口底部的数据包内容字段，寻找DATA 字段中包含"POST"的数据段

使用实验文档中提供包tcp-ethereal-trace-1的截图，

序列号为 `1`；

```
> Frame 4: 619 bytes on wire (4952 bits), 619 bytes captured (4952 bit
> Ethernet II, Src: Actionte_8a:70:1a (00:20:e0:8a:70:1a), Dst: Linksy
> Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.119.245.12
v Transmission Control Protocol, Src Port: 1161, Dst Port: 80, Seq: 1,
    Source Port: 1161
    Destination Port: 80
    [Stream index: 0]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 565]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 232129013
    [Next Sequence Number: 566    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 883061786
    0101 .... = Header Length: 20 bytes (5)
  v Flags: 0x018 (PSH, ACK)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Accurate ECN: Not set
      .... 0... .... = Congestion Window Reduced: 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
```

```
0020  f5 0c 04 89 00 50 0d d6  01 f5 34
0030  44 70 1f bd 00 00 50 4f  53 54 20
0040  72 65 61 6c 2d 6c 61 62  73 2f 6c
0050  2d 72 65 70 6c 79 2e 68  74 6d 20
0060  31 2e 31 0d 0a 48 6f 73  74 3a 20
0070  63 73 2e 75 6d 61 73 73  2e 65 64
0080  65 72 2d 41 67 65 6e 74  3a 20 4d
0090  61 2f 35 2e 30 20 28 57  69 6e 64
00a0  55 3b 20 57 69 6e 64 6f  77 73 20
00b0  31 3b 20 65 6e 2d 55 53  3b 20 72
00c0  2e 32 29 20 47 65 63 6b  6f 2f 32
00d0  30 38 20 4e 65 74 73 63  61 70 65
00e0  0d 0a 41 63 63 65 70 74  3a 20 74
00f0  6d 6c 2c 61 70 70 6c 69  63 61 74
0100  6d 6c 2c 61 70 70 6c 69  63 61 74
0110  68 74 6d 6c 2b 78 6d 6c  2c 74 65
0120  6d 6c 3b 71 3d 30 2e 39  2c 74 65
0130  61 69 6e 3b 71 3d 30 2e  38 2c 76
0140  78 2d 6d 6e 67 2c 69 6d  61 67 65
0150  69 6d 61 67 65 2f 6a 70  65 67 2c
0160  2f 67 69 66 3b 71 3d 30  2e 32 2c
0170  63 73 73 2c 2a 2f 2a 3b  71 3d 30
0180  63 63 65 70 74 2d 4c 61  6e 67 75
0190  65 6e 2d 75 73 2c 20 65  6e 3b 71
01a0  0d 0a 41 63 63 65 70 74  2d 45 6e
01b0  67 3a 20 67 7a 69 70 2c  20 64 65
01c0  2c 20 63 6f 6d 70 72 65  73 73 3b
01d0  0d 0a 41 63 63 65 70 74  2d 43 68
01e0  3a 20 49 53 4f 2d 38 38  35 39 2d
```

> 7.将包含 HTTP POST 的 TCP 段视为 TCP 连接中的第一个网段。连接中前六个段的序列号是多少？

使用实验文档中提供包tcp-ethereal-trace-1的截图，

这6个段分别对应编号为4,5,7,8,10,11的包，

No.	Time	Source	Destination	Protc	Length	Info
1	0.000000	192.168.1.102	128.119.245.12	TCP	62	1161 → 80 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM
2	0.023172	128.119.245.12	192.168.1.102	TCP	62	80 → 1161 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM
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 [TCP segment of a reassem…
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 [TCP segment of a reas…
6	0.053937	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=566 Win=6780 Len=0
7	0.054026	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=2026 Ack=1 Win=17520 Len=1460 [TCP segment of a reassemb…
8	0.054690	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=3486 Ack=1 Win=17520 Len=1460 [TCP segment of a reassemb…
9	0.077294	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=2026 Win=8760 Len=0
10	0.077405	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=4946 Ack=1 Win=17520 Len=1460 [TCP segment of a reassemb…
11	0.078157	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=6406 Ack=1 Win=17520 Len=1460 [TCP segment of a reassemb…
12	0.124085	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=3486 Win=11680 Len=0
13	0.124185	192.168.1.102	128.119.245.12	TCP	1201	1161 → 80 [PSH, ACK] Seq=7866 Ack=1 Win=17520 Len=1147 [TCP segment of a rea…

其ACK报文分别对应编号为6,9,12,14,15,16的包。

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 [TCP segment of a rea…
6	0.053937	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=566 Win=6780 Len=0
7	0.054026	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=2026 Ack=1 Win=17520 Len=1460 [TCP segment of a reassem…
8	0.054690	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=3486 Ack=1 Win=17520 Len=1460 [TCP segment of a reassem…
9	0.077294	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=2026 Win=8760 Len=0
10	0.077405	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=4946 Ack=1 Win=17520 Len=1460 [TCP segment of a reassem…
11	0.078157	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=6406 Ack=1 Win=17520 Len=1460 [TCP segment of a reassem…
12	0.124085	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=3486 Win=11680 Len=0
13	0.124185	192.168.1.102	128.119.245.12	TCP	1201	1161 → 80 [PSH, ACK] Seq=7866 Ack=1 Win=17520 Len=1147 [TCP segment of a re…
14	0.169118	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=4946 Win=14600 Len=0
15	0.217299	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=6406 Win=17520 Len=0
16	0.267802	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=7866 Win=20440 Len=0
17	0.304807	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=9013 Win=23360 Len=0
18	0.305040	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=9013 Ack=1 Win=17520 Len=1460 [TCP segment of a reassem…

则由题意得，如下表所示

	No.	序号	报文发送时间	对应ACK No.	ACK 接收时间	RTT	接收到 ACK 报文后的 EstimatedRTT
报文1	4	1	0.026477	6	0.053937	0.027460	0.027460
报文2	5	566	0.041737	9	0.077294	0.035557	0.028472
报文3	7	2026	0.054026	12	0.124085	0.070059	0.033670
报文4	8	3486	0.054690	14	0.169118	0.114428	0.043765
报文5	10	4946	0.077405	15	0.217299	0.139894	0.055781
报文6	11	6406	0.078157	16	0.267802	0.189645	0.072514

> 8.前六个 TCP 网段的长度各是多少?

`565`, `1460`, `1460`, `1460`, `1460`, `1460` 字节。

> 9.在整个跟踪过程中，接收方的可用缓冲空间最小值是多少？接收端缓冲空间不足是否会对发送端造成影响?

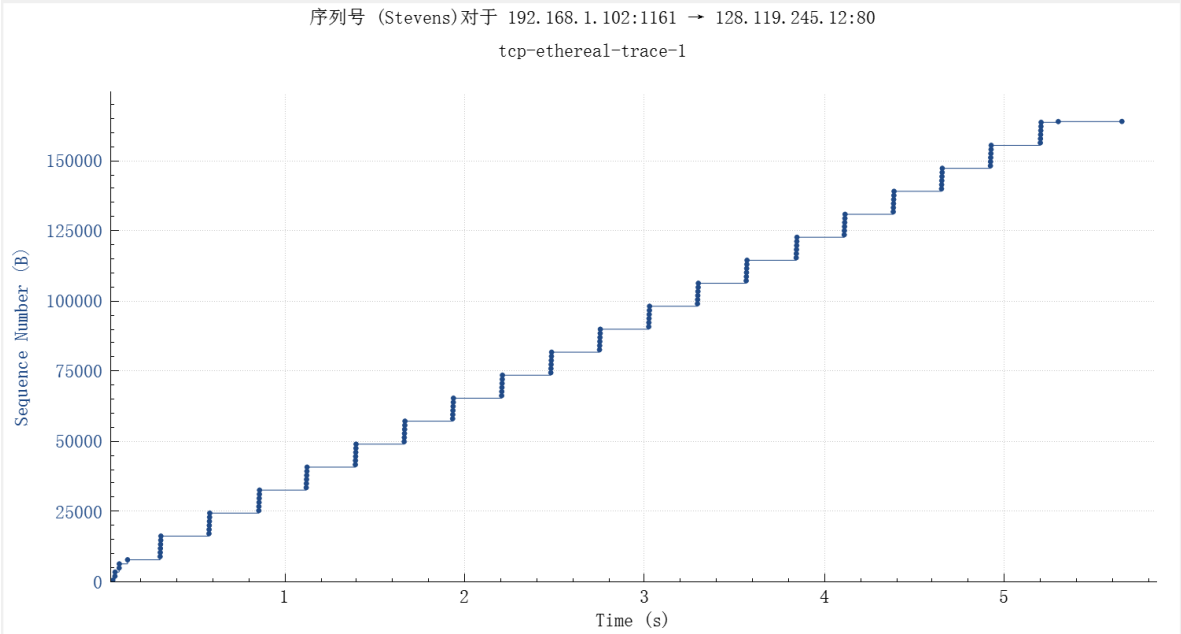
Calculated window size是5840最小可用缓冲区空间为 `5840` 字节。

接收方的缓冲区空间不会影响到发送端。因为接收方 buffer 一直有空间，并且请求方发送的 TCP 报文长度都小于buffer 剩余值，所以TCP段的可用空间总是多余的。

```
› Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.1.102
⌄ Transmission Control Protocol, Src Port: 80, Dst Port: 1161, Seq: 0, Ack: 1, Len: 0
    Source Port: 80
    Destination Port: 1161
    [Stream index: 0]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 883061785
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 232129013
    0111 .... = Header Length: 28 bytes (7)
  › Flags: 0x012 (SYN, ACK)
    Window: 5840
    [Calculated window size: 5840]
    Checksum: 0x774d [unverified]
    [Checksum Status: Unverified]
```

> 10.跟踪文件中有重发的文段吗？为了回答这个问题，您（在跟踪文件中）检查了哪些内容?

没有重发的报文。可以观察Stevens图，随着时间推移，发送报文的序号严格递增，而如果有重发的报文，序号是会减少的。



> 11.接收方通常在一个 ACK 中确认多少数据？您能否找出接收方每收到一个数据段就 ACK 一次的情况（见正文第 250 页表 3.2）？

接收方通常在一个ACK中确认1460字节的数据；

可以找到题中所述的情况，No.60和No.61，Ack=37969和Ack=40889，40889-37969=2920=1460*2，是最大报文段的二倍，满足条件。

```
 59 1.200421  128.119.245.12  192.168.1.102   TCP     60 80 → 1161 [ACK] Seq=1 Ack=35049 Win=62780 Len=0
 60 1.265026  128.119.245.12  192.168.1.102   TCP     60 80 → 1161 [ACK] Seq=1 Ack=37969 Win=62780 Len=0
 61 1.362074  128.119.245.12  192.168.1.102   TCP     60 80 → 1161 [ACK] Seq=1 Ack=40889 Win=62780 Len=0
 62 1.389886  128.119.245.12  192.168.1.102   TCP     60 80 → 1161 [ACK] Seq=1 Ack=41781 Win=62780 Len=0
 63 1.390110  192.168.1.102   128.119.245.12  TCP   1514 1161 → 80 [ACK] Seq=41781 Ack=1 Win=17520 Len=1460 [TCP segment of a reasse…
```

> 12.TCP 连接的吞吐量（单位时间内传输的字节数）是多少？请解释您是如何计算出这一数值的。

第一个有POST请求的TCP报文对应No.4，发送时间为0.026477s，Ack=1，

```
4 0.026477  192.168.1.102   128.119.245.12  TCP    619 1161 → 80 [PSH, ACK] Seq=1 Ack=1 Win=17520
```

最后一个ACK对应No.202，接收时间为5.455830s，Ack=164091，

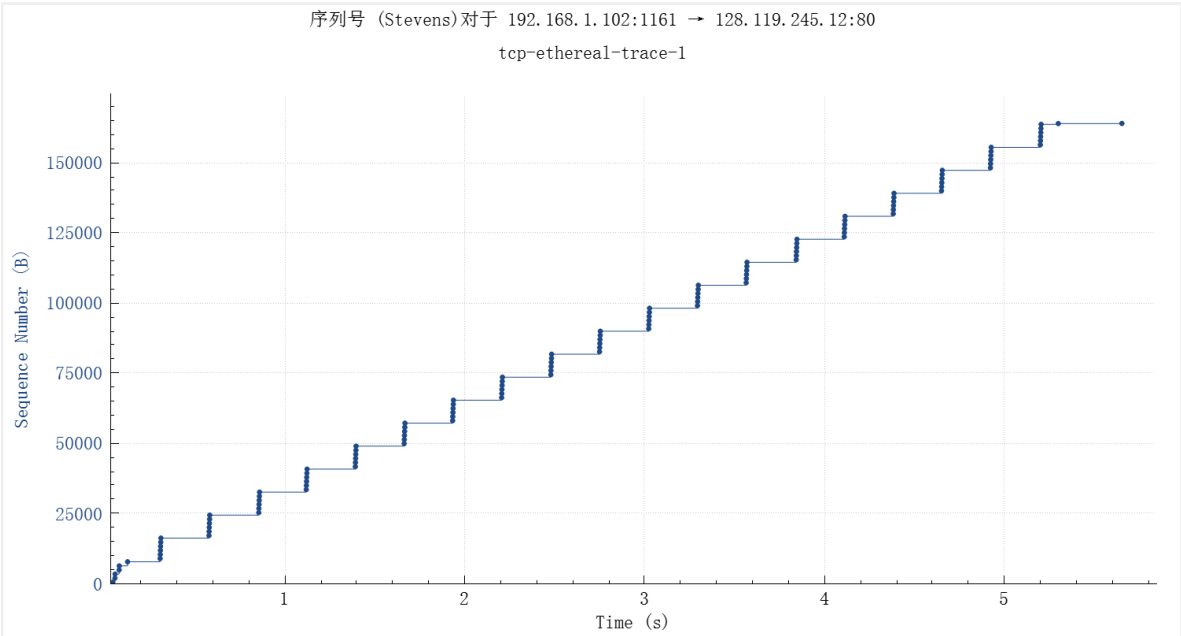
```
202 5.455830  128.119.245.12  192.168.1.102   TCP     60 80 → 1161 [ACK] Seq=1 Ack=164091 Win=62780
```

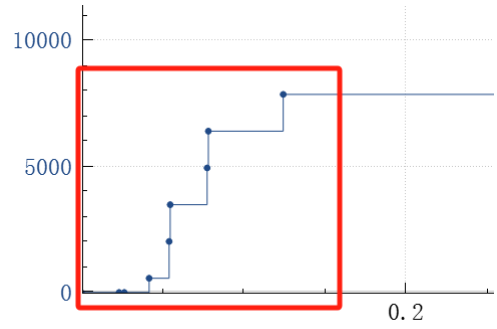
故吞吐量为 $\frac{164091-1}{5.455830-0.026477}=\frac{164090}{5.429353}=30222.75B/s$

> 13.使用时间序列图Stevens绘图工具查看从客户端发送到gaia.cs.umass.edu 服务器的数据段的序列号与时间的关系图。您能确定TCP 的慢启动阶段在哪里开始和结束，以及拥塞避免在哪里接管吗？请就测量数据与我们在课文中学习的理想化 TCP 行为的不同之处发表评论。

作Stevens图如下，从图中看到，TCP的慢启动阶段大致是在0~0.125s时间，后续在大约0.3s就进入拥塞避免阶段。

其与理想化的 TCP 行为不同的是，在慢启动后拥塞避免状态下，cwnd固定为一个常数6，而不是线性增长.





> 14.就您从计算机向 gaia.cs.umass.edu 传输文件时收集到的痕迹，回答上述两个问题中的每个问题。

作Stevens图如下，容易看出，TCP的慢启动阶段大致是在0~1.3s时间内，1.3s后开始拥塞避免阶段。

其与理想化的 TCP 行为不同的是，在慢启动时并不是理想的指数增长；且拥塞避免状态下，cwnd固定为一个常数，而不是线性增长.