

TCP 实验报告

胡延伸 PB22050983

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

由于本地网络抓取 packets 时干扰过大，因此读取官方提供的 Trace 文件。

打开后如下图：

No.	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=
2	0.023172	128.119.245.12	192.168.1.102	TCP	62	80 → 1161 [SYN, ACK] Seq=0 Ack=1 Win=5840
3	0.023265	192.168.1.102	128.119.245.12	TCP	54	1161 → 80 [ACK] Seq=1 Ack=1 Win=17520 Len=
4	0.026477	192.168.1.102	128.119.245.12	TCP	619	1161 → 80 [PSH, ACK] Seq=1 Ack=1 Win=17520
5	0.041737	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [PSH, ACK] Seq=566 Ack=1 Win=175
6	0.053937	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=566 Win=6780 Len=
7	0.054026	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=2026 Ack=1 Win=17520 L
8	0.054690	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=3486 Ack=1 Win=17520 L
9	0.077294	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=2026 Win=8760 Le
10	0.077405	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=4946 Ack=1 Win=17520 L
11	0.078157	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=6406 Ack=1 Win=17520 L
12	0.124085	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=3486 Win=11680 L
13	0.124185	192.168.1.102	128.119.245.12	TCP	1201	1161 → 80 [PSH, ACK] Seq=7866 Ack=1 Win=17
14	0.169118	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=4946 Win=14600 L
15	0.217299	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=6406 Win=17520 L
16	0.267802	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=7866 Win=20440 L
17	0.304807	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=9013 Win=23360 L
18	0.305040	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=9013 Ack=1 Win=17520 L
19	0.305813	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=10473 Ack=1 Win=17520
20	0.306692	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=11933 Ack=1 Win=17520
21	0.307571	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=13393 Ack=1 Win=17520

> Frame 1: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface 0
> Ethernet II, Src: ActiontecEle_Ba:70:1a (08:20:e0:8a:70:1a), Dst: 08:00:00:00:00:00
> Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.119.245.12
> Transmission Control Protocol, Src Port: 1161, Dst Port: 80

A first look at the captured trace

问题

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?

No.	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

如上图,本地电脑IP:192.168.1.102; TCP 端口为:1161

1. What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection?

同理, 如题 1 图: gaia.cs.umass.edu IP 为: 128.119.245.12; TCP 端口为:80

TCP Basics

问题

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 the segment that identifies the segment as a SYN segment?

No.	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_PERM

如上图, SYN序列号为0;

Transmission Control Protocol, Src Port: 1161, Dst Port: 80, Seq: 0, Len: 0	
Source Port: 1161	
Destination Port: 80	
[Stream index: 0]	
[Stream Packet Number: 1]	
> [Conversation completeness: Incomplete, DATA (15)]	
[TCP Segment Len: 0]	
Sequence Number: 0 (relative sequence number)	
Sequence Number (raw): 232129012	
[Next Sequence Number: 1 (relative sequence number)]	
Acknowledgment Number: 0	
Acknowledgment number (raw): 0	
0111 = Header Length: 28 bytes (7)	
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	
... ..1. = Syn: Set	
> [Expert Info (Chat/Sequence): Connection establish request (SYN): server port 80]	
... ..0 = Fin: Not set	
[TCP Flags:S.]	
Window: 16384	
[Calculated window size: 16384]	
Checksum: 0xf6e9 [unverified]	

如上图, segment 中 Flags 显示:0x002(SYN),表示这个 segment 为 SYN.

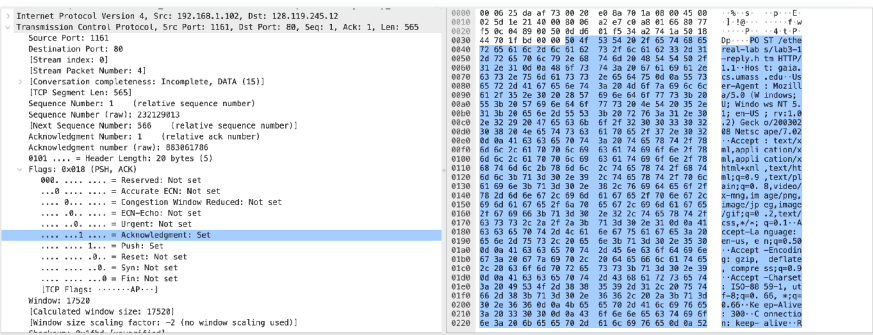
4. What is the sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment? How did

gaia.cs.umass.edu determine that value? What is it in the segment that identifies the segment as a SYNACK segment?

```
Transmission Control Protocol, Src Port: 1161, Dst Port: 80, Seq: 1, Ack: 1, Len: 0
  Source Port: 1161
  Destination Port: 80
  [Stream index: 0]
  [Stream Packet Number: 3]
  > [Conversation completeness: Incomplete, DATA (15)]
  [TCP Segment Len: 0]
  Sequence Number: 1 (relative sequence number)
  Sequence Number (raw): 232129013
  [Next Sequence Number: 1 (relative sequence number)]
  Acknowledgment Number: 1 (relative ack number)
  Acknowledgment number (raw): 883061786
  0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x010 (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
    ....0... = Syn: Not set
    ....0... = Fin: Not set
  [TCP Flags: .....A....]
  Window: 17520
  [Calculated window size: 17520]
  [Window size scaling factor: -2 (no window scaling used)]
  Checksum: 0x7671 [unverified]
```

Sequence number: 1;
Acknowledgement field: 1;
Acknowledgment:set表明 ACK字段有效，则观察 Acknowledgement 那一栏即可获取该值;
Flags field:0x010(ACK), 且下面Acknowledgment:set 表明这就是一个 SYNACK segment

5. 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.



The image shows a Wireshark packet capture of a TCP segment. The packet is selected, and the 'Packet Details' pane on the left shows the following structure:

- Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.119.245.12
- Transmission Control Protocol, Src Port: 1161, Dst Port: 80, Seq: 1, Ack: 1, Len: 565
- Source Port: 1161
- Destination Port: 80
- [Stream index: 0]
- [Stream Packet Number: 4]
- [Conversation completeness: Incomplete, DATA (15)]
- [TCP Segment Len: 565]
- Sequence Number: 1 (relative sequence number)
- Sequence Number (raw): 232129013
- [Next Sequence Number: 366 (relative sequence number)]
- Acknowledgment Number: 1 (relative ack number)
- Acknowledgment number (raw): 883061786
- 0101 = Header Length: 20 bytes (5)
- > Flags: 0x010 (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
-0... = Push: Set
-0... = Reset: Not set
-0... = Syn: Not set
-0... = Fin: Not set
- [TCP Flags:AP....]
- Window: 17520
- [Calculated window size: 17520]
- [Window size scaling factor: -2 (no window scaling used)]
- Checksum: 0x7671 [unverified]

The 'Packet Bytes' pane on the right shows the raw data of the packet, which is a hex dump of the HTTP POST command. The first few bytes of the data field are 'POST / HTTP/1.1', indicating the start of the command.

如上图，sequence number 为 1.

6. 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 segments

i. segment 1

```
4 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
Frame 4: 619 bytes on wire (4952 bits), 619 bytes captured (4952 bits)
Encapsulation type: Ethernet (1)
Arrival Time: Aug 21, 2004 21:44:20.596858000 CST
UTC Arrival Time: Aug 21, 2004 13:44:20.596858000 UTC
Epoch Arrival Time: 1093095860.596858000
[Time shift for this packet: 0.000000000 seconds]
[Time delta from previous captured frame: 0.003212000 seconds]
[Time delta from previous displayed frame: 0.003212000 seconds]
[Time since reference or first frame: 0.026477000 seconds]
Frame Number: 4
Frame Length: 619 bytes (4952 bits)
Capture Length: 619 bytes (4952 bits)
[Frame is marked: False]
[Frame is ignored: False]
[Protocols in frame: eth:ethertype:ip:tcp:data]
[Coloring Rule Name: HTTP]
[Coloring Rule String: http || tcp.port == 80 || http2]
Ethernet II, Src: ActiontecEle_8a:70:1a (00:20:e0:8a:70:1a), Dst: LinksysGroup_da:a:f:73 (00:06:25:da:a:f:73)
Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.119.245.12
Transmission Control Protocol, Src Port: 1161, Dst Port: 80, Seq: 1, Ack: 1, Len: 565
Data (565 bytes)
```

sequence number: 1

到达时间: Aug 21, 2004 21:44:20.596858000 CST

```
▼ [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 到达时间: Aug 21, 2004 21:44:20.624318000 CST

RTT: 0.027460000

EstimatedRTT = (1 - a) × EstimatedRTT + a × SampleRTT, a 取0.125, 初始值取第一次采样的RTT, 则

EstimatedRTT = RTT = 0.027460000s

ii. segment 2

```
5 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
Frame 5: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits)
Encapsulation type: Ethernet (1)
Arrival Time: Aug 21, 2004 21:44:20.612118000 CST
UTC Arrival Time: Aug 21, 2004 13:44:20.612118000 UTC
Epoch Arrival Time: 1093095860.612118000
[Time shift for this packet: 0.000000000 seconds]
[Time delta from previous captured frame: 0.015260000 seconds]
[Time delta from previous displayed frame: 0.015260000 seconds]
[Time since reference or first frame: 0.041737000 seconds]
Frame Number: 5
Frame Length: 1514 bytes (12112 bits)
Capture Length: 1514 bytes (12112 bits)
[Frame is marked: False]
[Frame is ignored: False]
[Protocols in frame: eth:ethertype:ip:tcp:data]
[Coloring Rule Name: HTTP]
[Coloring Rule String: http || tcp.port == 80 || http2]
```

sequence number: 566

到达时间: Aug 21, 2004 21:44:20.612118000 CST

[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 到达时间:Aug 21, 2004 21:44:20.647675000 CST

RTT: 0.035557000

$\text{EstimatedRTT} = (1 - a) \times \text{EstimatedRTT} + a \times \text{SampleRTT} = 0.875 \times 0.027460000 + 0.125 \times 0.035557000 = 0.028472125\text{s}$

iii. segment 3

```
7 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
Frame 7: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits)
Encapsulation type: Ethernet (1)
Arrival Time: Aug 21, 2004 21:44:20.624407000 CST
UTC Arrival Time: Aug 21, 2004 13:44:20.624407000 UTC
Epoch Arrival Time: 1093095860.624407000
[Time shift for this packet: 0.000000000 seconds]
[Time delta from previous captured frame: 0.000089000 seconds]
[Time delta from previous displayed frame: 0.000089000 seconds]
[Time since reference or first frame: 0.054026000 seconds]
Frame Number: 7
Frame Length: 1514 bytes (12112 bits)
Capture Length: 1514 bytes (12112 bits)
[Frame is marked: False]
[Frame is ignored: False]
[Protocols in frame: eth:ethertype:ip:tcp:data]
[Coloring Rule Name: HTTP]
[Coloring Rule String: http || tcp.port == 80 || http2]
```

sequence number: 2026

到达时间:Aug 21, 2004 21:44:20.624407000 CST

[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 到达时间:Aug 21, 2004 21:44:20.694466000 CST

RTT:0.070059000 seconds

$\text{EstimatedRTT} = 0.875 \times 0.028472125 + 0.125 \times 0.070059000 = 0.033670484 \text{ seconds}$

iv. segment 4

```
8 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
Frame 8: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits)
Encapsulation type: Ethernet (1)
Arrival Time: Aug 21, 2004 21:44:20.625071000 CST
UTC Arrival Time: Aug 21, 2004 13:44:20.625071000 UTC
Epoch Arrival Time: 1093095860.625071000
[Time shift for this packet: 0.000000000 seconds]
[Time delta from previous captured frame: 0.000664000 seconds]
[Time delta from previous displayed frame: 0.000664000 seconds]
[Time since reference or first frame: 0.054690000 seconds]
Frame Number: 8
Frame Length: 1514 bytes (12112 bits)
Capture Length: 1514 bytes (12112 bits)
[Frame is marked: False]
[Frame is ignored: False]
[Protocols in frame: eth:ethertype:ip:tcp:data]
[Coloring Rule Name: HTTP]
[Coloring Rule String: http || tcp.port == 80 || http2]
```

sequence number: 3486

到达时间: Aug 21, 2004 21:44:20.625071000 CST

[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 到达时间: Aug 21, 2004 21:44:20.739499000 CST

RTT: 0.114428000 seconds

EstimatedRTT = $0.875 \times 0.033670484 + 0.125 \times 0.114428000 = 0.043765173$ seconds

v. segment 5

```
10 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
Frame 10: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits)
Encapsulation type: Ethernet (1)
Arrival Time: Aug 21, 2004 21:44:20.647786000 CST
UTC Arrival Time: Aug 21, 2004 13:44:20.647786000 UTC
Epoch Arrival Time: 1093095860.647786000
[Time shift for this packet: 0.000000000 seconds]
[Time delta from previous captured frame: 0.000111000 seconds]
[Time delta from previous displayed frame: 0.000111000 seconds]
[Time since reference or first frame: 0.077405000 seconds]
Frame Number: 10
Frame Length: 1514 bytes (12112 bits)
Capture Length: 1514 bytes (12112 bits)
[Frame is marked: False]
[Frame is ignored: False]
[Protocols in frame: eth:ethertype:ip:tcp:data]
[Coloring Rule Name: HTTP]
[Coloring Rule String: http || tcp.port == 80 || http2]
```

sequence number: 4946

到达时间: Aug 21, 2004 21:44:20.647786000 CST

[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 到达时间: Aug 21, 2004 21:44:20.787680000 CST

RTT: 0.139894000 seconds

EstimatedRTT = $0.875 \times 0.043765173 + 0.125 \times 0.139894000 = 0.055781277$ seconds

vi. segment 6

```

11 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
Frame 11: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits)
Encapsulation type: Ethernet (1)
Arrival Time: Aug 21, 2004 21:44:20.648538000 CST
UTC Arrival Time: Aug 21, 2004 13:44:20.648538000 UTC
Epoch Arrival Time: 1093095860.648538000
[Time shift for this packet: 0.000000000 seconds]
[Time delta from previous captured frame: 0.000752000 seconds]
[Time delta from previous displayed frame: 0.000752000 seconds]
[Time since reference or first frame: 0.078157000 seconds]
Frame Number: 11
Frame Length: 1514 bytes (12112 bits)
Capture Length: 1514 bytes (12112 bits)
[Frame is marked: False]
[Frame is ignored: False]
[Protocols in frame: eth:ethertype:ip:tcp:data]
[Coloring Rule Name: HTTP]
[Coloring Rule String: http || tcp.port == 80 || http2]

```

sequence number: 6406

到达时间: Aug 21, 2004 21:44:20.648538000 CST

[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]

ACK 到达时间: Aug 21, 2004 21:44:20.838183000 CST

RTT:0.189645000 seconds

EstimatedRTT = $0.875 \times 0.055781277 + 0.125 \times 0.189645000 = 0.072514242$ seconds

7. What is the length of each of the first six TCP segments?

如上面图片所示，前 6 个 segments 的长度依次为: 565, 1460, 1460, 1460, 1460, 1460

8. 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?

```

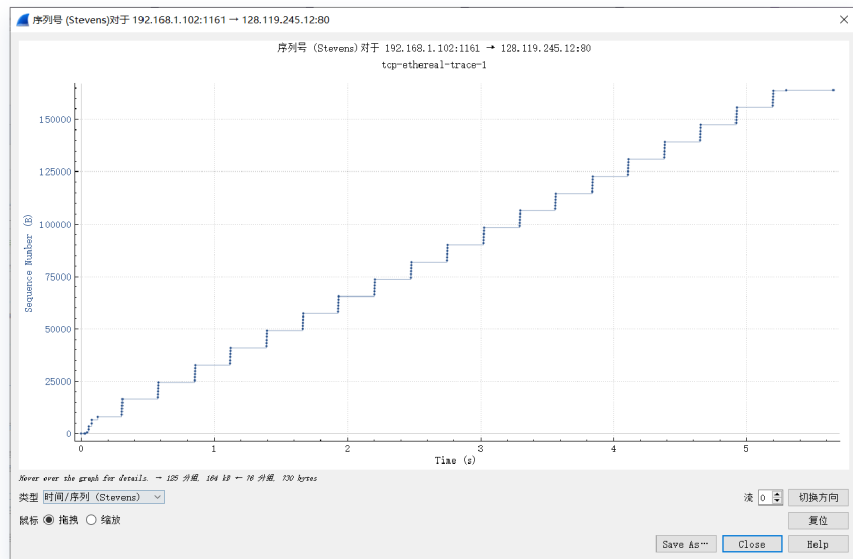
2 21:44:20.593553 128.119.245.12 192.168.1.102 TCP 62 80 → 1161 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM

```

最小缓冲区大小如上图所示为5840；由于 TCP 的流量控制服务，缺少接收器缓冲区空间会限制发送方传送 TCP 区段，能够消除发送方使接收方缓存溢出的可能性，使得发送方的发送速率与接收方应用程序的读取速率相匹配。

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

检查数据包的时间序列如下图:



如上图，segments 的序列呈递增的趋势，因此不存在重传的情况。

10. 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).

ACK 中的数据量取决于窗口大小 cwnd，接收方会从 ACK 中识别确认号。

通过查看连续的 ACK segment 的 traces，可以检查出其确认的 segment 是否是连续的或是每隔一个 segment 再确认的。

11. What is the throughput (bytes transferred per unit time) for the TCP connection?

查看 POST 请求中运输的数据量:

```
199 21:44:25.867722 192.168.1.102 128.119.245.12 HTTP 104 POST /ethereal-labs/lab3-1-reply.htm HTTP/1.1 (text/plain)
122 Reassembled TCP Segments (164090 bytes): #4(565), #5(1460), #7(1460), #8(1460), #10(1460), #11(1460), #13(1147), #18(1460), #19(1460), #20(1460), #21(1460), #22(1460), #23(892), #30(1460)
[Frame: 4, payload: 0-564 (565 bytes)]
[Frame: 5, payload: 565-2024 (1460 bytes)]
[Frame: 7, payload: 2025-3484 (1460 bytes)]
[Frame: 8, payload: 3485-4944 (1460 bytes)]
[Frame: 10, payload: 4945-6404 (1460 bytes)]
[Frame: 11, payload: 6405-7864 (1460 bytes)]
[Frame: 13, payload: 7865-9011 (1147 bytes)]
[Frame: 18, payload: 9012-10471 (1460 bytes)]
[Frame: 19, payload: 10472-11931 (1460 bytes)]
[Frame: 20, payload: 11932-13391 (1460 bytes)]
[Frame: 21, payload: 13392-14851 (1460 bytes)]
[Frame: 22, payload: 14852-16311 (1460 bytes)]
[Frame: 23, payload: 16312-17203 (892 bytes)]
[Frame: 30, payload: 17204-18663 (1460 bytes)]
```

如上图，为 164090 bytes。

在查看接收方的接收时间:


```

199 21:44:25.867722 192.168.1.102 128.119.245.12 HTTP 104 POST /ethereal-labs/lab3-1-rep1
760 21:44:25.868853 192.168.1.102 128.119.245.12 TCP 80 80 → 1161 1161 Seq=164090
> [Conversation completeness: Incomplete, DATA (15)]
[TCP Segment Len: 50]
Sequence Number: 164041 (relative sequence number)
Sequence Number (raw): 232293053
[Next Sequence Number: 164091 (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
  [TCP Flags: .....AP...]
Window: 17520
[Calculated window size: 17520]
[Window size scaling factor: -2 (no window scaling used)]
Checksum: 0x9f0f [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
v [Timestamps]
  [Time since first frame in this TCP stream: 5.297341000 seconds]
  [Time since previous frame in this TCP stream: 0.000084000 seconds]

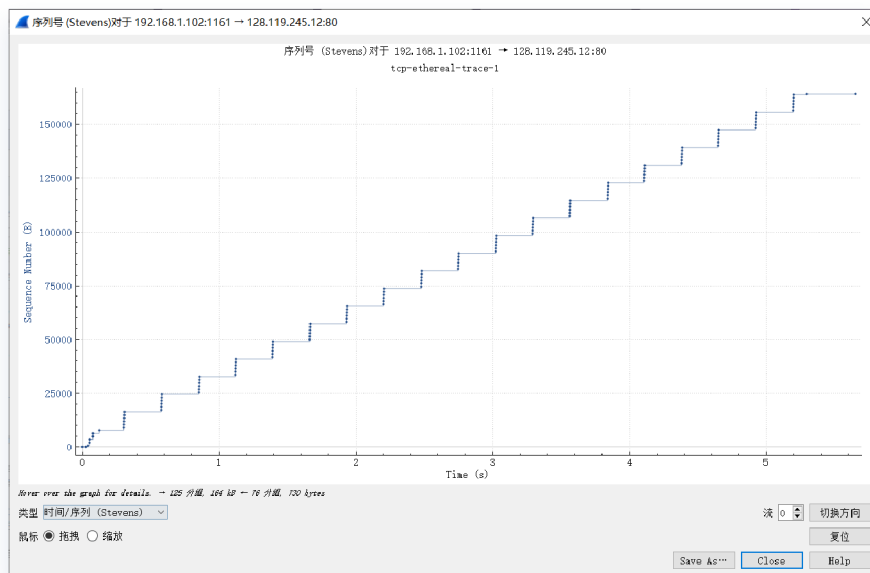
```

如上图 为 5.297341000 seconds

则平均吞吐量为: 吞吐量 = $164090 \div 5.297341000 = 30975.9179181$ bytes/s

TCP congestion control in action

打开 Wireshark 的时序分析, 如下图:

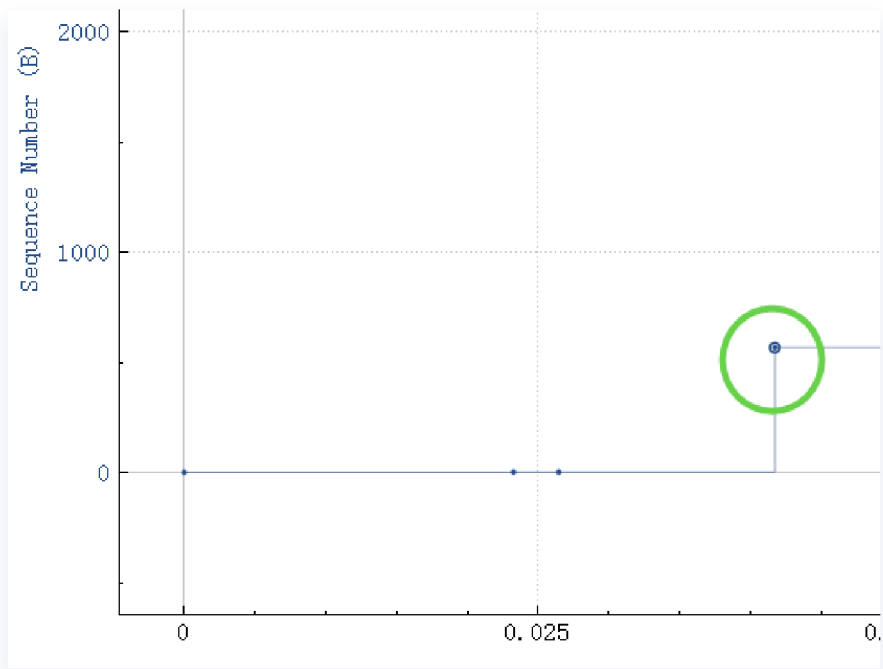


问题

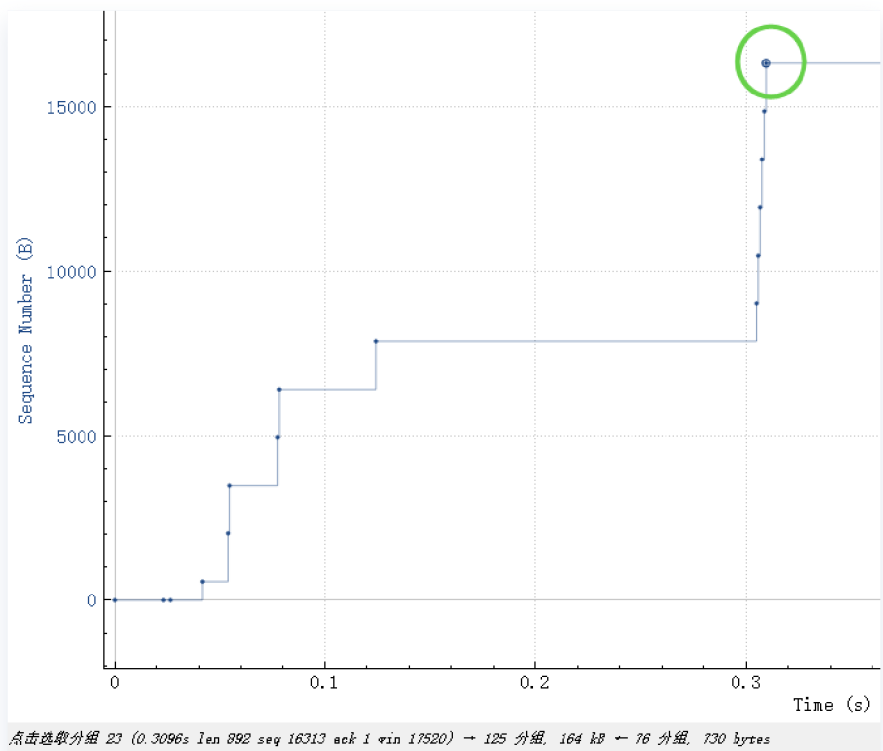
12. Use the Time-Sequence-Graph(Stevens) plotting tool to view the sequence number versus time plot of segments being sent from the client to the gaia.cs.umass.edu server. Can you identify where TCP's slowstart phase begins

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.

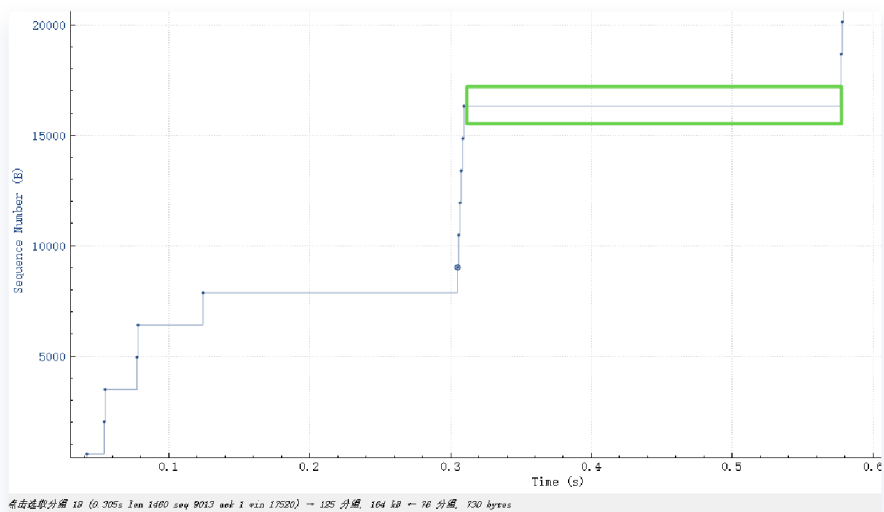
慢启动阶段开始时，TCP发送速率较慢，但呈指数型增长，如下图，当第一个 TCP 区段刚刚发送出去时，此时即为慢启动刚开始时：



在分组 23 处, 这样的指数型增长的速率突然停滞，并且这个分组之后，TCP 发送速率增长呈线性性（即每一个区段发送后增长量相同）说明这个时候发生了拥塞，进入拥塞避免阶段。



如下图，随之而来的区段即为 congestion-avoidance 开始阶段:



13. Comment on ways in which the measured data differs from the idealized behavior of TCP that we've studied in the text.

慢启动并非永远是最高效的。在本例中，刚开始发送的文件大小非常小，由于慢启动阶段开始时发送速率很小，故发送这个文件需要较长时间，而如果不采用慢启动，将一开始的窗口大小设置的很大，则只需要很段时间即可发送完，并且不会引起拥塞