

# 计算机网络实验报告\_TCP实验

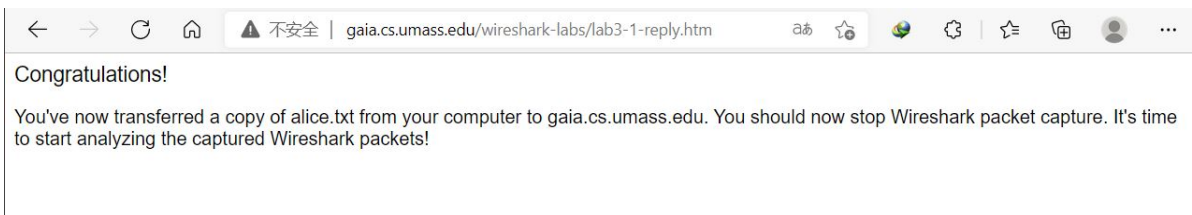
姓名:李蔚林 学号:PB20000219

## 实验目的

1. 了解TCP的SEQ和ACK序列号及确认号的作用.
2. 了解TCP建立连接的三次握手过程
3. 了解TCP的拥塞控制算法和报文段结构
4. 对TCP连接性能进行计算

## 实验过程及题目

下载Alice in Wonderland文件,然后上传到目标网页



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? To answer this question, it's probably easiest to select an HTTP message and explore the details of the TCP packet used to carry this HTTP message, using the "details of the selected packet header window" (refer to Figure 2 in the "Getting Started with Wireshark" Lab if you're uncertain about the Wireshark windows.

184	21:44:25.494244	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80
185	21:44:25.495048	192.168.1.102	128.119.245.12	TCP	946	1161 → 80
186	21:44:25.589570	128.119.245.12	192.168.1.102	TCP	60	80 → 1161
190	21:44:25.695400	128.119.245.12	192.168.1.102	TCP	60	80 → 1161
191	21:44:25.767667	128.119.245.12	192.168.1.102	TCP	60	80 → 1161
192	21:44:25.767889	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80
193	21:44:25.768769	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80

>	Frame 185: 946 bytes on wire (7568 bits), 946 bytes captured (7568 bits)
>	Ethernet II, Src: Actionte_8a:70:1a (00:20:e0:8a:70:1a), Dst: LinksysG_da:af:73 (00:06:25:d
>	Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.119.245.12
▼	Transmission Control Protocol, Src Port: 1161, Dst Port: 80, Seq: 155577, Ack: 1, Len: 892
	Source Port: 1161
	Destination Port: 80
	[Stream index: 0]

source IP:192.168.1.102

source port number: 1161

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?

IP:128.119.245.12

port number:80

184	21:44:25.494244	192.168.1.102	128.119.245.12	TCP	1514	1161 → 8
185	21:44:25.495048	192.168.1.102	128.119.245.12	TCP	946	1161 → 8
186	21:44:25.589570	128.119.245.12	192.168.1.102	TCP	60	80 → 116
190	21:44:25.695400	128.119.245.12	192.168.1.102	TCP	60	80 → 116
191	21:44:25.767667	128.119.245.12	192.168.1.102	TCP	60	80 → 116
192	21:44:25.767889	192.168.1.102	128.119.245.12	TCP	1514	1161 → 8
193	21:44:25.768769	192.168.1.102	128.119.245.12	TCP	1514	1161 → 8
194	21:44:25.769656	192.168.1.102	128.119.245.12	TCP	1514	1161 → 8

Frame 185: 946 bytes on wire (7568 bits), 946 bytes captured (7568 bits)  
 Ethernet II, Src: Actionte\_8a:70:1a (00:20:e0:8a:70:1a), Dst: LinksysG\_da:af:73 (00:06:  
 Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.119.245.12  
 Transmission Control Protocol, Src Port: 1161, Dst Port: 80, Seq: 155577, Ack: 1, Len:  
 Source Port: 1161  
 Destination Port: 80  
 [Stream index: 0]

3.What is the IP address and TCP port number used by your client computer (source) to transfer the file to gaia.cs.umass.edu?

由于没有自己抓包,本题目不做.

**4.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?**

Source	Destination	Protocol	Length	Info
570381 192.168.1.102	128.119.245.12	TCP	62	1161 → 80 [SYN] Seq=0 Win=16
593553 128.119.245.12	192.168.1.102	TCP	62	80 → 1161 [SYN, ACK] Seq=0 A
593646 192.168.1.102	128.119.245.12	TCP	54	1161 → 80 [ACK] Seq=1 Ack=1
596858 192.168.1.102	128.119.245.12	TCP	619	1161 → 80 [PSH, ACK] Seq=1 A
612118 192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [PSH, ACK] Seq=566
624318 128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=56
624407 192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=2026 Ack
625071 192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=3486 Ack
647675 128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=20
647786 192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=4946 Ack
648538 192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=6406 Ack
694466 128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=34

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

.如图:Seq=0,根据TCP协议建立连接时的三次握手,主机会首先发送SYN来请求建立连接(三次握手的第一步).

5.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?



533	128.119.245.12	192.168.1.102	TCP	62	80 → 1161	[SYN, ACK]	Seq=0 Ack=
546	192.168.1.102	128.119.245.12	TCP	54	1161 → 80	[ACK]	Seq=1 Ack=1 Win=
558	192.168.1.102	128.119.245.12	TCP	619	1161 → 80	[PSH, ACK]	Seq=1 Ack=
568	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80	[PSH, ACK]	Seq=566 Ack=
578	128.119.245.12	192.168.1.102	TCP	60	80 → 1161	[ACK]	Seq=1 Ack=566 Win=
587	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80	[ACK]	Seq=2026 Ack=1 Win=
597	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80	[ACK]	Seq=3486 Ack=1 Win=
607	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80	[ACK]	Seq=3486 Ack=1 Win=
617	128.119.245.12	192.168.1.102	TCP	60	80 → 1161	[ACK]	Seq=1 Ack=2026 Win=
626	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80	[ACK]	Seq=4946 Ack=1 Win=
636	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80	[ACK]	Seq=6406 Ack=1 Win=
646	128.119.245.12	192.168.1.102	TCP	60	80 → 1161	[ACK]	Seq=1 Ack=3486 Win=

< >

> 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]

[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)

Seq=0, Acknowledgement Number的值为1,该值是由gaia.cs.umass.edu确定,通过将来自客户端ACK字段加1得到.用于标识服务器成功接收了发送的请求.

6.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.

tcp						
	Source	Destination	Protocol	Length	Info	
70381	192.168.1.102	128.119.245.12	TCP	62	1161 → 80 [SYN] Seq=0 Win=16	
93553	128.119.245.12	192.168.1.102	TCP	62	80 → 1161 [SYN, ACK] Seq=0 Ack=	
93646	192.168.1.102	128.119.245.12	TCP	54	1161 → 80 [ACK] Seq=1 Ack=1 Win=	
96858	192.168.1.102	128.119.245.12	TCP	619	1161 → 80 [PSH, ACK] Seq=1 Ack=	
12118	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [PSH, ACK] Seq=566 Ack=	
24318	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=56	
24407	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=2026 Ack=	
25071	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=3486 Ack=	
47675	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=20	
47786	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=4946 Ack=	
48538	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=6406 Ack=	
94466	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=34	

< >

Source Port: 1161

Destination Port: 80

[Stream index: 0]

[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)

> Flags: 0x018 (PSH, ACK)

< >

0020	f5 0c 04 89 00 50 0d d6	01 f5 34 a2 74 1a 50 18	.....P...4.t.P.
0030	44 70 1f bd 00 00 50 4f	53 54 20 2f 65 74 68 65	Dp...PO ST /ethe
0040	72 65 61 6c 2d 6c 61 62	73 2f 6c 61 62 33 2d 31	real-lab s/lab3-1

Stream index (tcp stream)      分组: 213      已显示: 202 / 94 9%      配置: Defa

Seq=1

7. 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.

第一个区段:

4	0.026477	192.168.1.102	128.119.245.12	TCP	619 1161 → 80 [PSH, ACK]	Seq=1 Ack=1
5	0.041737	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [PSH, ACK]	Seq=566 Ack=1 Win=1
6	0.053937	128.119.245.12	192.168.1.102	TCP	60 80 → 1161 [ACK]	Seq=1 Ack=566 Win=6780
7	0.054026	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [ACK]	Seq=2026 Ack=1 Win=17520
8	0.054690	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [ACK]	Seq=3486 Ack=1 Win=17520
9	0.077294	128.119.245.12	192.168.1.102	TCP	60 80 → 1161 [ACK]	Seq=1 Ack=2026 Win=8760
10	0.077405	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [ACK]	Seq=4946 Ack=1 Win=17520
11	0.078157	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [ACK]	Seq=6406 Ack=1 Win=17520
12	0.124085	128.119.245.12	192.168.1.102	TCP	60 80 → 1161 [ACK]	Seq=1 Ack=3486 Win=11680

[Calculated window size: 6780]  
[Window size scaling factor: -2 (no window scaling used)]  
Checksum: 0x9e30 [unverified]  
[Checksum Status: Unverified]  
Urgent Pointer: 0

✓ [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]

> [Timestamps]  
> VSS Monitoring Ethernet trailer, Source Port: 55826

第二个区段:

5	0.041737	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [PSH, ACK]	Seq=566 Ack=1 Win=1
6	0.053937	128.119.245.12	192.168.1.102	TCP	60 80 → 1161 [ACK]	Seq=1 Ack=566 Win=6780
7	0.054026	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [ACK]	Seq=2026 Ack=1 Win=17520
8	0.054690	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [ACK]	Seq=3486 Ack=1 Win=17520
9	0.077294	128.119.245.12	192.168.1.102	TCP	60 80 → 1161 [ACK]	Seq=1 Ack=2026 Win=8760
10	0.077405	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [ACK]	Seq=4946 Ack=1 Win=17520
11	0.078157	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [ACK]	Seq=6406 Ack=1 Win=17520
12	0.124085	128.119.245.12	192.168.1.102	TCP	60 80 → 1161 [ACK]	Seq=1 Ack=3486 Win=11680

[Calculated window size: 8760]  
[Window size scaling factor: -2 (no window scaling used)]  
Checksum: 0x90c0 [unverified]  
[Checksum Status: Unverified]  
Urgent Pointer: 0

✓ [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]

> [Timestamps]  
> VSS Monitoring Ethernet trailer, Source Port: 34718

第三个区段:

7	0.054026	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [ACK]	Seq=2026 Ack=1 Win=17520
---	----------	---------------	----------------	-----	----------------------	--------------------------



7	0.054026	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [ACK]	Seq=2026 Ack=1 Win=17520
8	0.054690	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [ACK]	Seq=3486 Ack=1 Win=17520
9	0.077294	128.119.245.12	192.168.1.102	TCP	60 80 → 1161 [ACK]	Seq=1 Ack=2026 Win=8760
10	0.077405	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [ACK]	Seq=4946 Ack=1 Win=17520
11	0.078157	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [ACK]	Seq=6406 Ack=1 Win=17520
12	0.124085	128.119.245.12	192.168.1.102	TCP	60 80 → 1161 [ACK]	Seq=1 Ack=3486 Win=11680
13	0.124185	192.168.1.102	128.119.245.12	TCP	1201 1161 → 80 [PSH, ACK]	Seq=7866 Ack=1 Win=
14	0.169118	128.119.245.12	192.168.1.102	TCP	60 80 → 1161 [ACK]	Seq=1 Ack=4946 Win=14600
15	0.217299	128.119.245.12	192.168.1.102	TCP	60 80 → 1161 [ACK]	Seq=1 Ack=6406 Win=17520
16	0.267802	128.119.245.12	192.168.1.102	TCP	60 80 → 1161 [ACK]	Seq=1 Ack=7866 Win=20440
17	0.304807	128.119.245.12	192.168.1.102	TCP	60 80 → 1161 [ACK]	Seq=1 Ack=9013 Win=23360
18	0.305040	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [ACK]	Seq=9013 Ack=1 Win=17520

Window: 11680  
[Calculated window size: 11680]  
[Window size scaling factor: -2 (no window scaling used)]  
Checksum: 0x7fa4 [unverified]  
[Checksum Status: Unverified]  
Urgent Pointer: 0  
▼ [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]  
> [Timestamps]

第四个区段:

8	0.054690	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [ACK]	Seq=3486 Ack=1 Win=17520
13	0.124185	192.168.1.102	128.119.245.12	TCP	1201 1161 → 80 [PSH, ACK]	Seq=7866 Ack=1 Win=17520
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=1
19	0.305813	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [ACK]	Seq=10473 Ack=1 Win=17520 Len=
20	0.306692	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [ACK]	Seq=11933 Ack=1 Win=17520 Len=
21	0.307571	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [ACK]	Seq=13393 Ack=1 Win=17520 Len=
22	0.308699	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [ACK]	Seq=14853 Ack=1 Win=17520 Len=
23	0.309573	192.168.1.102	128.119.245.12	TCP	046 1161 → 80 [PSH, ACK]	Seq=16313 Ack=1 Win=17520

Window: 14600  
[Calculated window size: 14600]  
[Window size scaling factor: -2 (no window scaling used)]  
Checksum: 0x6e88 [unverified]  
[Checksum Status: Unverified]  
Urgent Pointer: 0  
▼ [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]  
> [Timestamps]  
> VSS Monitoring Ethernet trailer, Source Port: 54332

第五个区段:

10	0.077405	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [ACK]	Seq=4946 Ack=1 Win=17520
10	0.077405	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [ACK]	Seq=4946 Ack=1 Win=17520
11	0.078157	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [ACK]	Seq=6406 Ack=1 Win=17520
12	0.124085	128.119.245.12	192.168.1.102	TCP	60 80 → 1161 [ACK]	Seq=1 Ack=3486 Win=11680
13	0.124185	192.168.1.102	128.119.245.12	TCP	1201 1161 → 80 [PSH, ACK]	Seq=7866 Ack=1 Win=
14	0.169118	128.119.245.12	192.168.1.102	TCP	60 80 → 1161 [ACK]	Seq=1 Ack=4946 Win=14600
15	0.217299	128.119.245.12	192.168.1.102	TCP	60 80 → 1161 [ACK]	Seq=1 Ack=6406 Win=17520
16	0.267802	128.119.245.12	192.168.1.102	TCP	60 80 → 1161 [ACK]	Seq=1 Ack=7866 Win=20440
17	0.304807	128.119.245.12	192.168.1.102	TCP	60 80 → 1161 [ACK]	Seq=1 Ack=9013 Win=23360
18	0.305040	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80 [ACK]	Seq=9013 Ack=1 Win=17520

Window: 17520  
[Calculated window size: 17520]  
[Window size scaling factor: -2 (no window scaling used)]  
Checksum: 0x5d6c [unverified]  
[Checksum Status: Unverified]  
Urgent Pointer: 0  
▼ [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]  
> [Timestamps]

第六个区段:

11	0.078157	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80	[ACK] Seq=6406 Ack=1 Win=17520
12	0.124085	128.119.245.12	192.168.1.102	TCP	60 80 → 1161	[ACK] Seq=1 Ack=3486 Win=11680
13	0.124185	192.168.1.102	128.119.245.12	TCP	1201 1161 → 80	[PSH, ACK] Seq=7866 Ack=1 Win=
14	0.169118	128.119.245.12	192.168.1.102	TCP	60 80 → 1161	[ACK] Seq=1 Ack=4946 Win=14600
15	0.217299	128.119.245.12	192.168.1.102	TCP	60 80 → 1161	[ACK] Seq=1 Ack=6406 Win=17520
16	0.267802	128.119.245.12	192.168.1.102	TCP	60 80 → 1161	[ACK] Seq=1 Ack=7866 Win=20440
17	0.304807	128.119.245.12	192.168.1.102	TCP	60 80 → 1161	[ACK] Seq=1 Ack=9013 Win=23360
18	0.305040	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80	[ACK] Seq=9013 Ack=1 Win=17520
19	0.305813	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80	[ACK] Seq=10473 Ack=1 Win=1752
20	0.306692	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80	[ACK] Seq=11933 Ack=1 Win=1752
21	0.307571	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80	[ACK] Seq=13393 Ack=1 Win=1752
22	0.308699	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80	[ACK] Seq=14853 Ack=1 Win=1752
23	0.309552	192.168.1.102	128.119.245.12	TCP	1514 1161 → 80	[ACK] Seq=16313 Ack=1 Win=1752

Window: 20440  
[Calculated window size: 20440]  
[Window size scaling factor: -2 (no window scaling used)]  
Checksum: 0x4c50 [unverified]  
[Checksum Status: Unverified]  
Urgent Pointer: 0  
[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]  
[Timestamps]

问题答案如下表:(Send Time 和ACK Receive Time都是自捕捉开始的时间):

Estimated RTT用如下公式计算:(其中 $\alpha$ 这里取0.125, 保留六位小数)

$$EstimatedRTT = (1 - \alpha) * EstimatedRTT + \alpha * SampleRTT$$

Segment	Seq	Send Time	ACK Receive Time	RTT	Estimated RTT
Segment 1	1	0.026477	0.053937	0.027460	0.027460
Segment 2	566	0.041737	0.077294	0.035557	0.028472
Segment 3	2026	0.054026	0.124085	0.070059	0.033670
Segment 4	3486	0.054690	0.169118	0.114428	0.043764
Segment 5	4946	0.077405	0.217299	0.139894	0.055780
Segment 6	6406	0.078157	0.267802	0.189645	0.072513

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

由前一问表格的Seq可以计算得到:

$$Segment_1.len() = 566 - 1 = 565$$

对于  $i = 2, 3, 4, 5, 6$

$$Segment_i.len() = 1460$$

9.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 0.023172 128.119.245.12 192.168.1.102 TCP 62 80 → 1161 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0

如图,接收窗口最小量=5840字节(截图为第一个来自服务器的确认)

由于TCP的流量控制,缺少接收窗口大小时会限制发送方传输TCP Segment.但是在本次抓包中,接收窗口并未降到小于发送者的数据包,因而未受到限制.

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



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=1
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=1
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
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
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
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
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
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
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
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

没有重传的Segment,如图,检查ACK,没有重复的ACK.

11.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).

由前面对Segment1-6的分析可以知道,典型的ACK为1460字节,TCP是累积确认的.

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

197	5.202024	192.168.1.102	128.119.245.12	TCP	326	1161 → 80	[PSH, ACK]	Seq=163769 Ack=1 Win=17520 Len=1460
198	5.297257	128.119.245.12	192.168.1.102	TCP	60	80 → 1161	[ACK]	Seq=1 Ack=159389 Win=627
199	5.297341	192.168.1.102	128.119.245.12	HTTP	104	POST	/ethereal-labs/lab3-1-reply.htm	HTTP/1.1 200 OK (text/html)
200	5.389471	128.119.245.12	192.168.1.102	TCP	60	80 → 1161	[ACK]	Seq=1 Ack=162309 Win=627
201	5.447887	128.119.245.12	192.168.1.102	TCP	60	80 → 1161	[ACK]	Seq=1 Ack=164041 Win=627
202	5.455830	128.119.245.12	192.168.1.102	TCP	60	80 → 1161	[ACK]	Seq=1 Ack=164091 Win=627
203	5.461175	128.119.245.12	192.168.1.102	HTTP	784	HTTP/1.1	200 OK	(text/html)
206	5.651141	192.168.1.102	128.119.245.12	TCP	54	1161 → 80	[ACK]	Seq=164091 Ack=731 Win=17520 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=1

```

[Protocols in frame: eth:ethertype:ip:tcp:http:mime_multipart:data-text-lines]
[Coloring Rule Name: HTTP]
[Coloring Rule String: http || tcp.port == 80 || http2]
> Ethernet II, Src: Actionte_8a:70:1a (00:20:e0:8a:70:1a), Dst: LinksysG_da:af:73 (00:06:25:da:af: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: 164041, Ack: 1, Len: 50
▼ [122 Reassembled TCP Segments (164090 bytes): #4(565), #5(1460), #7(1460), #8(1460), #10(1460), #11(1460), #13(1147)]
    [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)]

```

```

[Window size scaling factor: -2 (no window scaling used)]
Checksum: 0x9f0f [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
▼ [SEQ/ACK analysis]
    [iRTT: 0.023265000 seconds]
    [Bytes in flight: 4702]
    [Bytes sent since last PSH flag: 50]
▼ [Timestamps]
    [Time since first frame in this TCP stream: 5.297341000 seconds]
    [Time since previous frame in this TCP stream: 0.000084000 seconds]
    TCP payload (50 bytes)
    TCP segment data (50 bytes)
▼ [122 Reassembled TCP Segments (164090 bytes): #4(565), #5(1460), #7(1460), #8(1460), #10(1460), #11(1460), #13(1147)]

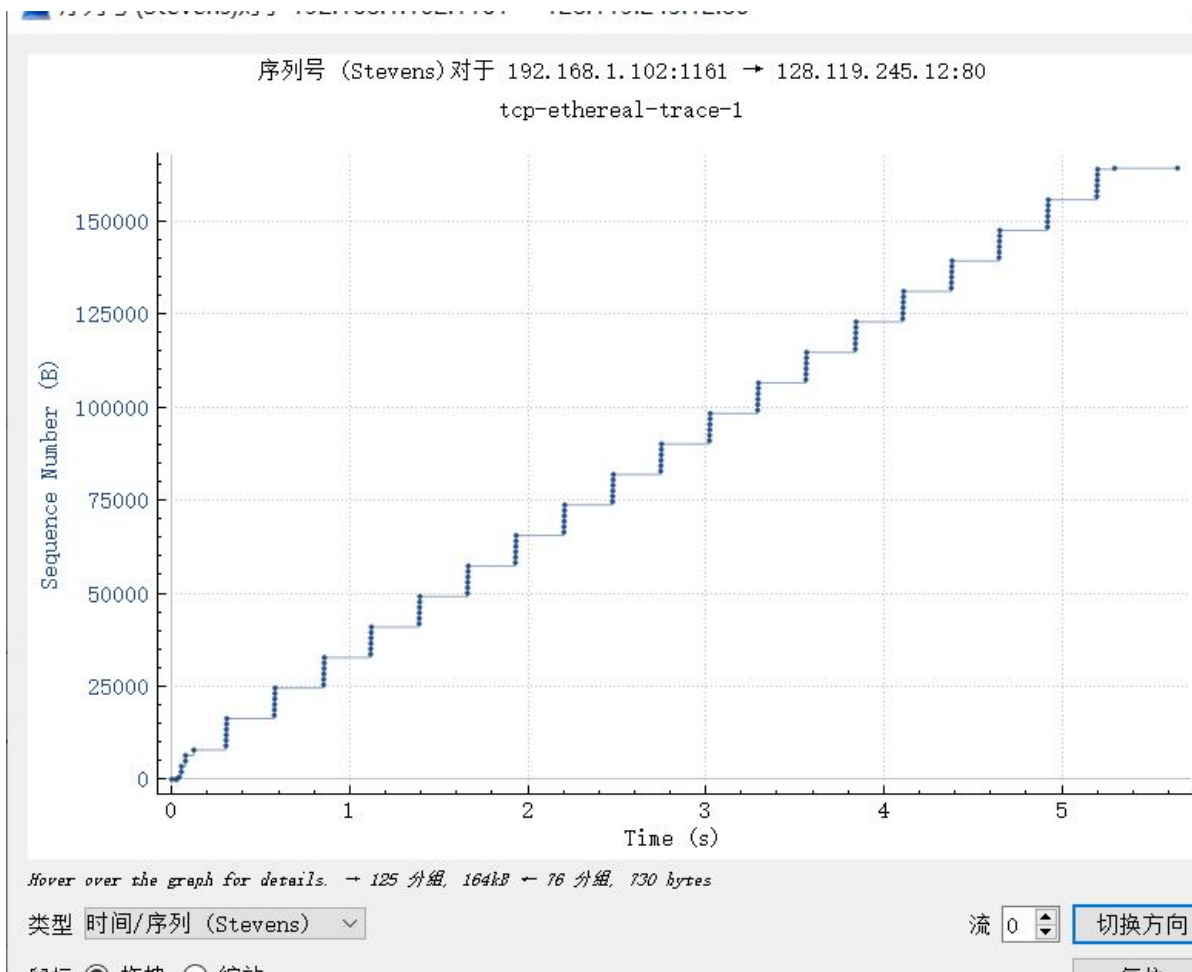
```

如图,传输数据比特数=164090 bytes

总时间=5.297341s

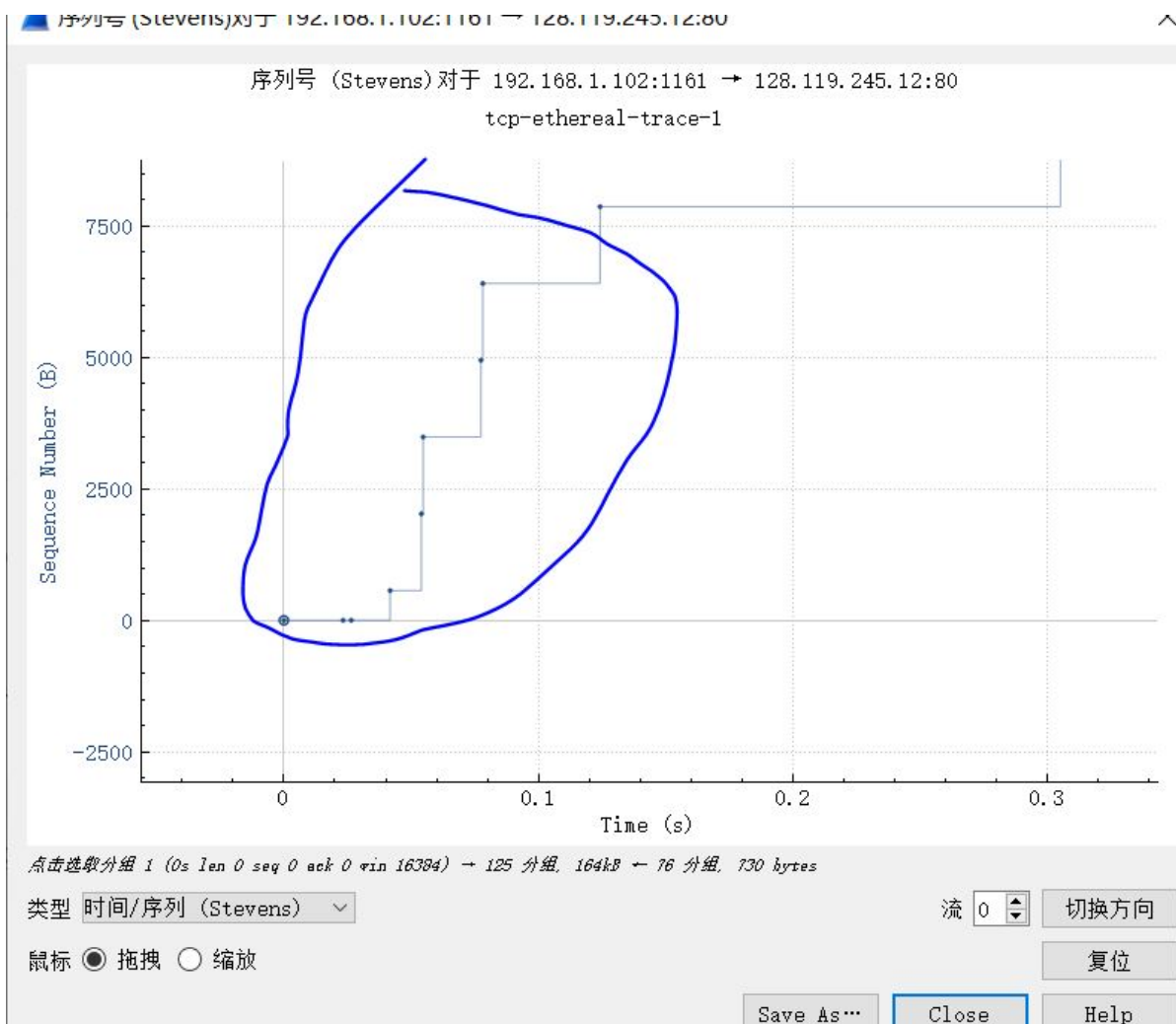
平均吞吐量=传输数据的比特数/接收方接受所有数据所需时间=30975.9179bytes/s

13. 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.

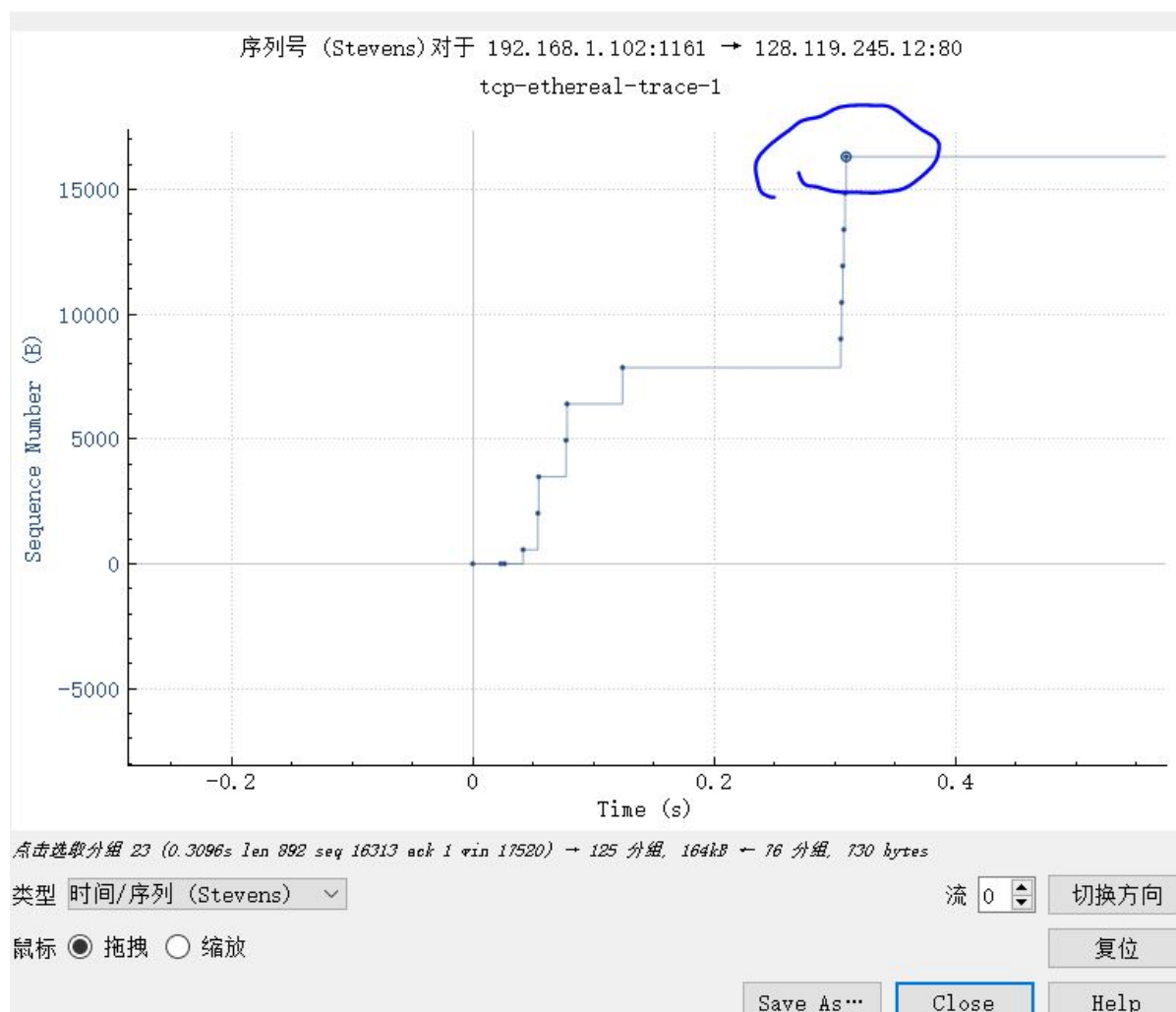


慢启动阶段开始是在第一个TCP segment发出去的时候,发送速率呈指数型增长(segment No.5)

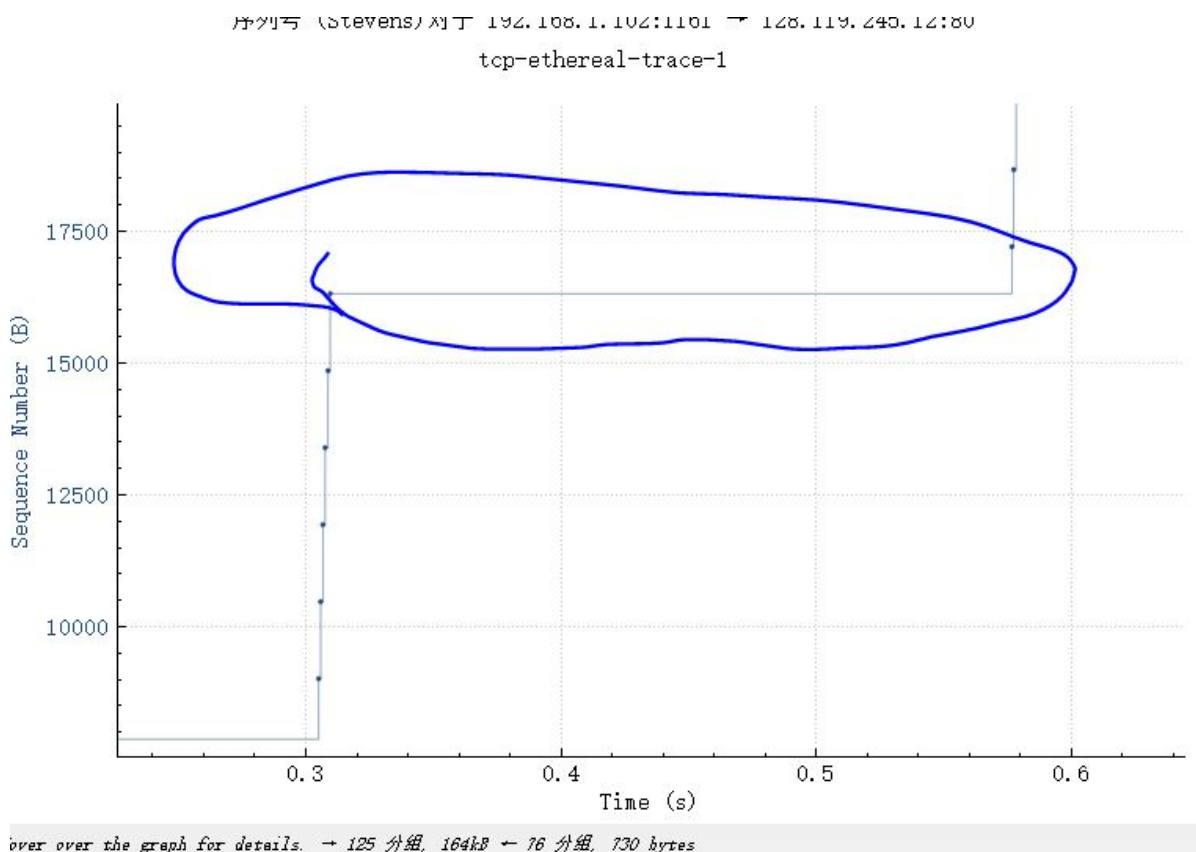




结束是在segment No.23(速率增长变慢):



避免拥塞避免区段如图:



最后分析理论和实际的不同:



理想情况下TCP的发送者会发送过多的数据,导致网络拥堵,反过来作用于发送者使其遵循AMID来避免丢包,从而大大减小发送窗口长度.实际上TCP的发送者并没有过多的数据需要传输时,上述现象不会发生,甚至可能在慢启动阶段TCP发送已经完成,.

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

由于没有自己抓包,本题目不做.

## 实验心得

---

通过本实验更深入的理解了TCP,同时明白了实践意义下的TCP工作方式,学会了使用Wireshark的时间序号图来分析结果.