

# Wireshark Lab -- TCP

Jiaying Li  
jl10919

Lab environment:

Answer: My PC uses macOS Catalina 10.15.6, shows the following setting with *ifconfig*:

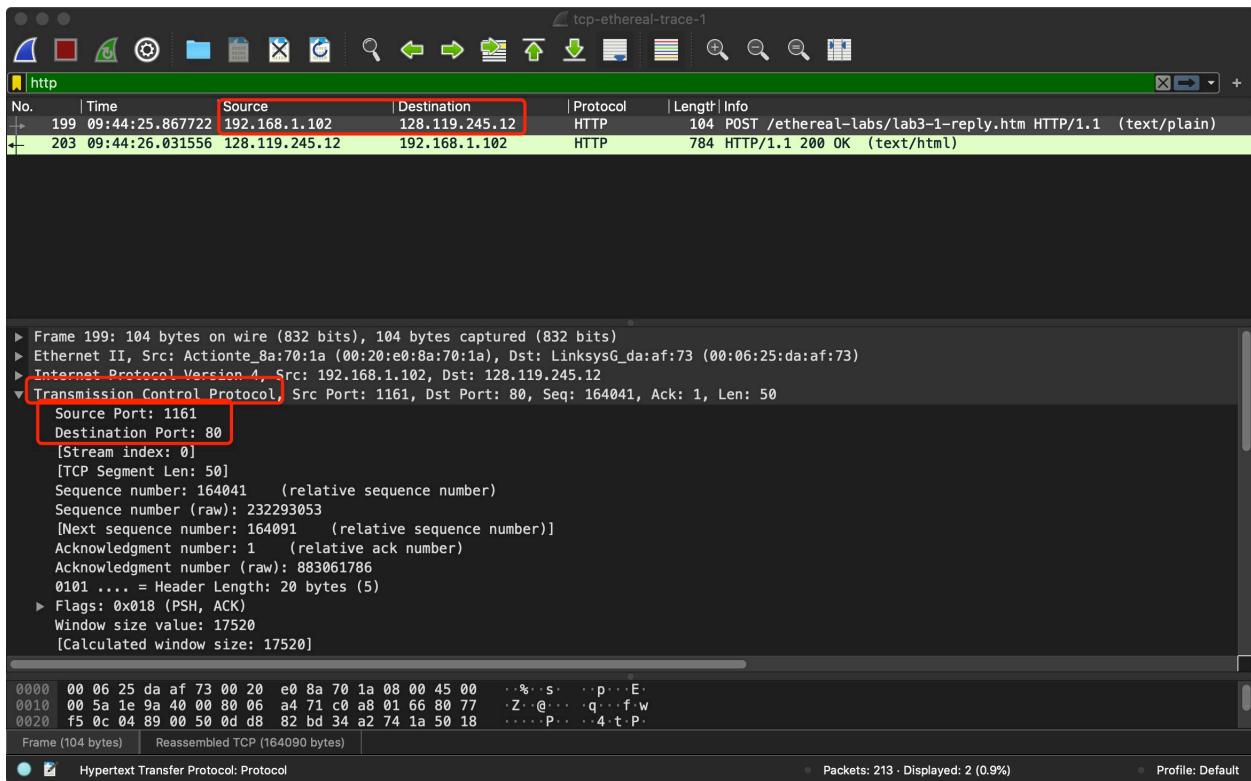


A terminal window titled "jeslee — bash" displaying the output of the "ifconfig" command. The output lists various network interfaces (lo0, gif0, stf0, en0, en1, en2, bridge0, p2p0, awdl0, llw0, utun0, utun1, utun2, utun3) with their flags, MTU, options, and configuration details. The "bridge0" interface is shown with its configuration, including port 0 and port 1 settings. The "awdl0" interface has a MAC address of e2:8b:ae:0c:af:c5.

```
(base) JesLeee@JesLeee-MBP:~ jesLeee$ ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
    options=1203<RXCSUM,TXCSUM,TXSTATUS,SW_TIMESTAMP>
    inet 127.0.0.1 netmask 0fff000000
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
            nd6 options=201<PERFORMNUD,DAD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=400<CHANNEL_IO>
    ether ac:bc:32:7b:7f:83
        inet6 fe80::1c81:1c1c%en0 prefixlen 64 secured scopeid 0x4
            inet 192.168.1.155 netmask 0xffffffff broadcast 192.168.1.255
            nd6 options=201<PERFORMNUD,DAD>
            media: autoselect
            status: active
en1: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
    options=460<TS04,TS06,CHANNEL_IO>
    ether 82:13:09:8a:17:40
        media: autoselect <full-duplex>
        status: inactive
en2: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
    options=460<TS04,TS06,CHANNEL_IO>
    ether 82:13:09:8a:17:41
        media: autoselect <full-duplex>
        status: inactive
bridge0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=63<RXCSUM,TXCSUM,TS04,TS06>
    ether 82:13:09:8a:17:40
    Configuration:
        id 0:0:0:0:0:0 priority 0 hellotime 0 fwddelay 0
        maxage 0 holdcnt 0 proto stp maxaddr 100 timeout 1200
        root id 0:0:0:0:0:0 priority 0 ifcost 0 port 0
        ipfilter disabled flags 0x0
    member: en1 flags=3<LEARNING,DISCOVER>
        ifmaxaddr 0 port 5 priority 0 path cost 0
    member: en2 flags=3<LEARNING,DISCOVER>
        ifmaxaddr 0 port 6 priority 0 path cost 0
        nd6 options=201<PERFORMNUD,DAD>
        media: <unknown type>
        status: inactive
p2p0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 2304
    options=400<CHANNEL_IO>
    ether 0e:bc:32:7b:7f:83
    media: autoselect
    status: inactive
awdl0: flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1484
    options=400<CHANNEL_IO>
    ether e2:8b:ae:0c:af:c5
    inet6 fe80::e0b:aaff:fe0c:afc5%awdl0 prefixlen 64 scopeid 0x9
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
llw0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=400<CHANNEL_IO>
    ether e2:8b:ae:0c:af:c5
    inet6 fe80::e00b:aaff:fe0c:afc5%llw0 prefixlen 64 scopeid 0xa
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
utun0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1380
    inet6 fe80::f58d:6031:ca37:b987%utun0 prefixlen 64 scopeid 0xb
        nd6 options=201<PERFORMNUD,DAD>
utun1: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 2000
    inet6 fe80::af3a:6b7e:729f:e148%utun1 prefixlen 64 scopeid 0xc
        nd6 options=201<PERFORMNUD,DAD>
utun2: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1380
    inet6 fe80::8888:4408:7f6e:76ea%utun2 prefixlen 64 scopeid 0xd
        nd6 options=201<PERFORMNUD,DAD>
utun3: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1380
    inet6 fe80::2b53:b72c:6e42:c414%utun3 prefixlen 64 scopeid 0xe
        nd6 options=201<PERFORMNUD,DAD>
```

Q1. 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

*Answer:* As shown in the screenshot below, the IP address of client computer(source) is 192.168.1.102 and the TCP port number of client computer is 1161.

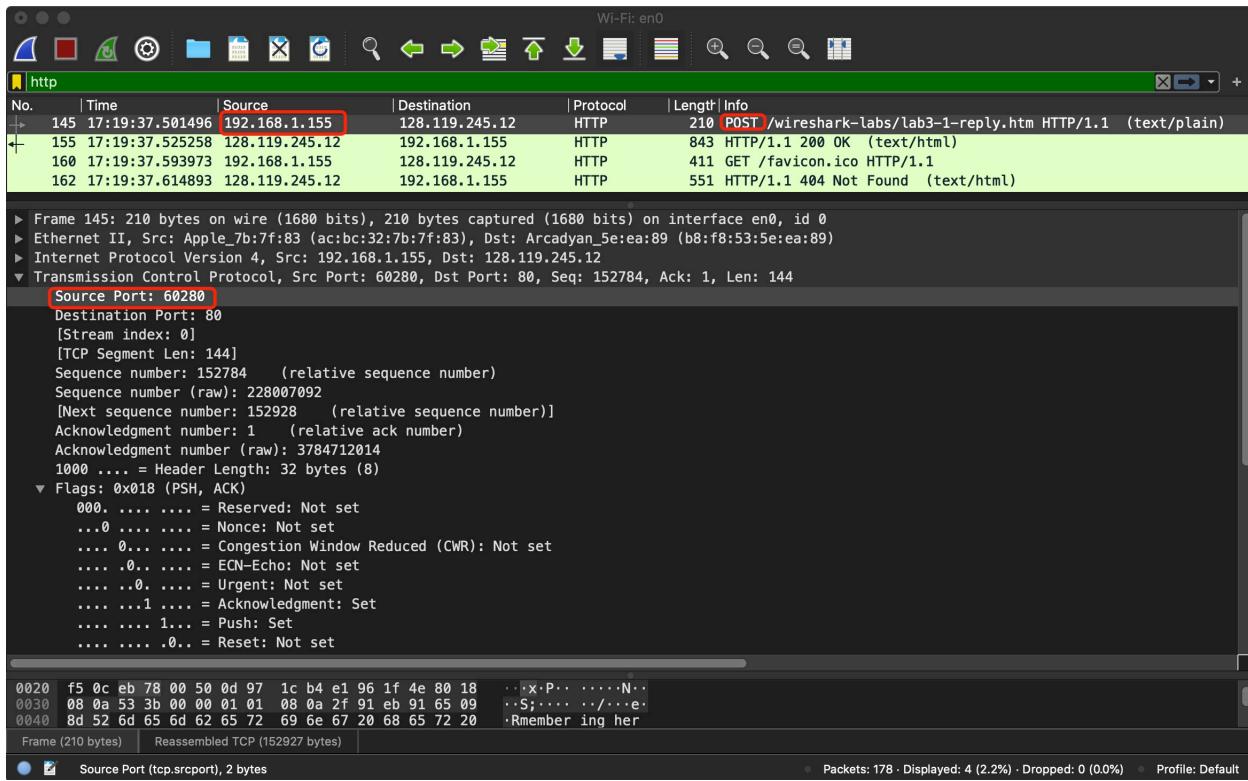


Q2. What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection?

*Answer:* As shown in the screenshot above, the IP address of gaia.cs.umass.edu(destination) is 128.119.245.12 and the TCP port number of destination is 80.

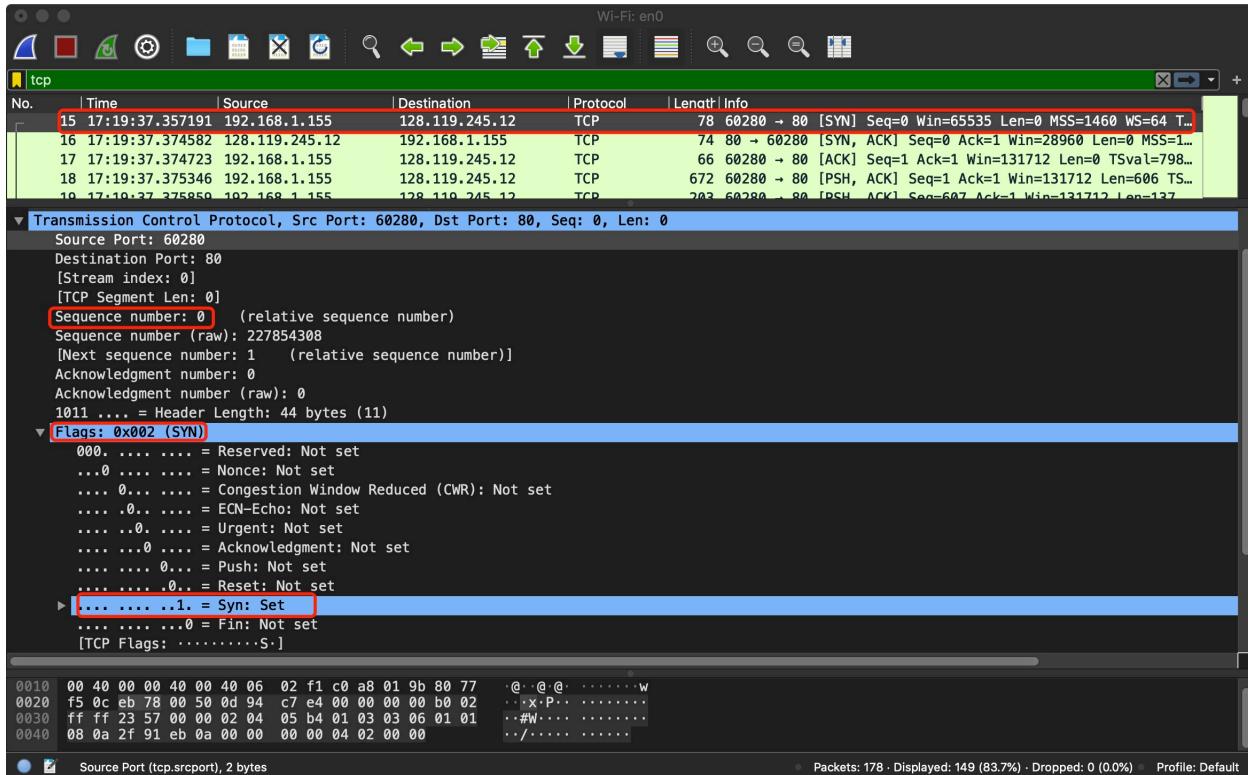
3. What is the IP address and TCP port number used by your client computer (source) to transfer the file to gaia.cs.umass.edu?

*Answer:* As shown in the screenshot below, the IP address of source(my laptop) is 192.168.1.155 and the TCP port number used by my client computer is 60280.



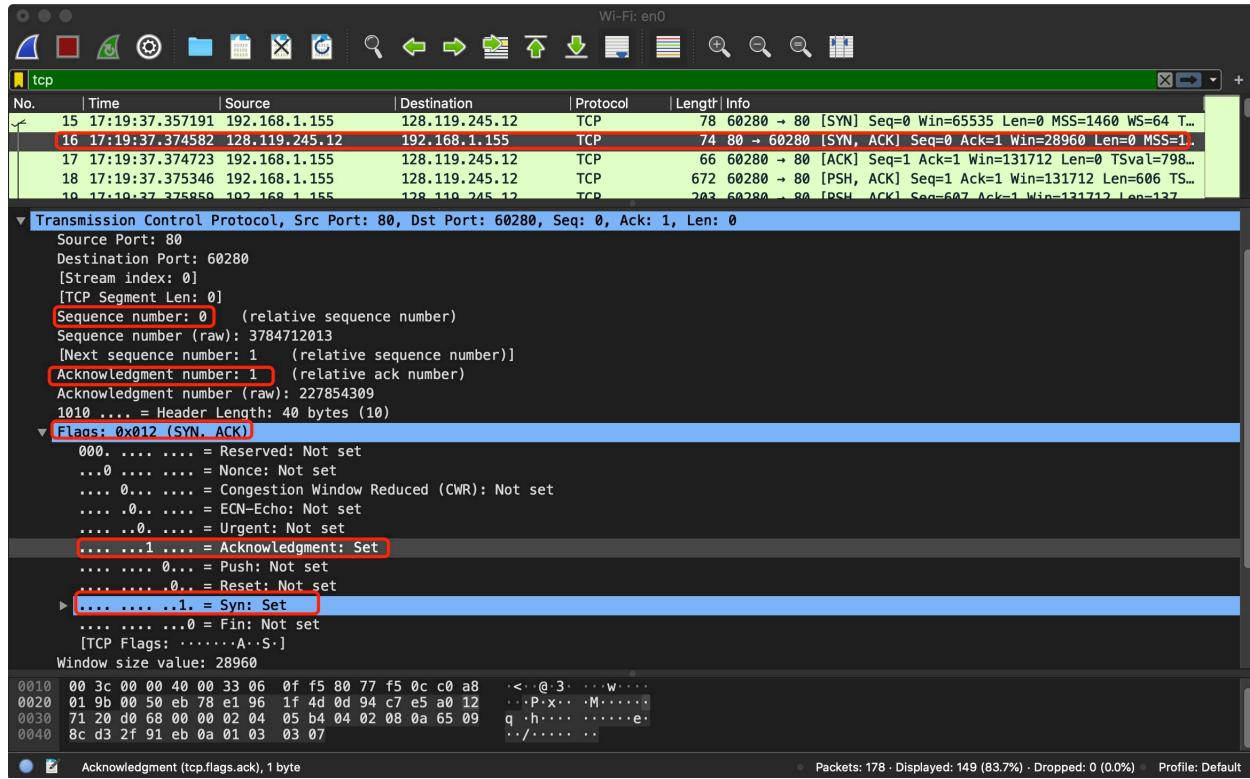
4. What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu? What is it in the segment that identifies the segment as a SYN segment?

*Answer:* As shown in the screenshot below, 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 is 0. Here the SYN flag is 1 so as to identify the segment as a SYN segment.



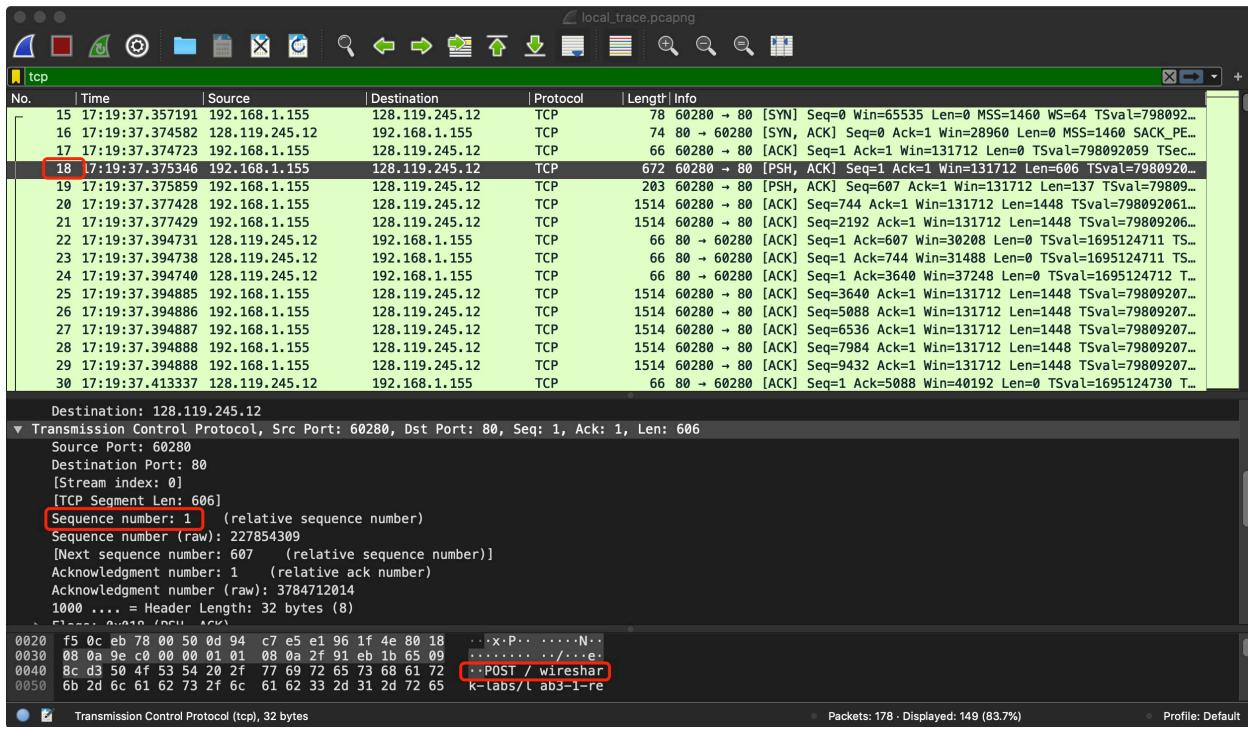
5. What is the sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment? How did gaia.cs.umass.edu determine that value? What is it in the segment that identifies the segment as a SYNACK segment?

*Answer:* As shown in the screenshot below, the sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN is 0. Here the value of the Acknowledgement field in the SYNACK segment is 1. gaia.cs.umass.edu determined that value by adding one to the value of the initial sequence number of SYN segment sent by the client. Here, as we can see in the last screenshot, the initial sequence number of SYN segment sent by client is 0. Here, the SYN flag and Acknowledgement flag were set to 1, which identifies the segment as a SYNACK segment.



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.

*Answer:* As shown in the screenshot below, the packet is No. 18 contains the HTTP POST command, and the sequence number of the TCP segment containing the HTTP POST command is 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.

Note: Wireshark has a nice feature that allows you to plot the RTT for each of the TCP segments sent. Select a TCP segment in the “listing of captured packets” window that is being sent from the client to the gaia.cs.umass.edu server. Then select: Statistics->TCP Stream Graph->Round Trip Time Graph.

*Answer:* As shown in the screenshot below, the first six segments are packets No.18, No.19, No.20, No.21, No.25, No.26.

The ACKs of these segments are packets No. 22, No.23, No.24, No.24, No.30 and No.33

And the sequence numbers of segments are:

Segment 1(No.18) seq number: 1, sent at 17:19:37.375346, ack received at 17:19:37.394731

Segment 2 (No.19) seq number: 607, sent at 17:19:37.375859, ack received at 17:19:37.394738

Segment 3 (No.20) seq number: 744, sent at 17:19:37.377428, ack received at 17:19:37.394740

Segment 4 (No.21) seq number: 2192, sent at 17:19:37.377429, ack received at 17:19:37.394740

Segment 5 (No.25) seq number: 3640, sent at 17:19:37.394885, ack received at 17:19:37.413337

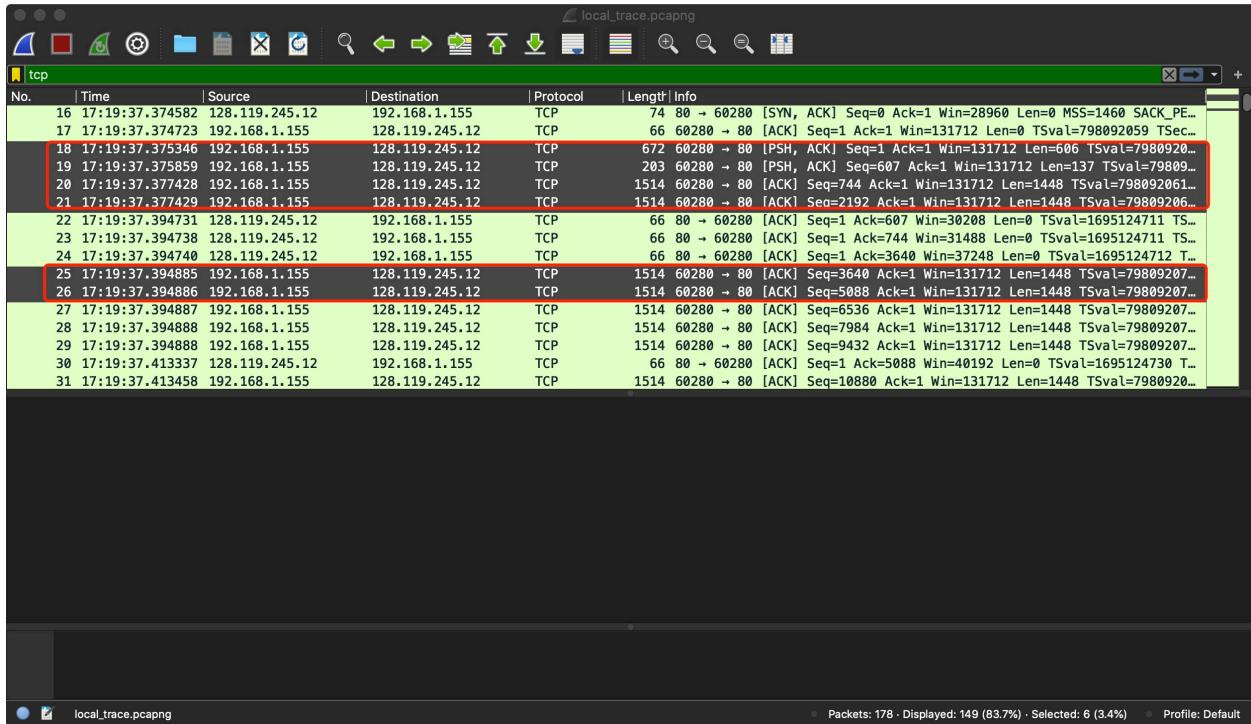
Segment 6 (No.26) seq number: 5088, sent at 17:19:37.394886, ack received at 17:19:37.413726

Here are the RTTs:

Segment 1 RTT:  $0.394731 - 0.375346 = 0.019385$  sec  
Segment 2 RTT:  $0.394738 - 0.375859 = 0.018879$  sec  
Segment 3 RTT:  $0.394740 - 0.377428 = 0.017312$  sec  
Segment 4 RTT:  $0.394740 - 0.377429 = 0.017311$  sec  
Segment 5 RTT:  $0.413337 - 0.394885 = 0.018452$  sec  
Segment 6 RTT:  $0.413726 - 0.394886 = 0.018840$  sec

Here are the EstimatedRTTs (EstimatedRTT=0.875·EstimatedRTT+0.125·SampleRTT):

Segment 1 EstimatedRTT: Segment 1 RTT = 0.019385 sec  
Segment 2 EstimatedRTT:  $0.875 * 0.019385 + 0.125 * 0.018879 = 0.019322$  sec  
Segment 3 EstimatedRTT:  $0.875 * 0.019322 + 0.125 * 0.017312 = 0.019071$  sec  
Segment 4 EstimatedRTT:  $0.875 * 0.019071 + 0.125 * 0.017311 = 0.018851$  sec  
Segment 5 EstimatedRTT:  $0.875 * 0.018851 + 0.125 * 0.018452 = 0.018801$  sec  
Segment 6 EstimatedRTT:  $0.875 * 0.018801 + 0.125 * 0.018840 = 0.018806$  sec

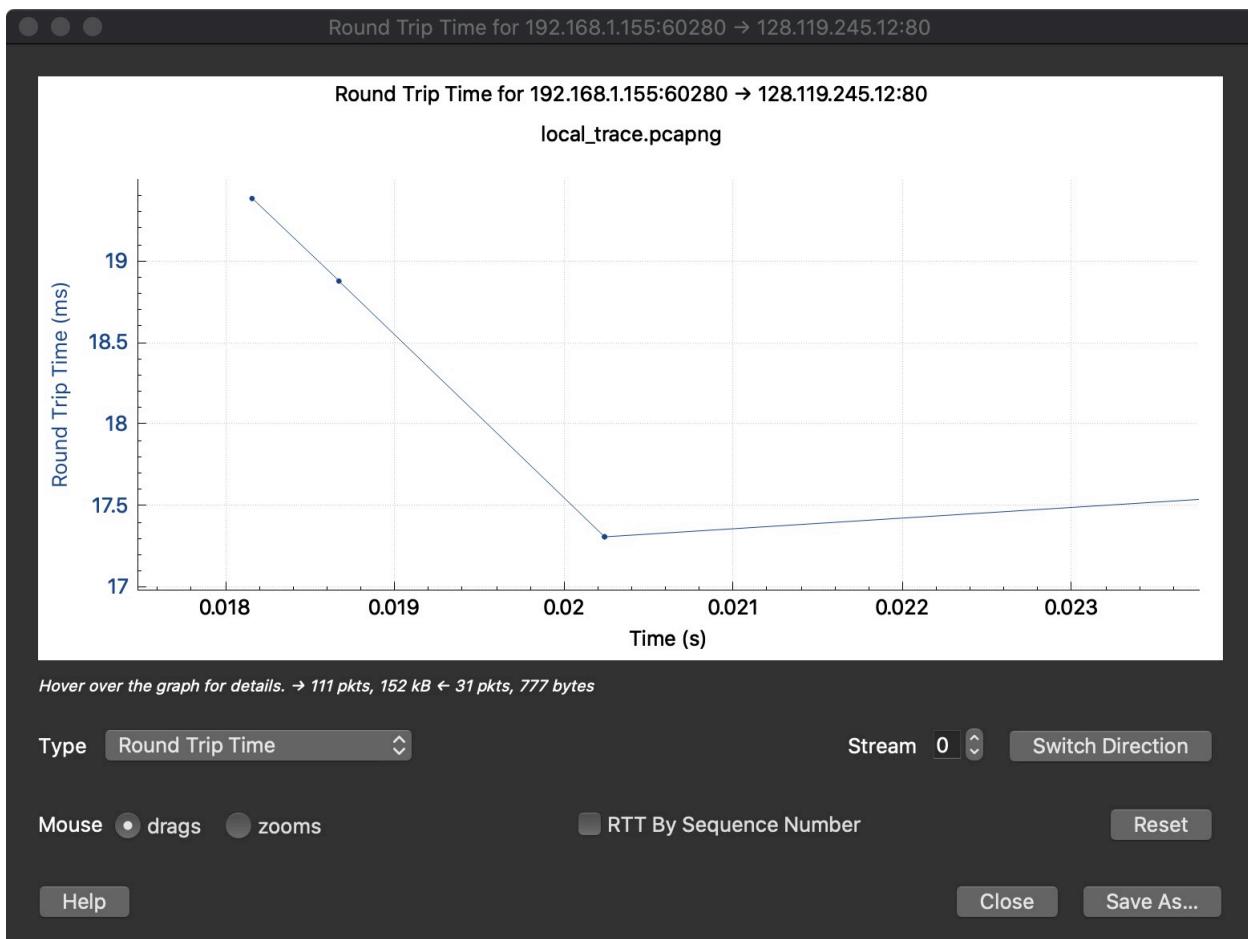


local\_trace.pcapng

tcp

No.	Time	Source	Destination	Protocol	Length	Info
21	17:19:37.377429	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=2192 Ack=1 Win=131712 Len=1448 TSval=79809206...
22	17:19:37.394731	128.119.245.12	192.168.1.155	TCP	66	80 → 60280 [ACK] Seq=1 Ack=607 Win=30208 Len=0 TSval=1695124711 TS...
23	17:19:37.394738	128.119.245.12	192.168.1.155	TCP	66	80 → 60280 [ACK] Seq=1 Ack=744 Win=31488 Len=0 TSval=1695124711 TS...
24	17:19:37.394740	128.119.245.12	192.168.1.155	TCP	66	80 → 60280 [ACK] Seq=1 Ack=3640 Win=37248 Len=0 TSval=1695124712 T...
25	17:19:37.394885	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=3640 Ack=1 Win=131712 Len=1448 TSval=79809207...
26	17:19:37.394886	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=5088 Ack=1 Win=131712 Len=1448 TSval=79809207...
27	17:19:37.394887	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=6536 Ack=1 Win=131712 Len=1448 TSval=79809207...
28	17:19:37.394888	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=7984 Ack=1 Win=131712 Len=1448 TSval=79809207...
29	17:19:37.394888	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=9432 Ack=1 Win=131712 Len=1448 TSval=79809207...
30	17:19:37.413337	128.119.245.12	192.168.1.155	TCP	66	80 → 60280 [ACK] Seq=1 Ack=5088 Win=40192 Len=0 TSval=1695124730 T...
31	17:19:37.413458	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=10880 Ack=1 Win=131712 Len=1448 TSval=7980920...
32	17:19:37.413459	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=12328 Ack=1 Win=131712 Len=1448 TSval=7980920...
33	17:19:37.413726	128.119.245.12	192.168.1.155	TCP	66	80 → 60280 [ACK] Seq=1 Ack=10880 Win=51712 Len=0 TSval=1695124731 ...
34	17:19:37.413842	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=13776 Ack=1 Win=131712 Len=1448 TSval=7980920...
35	17:19:37.413842	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=15224 Ack=1 Win=131712 Len=1448 TSval=7980920...
36	17:19:37.413843	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=16672 Ack=1 Win=131712 Len=1448 TSval=7980920...

Packets: 178 - Displayed: 149 (83.7%) · Selected: 5 (2.8%) · Profile: Default



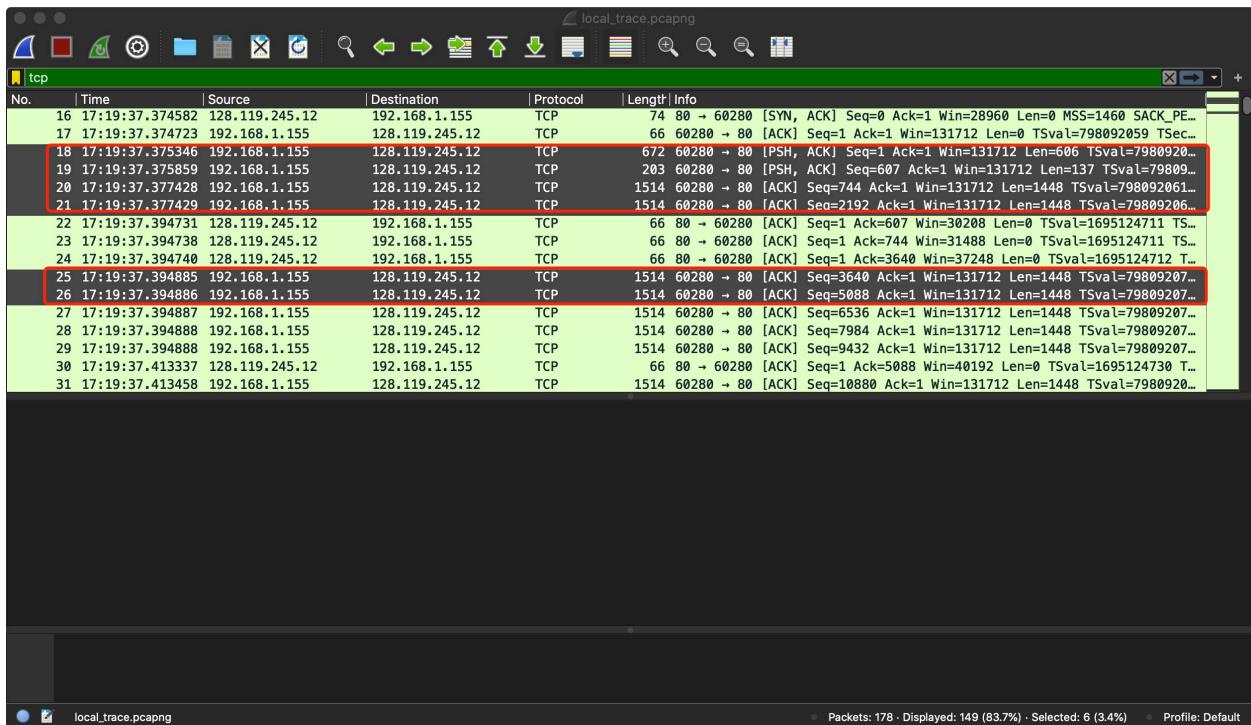
8. What is the length of each of the first six TCP segments?

*Answer:* As shown in the screenshot below, the length of each of the first six TCP segments are:

Segment 1: 606 bytes

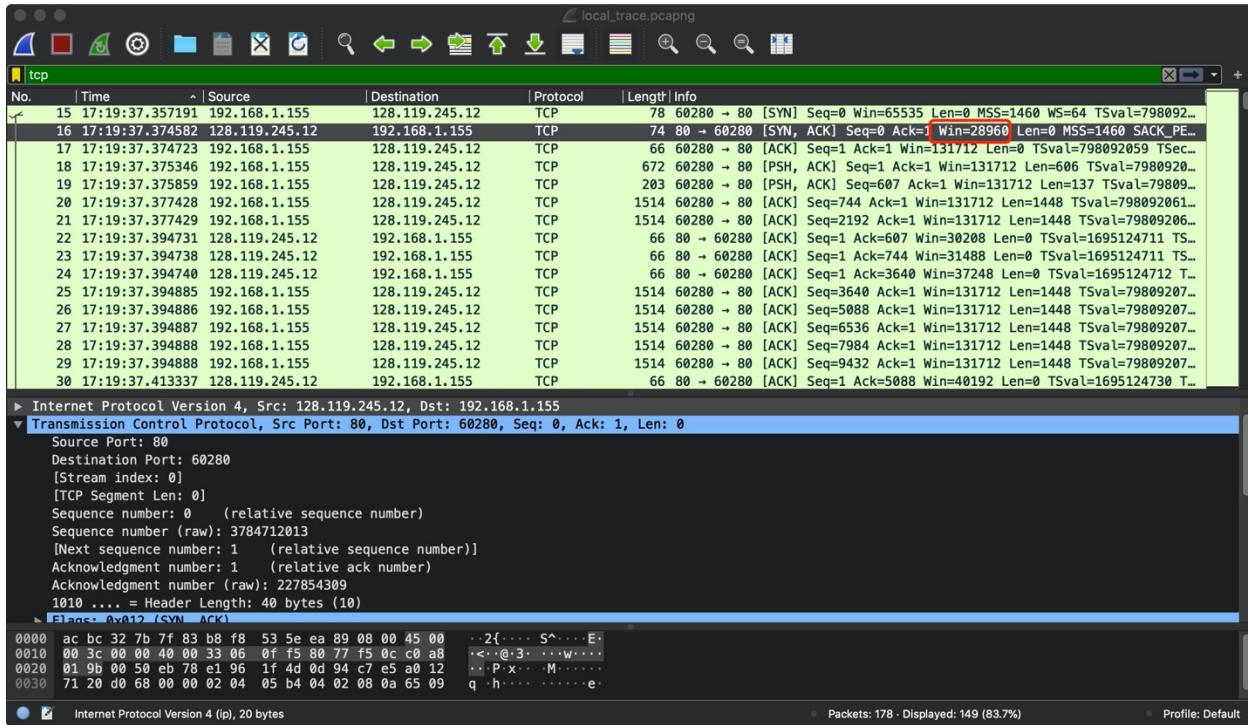
Segment 2: 137 bytes

Segment 3,4,5,6 : 1448 bytes for each



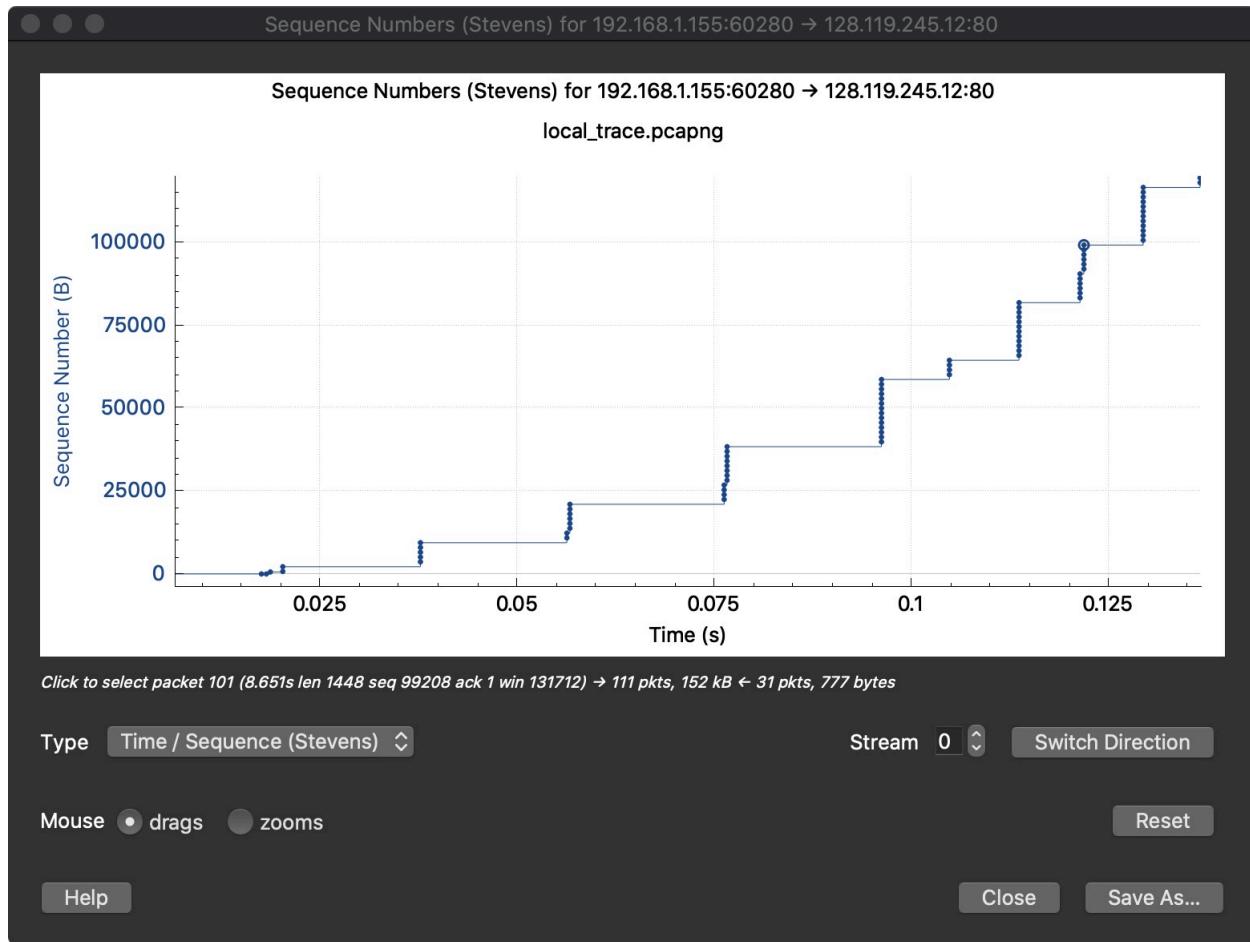
9. What is the minimum amount of available buffer space advertised at the received for the entire trace? Does the lack of receiver buffer space ever throttle the sender?

*Answer:* As shown in the screenshot below the minimum amount of available buffer space advertised at the received for the entire trace is 28960 bytes, which is the first ACK from the server. The window size of other ACK is all larger than 28960 bytes. Lack of receiver buffer space didn't ever throttle the sender as length of the TCP segments didn't decrease, it was keeping in 1448 after the first segment.



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

*Answer:* There isn't any retransmitted segments in the trace file. We can know this from the Time-Sequence\_graph(Stevens) shown below. The Sequence Numbers are always increasing during the transmit process. If there is any retransmitted segment, there should be fluctuation in the graph. The seq number of the retransmitted segment would smaller than the seq number of the segments that sent before and after it.



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).

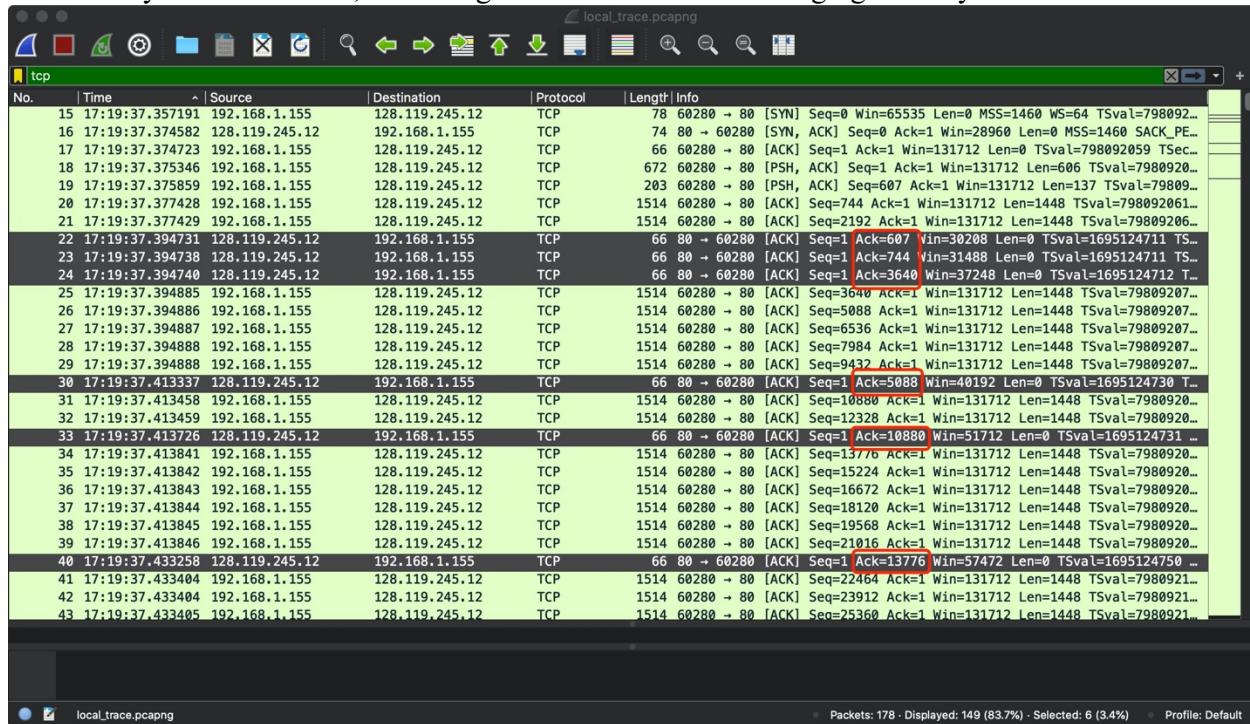
*Answer:*

	Acknowledged sequence number	Acknowledged data
ACK1	607	607
ACK2	744	137
ACK3	3640	2896
ACK4	5088	1448
ACK5	10880	5792
ACK6	13776	2896
ACK7	22464	8688
ACK8	28256	5792
ACK9	36944	8688
ACK10	39840	2896
ACK11	54320	14480
ACK12	60112	5792

ACK13	65904	5792
ACK14	80384	14480
ACK15	83280	2896
ACK17	91968	8688
ACK18	94864	2896
ACK19	96312	1448

The data received by the sever = the difference between the acknowledge sequence numbers of two consecutive ACKs.

We can see the ACK numbers increase in the sequence of 2896, 1448, 5792, 8688 and so on. There were cases where the receiver is ACKing every other segment. For instance, the ACK numbers increases by 2896 each time, indicating the receiver is acknowledging 2896 bytes.





local\_trace.pcapng

tcp

No.	Time	Source	Destination	Protocol	Length	Info
74	17:19:37.461977	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=04450 Ack=1 Win=131712 Len=1448 TStamp=7980921...
75	17:19:37.470673	128.119.245.12	192.168.1.155	TCP	66	80 → 60280 [ACK] Seq=1 Ack=54320 Win=138624 Len=0 TStamp=1695124787...
76	17:19:37.470795	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=65984 Ack=1 Win=131712 Len=1448 TStamp=7980921...
77	17:19:37.470797	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=67352 Ack=1 Win=131712 Len=1448 TStamp=7980921...
78	17:19:37.470799	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=68808 Ack=1 Win=131712 Len=1448 TStamp=7980921...
79	17:19:37.470800	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=70248 Ack=1 Win=131712 Len=1448 TStamp=7980921...
80	17:19:37.470800	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=71696 Ack=1 Win=131712 Len=1448 TStamp=7980921...
81	17:19:37.470802	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=73144 Ack=1 Win=131712 Len=1448 TStamp=7980921...
82	17:19:37.470802	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=74592 Ack=1 Win=131712 Len=1448 TStamp=7980921...
83	17:19:37.470803	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=76040 Ack=1 Win=131712 Len=1448 TStamp=7980921...
84	17:19:37.470804	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=77488 Ack=1 Win=131712 Len=1448 TStamp=7980921...
85	17:19:37.470804	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=78934 Ack=1 Win=131712 Len=1448 TStamp=7980921...
86	17:19:37.470805	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=80384 Ack=1 Win=131712 Len=1448 TStamp=7980921...
87	17:19:37.470805	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=81832 Ack=1 Win=131712 Len=1448 TStamp=7980921...
88	17:19:37.478437	128.119.245.12	192.168.1.155	TCP	66	80 → 60280 [ACK] Seq=1 Ack=60112 Win=150144 Len=0 TStamp=1695124795...
89	17:19:37.478539	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=83280 Ack=1 Win=131712 Len=1448 TStamp=7980921...
90	17:19:37.478540	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=84728 Ack=1 Win=131712 Len=1448 TStamp=7980921...
91	17:19:37.478541	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=86176 Ack=1 Win=131712 Len=1448 TStamp=7980921...
92	17:19:37.478541	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=87624 Ack=1 Win=131712 Len=1448 TStamp=7980921...
93	17:19:37.478542	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=89072 Ack=1 Win=131712 Len=1448 TStamp=7980921...
94	17:19:37.478542	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=90502 Ack=1 Win=131712 Len=1448 TStamp=7980921...
95	17:19:37.478947	128.119.245.12	192.168.1.155	TCP	66	80 → 60280 [ACK] Seq=1 Ack=65984 Win=161792 Len=0 TStamp=1695124796...
96	17:19:37.479825	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=91968 Ack=1 Win=131712 Len=1448 TStamp=7980921...
97	17:19:37.479825	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=93416 Ack=1 Win=131712 Len=1448 TStamp=7980921...
98	17:19:37.479826	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=94864 Ack=1 Win=131712 Len=1448 TStamp=7980921...
99	17:19:37.479827	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=96312 Ack=1 Win=131712 Len=1448 TStamp=7980921...
100	17:19:37.479828	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=97768 Ack=1 Win=131712 Len=1448 TStamp=7980921...
101	17:19:37.479829	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=99228 Ack=1 Win=131712 Len=1448 TStamp=7980921...
102	17:19:37.486387	128.119.245.12	192.168.1.155	TCP	66	80 → 60280 [ACK] Seq=1 Ack=80384 Win=174592 Len=0 TStamp=1695124803...

Packets: 178 - Displayed: 149 (83.7%) - Selected: 8 (4.5%) - Profile: Default

local\_trace.pcapng

tcp

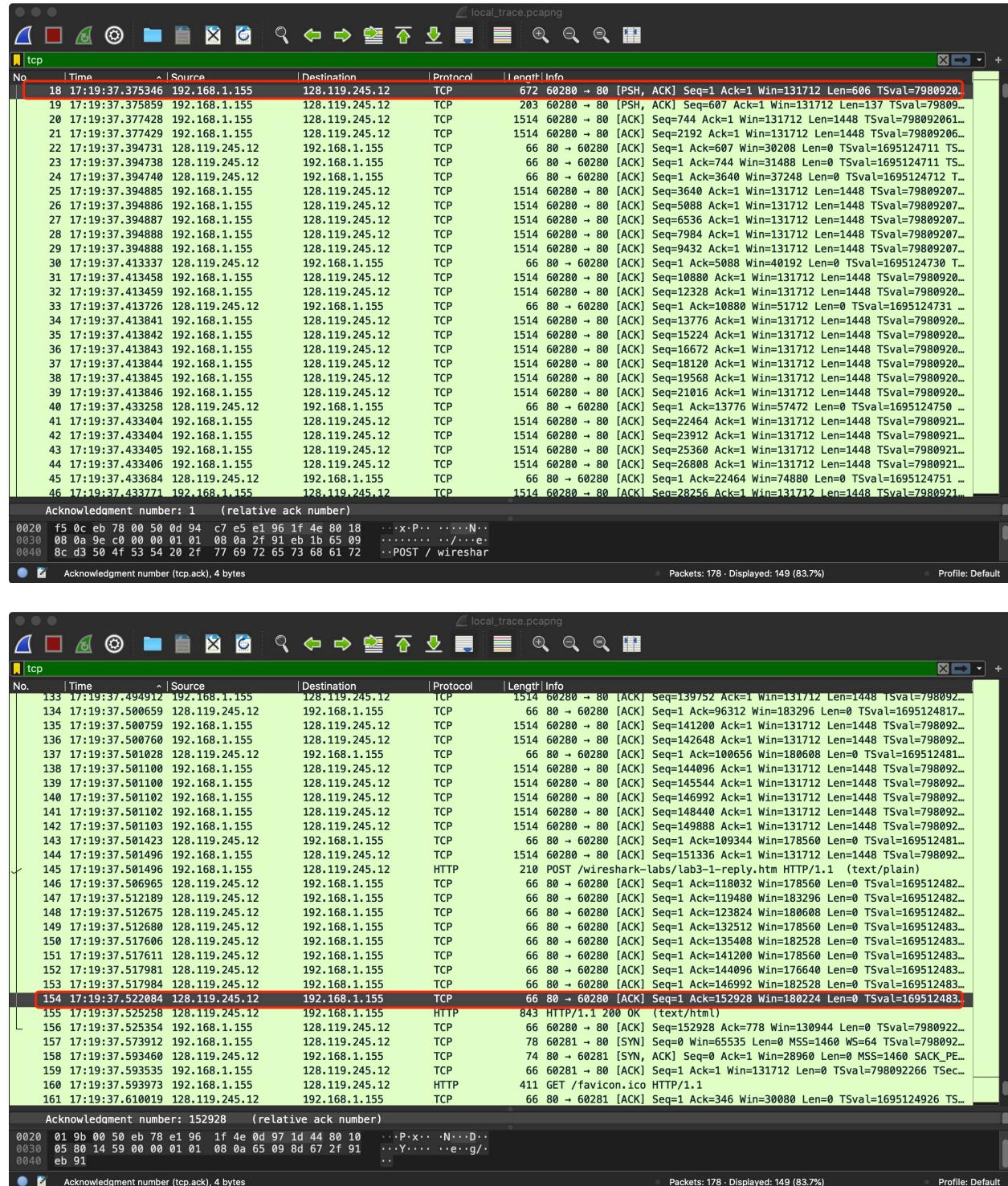
No.	Time	Source	Destination	Protocol	Length	Info
113	17:19:37.486552	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=115136 Ack=1 Win=131712 Len=1448 TStamp=7980921...
114	17:19:37.486553	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=116584 Ack=1 Win=131712 Len=1448 TStamp=7980921...
115	17:19:37.493661	128.119.245.12	192.168.1.155	TCP	66	80 → 60280 [ACK] Seq=1 Ack=83280 Win=182528 Len=0 TStamp=1695124810...
116	17:19:37.493761	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=118832 Ack=1 Win=131712 Len=1448 TStamp=7980921...
117	17:19:37.493761	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=119480 Ack=1 Win=131712 Len=1448 TStamp=7980921...
118	17:19:37.493762	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=120928 Ack=1 Win=131712 Len=1448 TStamp=7980921...
119	17:19:37.493763	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=122376 Ack=1 Win=131712 Len=1448 TStamp=7980921...
120	17:19:37.494033	128.119.245.12	192.168.1.155	TCP	66	80 → 60280 [ACK] Seq=1 Ack=91968 Win=176640 Len=0 TStamp=1695124811...
121	17:19:37.494143	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=123824 Ack=1 Win=131712 Len=1448 TStamp=7980921...
122	17:19:37.494143	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=125272 Ack=1 Win=131712 Len=1448 TStamp=7980921...
123	17:19:37.494145	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=126720 Ack=1 Win=131712 Len=1448 TStamp=7980921...
124	17:19:37.494146	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=128168 Ack=1 Win=131712 Len=1448 TStamp=7980921...
125	17:19:37.494147	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=129616 Ack=1 Win=131712 Len=1448 TStamp=7980921...
126	17:19:37.494148	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=131064 Ack=1 Win=131712 Len=1448 TStamp=7980921...
127	17:19:37.494150	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=132512 Ack=1 Win=131712 Len=1448 TStamp=7980921...
128	17:19:37.494151	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=133968 Ack=1 Win=131712 Len=1448 TStamp=7980921...
129	17:19:37.494813	128.119.245.12	192.168.1.155	TCP	66	80 → 60280 [ACK] Seq=1 Ack=94864 Win=180480 Len=0 TStamp=1695124811...
130	17:19:37.494910	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=135408 Ack=1 Win=131712 Len=1448 TStamp=7980921...
131	17:19:37.494910	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=136856 Ack=1 Win=131712 Len=1448 TStamp=7980921...
132	17:19:37.494911	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=138304 Ack=1 Win=131712 Len=1448 TStamp=7980921...
133	17:19:37.494912	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=139752 Ack=1 Win=131712 Len=1448 TStamp=7980921...
134	17:19:37.500659	128.119.245.12	192.168.1.155	TCP	66	80 → 60280 [ACK] Seq=1 Ack=96312 Win=183296 Len=0 TStamp=1695124817...
135	17:19:37.500759	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=141200 Ack=1 Win=131712 Len=1448 TStamp=7980921...
136	17:19:37.500760	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=142648 Ack=1 Win=131712 Len=1448 TStamp=7980921...
137	17:19:37.501028	128.119.245.12	192.168.1.155	TCP	66	80 → 60280 [ACK] Seq=1 Ack=100656 Win=180608 Len=0 TStamp=169512481...
138	17:19:37.501100	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=144096 Ack=1 Win=131712 Len=1448 TStamp=7980921...
139	17:19:37.501100	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=145544 Ack=1 Win=131712 Len=1448 TStamp=7980921...
140	17:19:37.501102	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=146992 Ack=1 Win=131712 Len=1448 TStamp=7980921...
141	17:19:37.501102	192.168.1.155	128.119.245.12	TCP	1514	60280 → 80 [ACK] Seq=148440 Ack=1 Win=131712 Len=1448 TStamp=7980921...

Packets: 178 - Displayed: 149 (83.7%) - Selected: 13 (7.3%) - Profile: Default

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

*Answer:* Acknowledged sequence number of last ACK is 152928 and the sequence number of the first segment is 1, thus the total data size is  $152928 - 1 = 152927$  bytes.

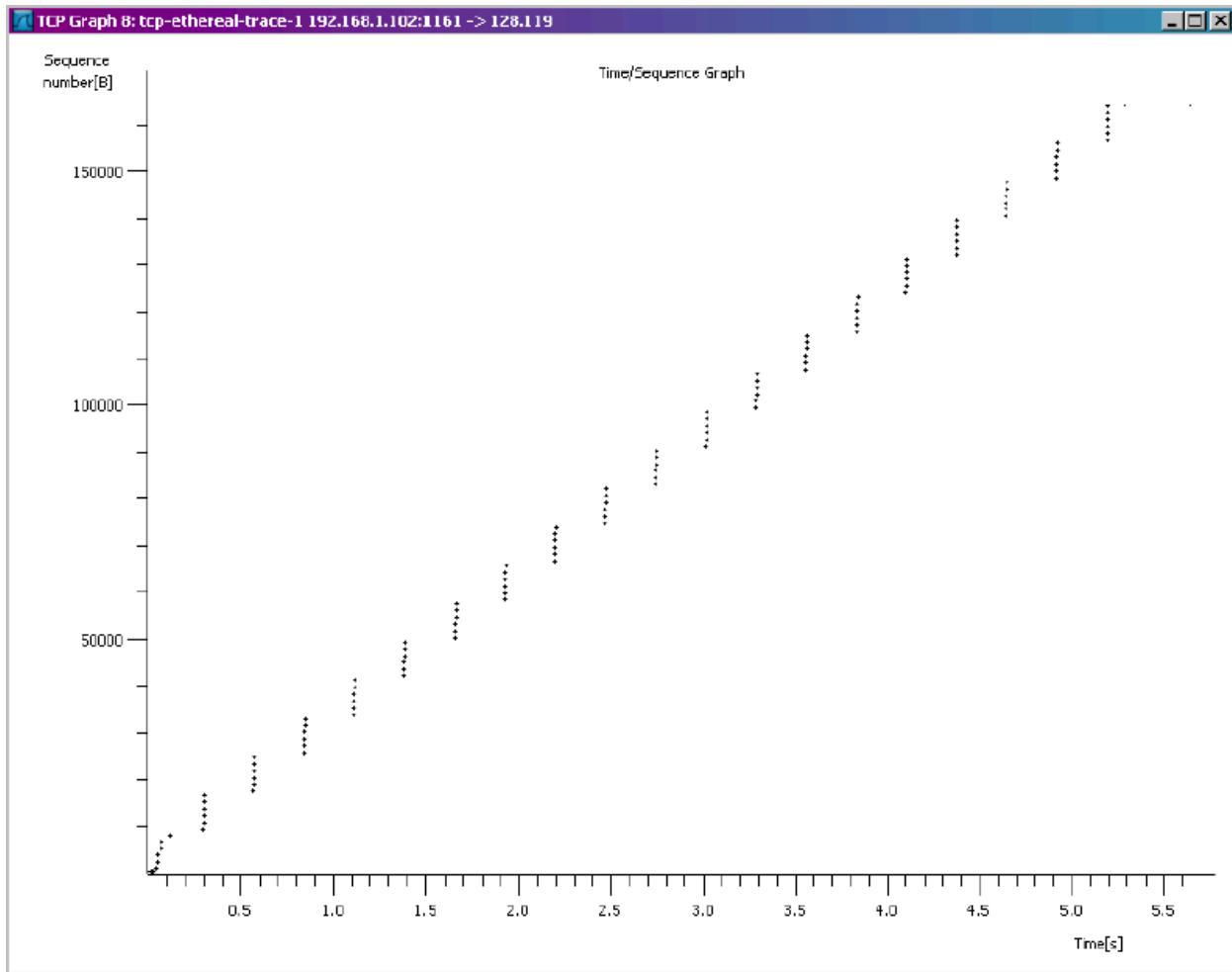
The total transmission time is the time from the first TCP segment (No.18 packet) sent to No.154 packet(the last ACK) sent, thus the time is  $17:19:37.522084 - 17:19:37.375346 = 0.146738$  seconds. Thus, the throughput is  $152927/0.146738 = 1042177.2138$  bytes/sec =  $1.0421772138$  Mb/sec



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.

*Answer:* As we can see from the graph, the slow start phase is around 0 second and ends around 0.15 second. After the slow start phase, congestion tasks over. The measured data uses only a fraction of the window size, rather than 1/3 to 1/2.



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

*Answer:* As we can see from the graph below, the slow start phase is around 0.017 second and ends around 0.021 second. After the slow start phase, congestion tasks over. Here we can't see the expected linear increase behavior. Instead, it appears the repeated behavior pattern, and it seems that the sender transmits packets in batches of around 12. The reason for this behavior might be due to the fact that the HTTP server has enforced a rate-limit of some sort.

