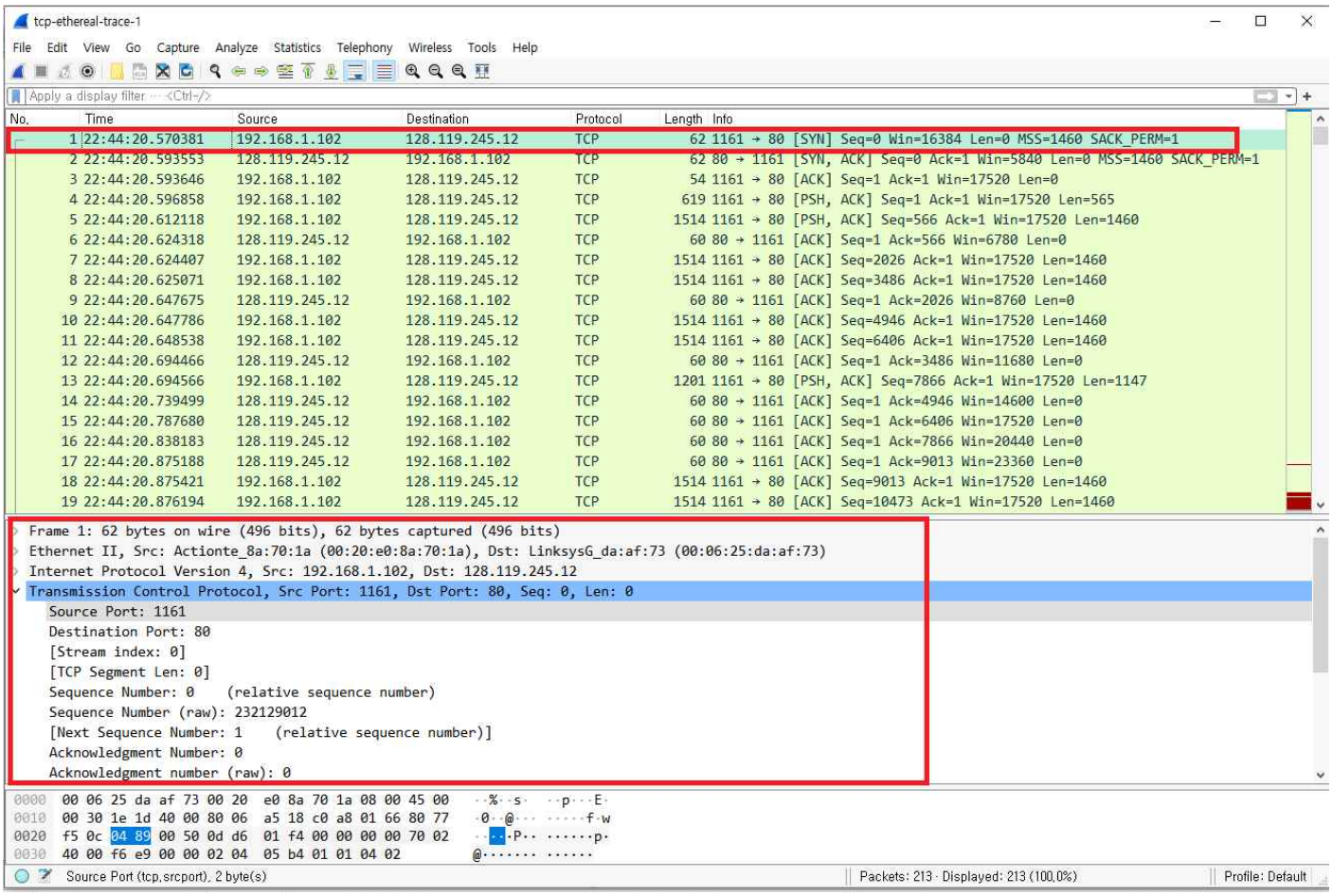
Homework #5 (Wireshark TCP)

201724419 KimDongUk

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.



- Answer

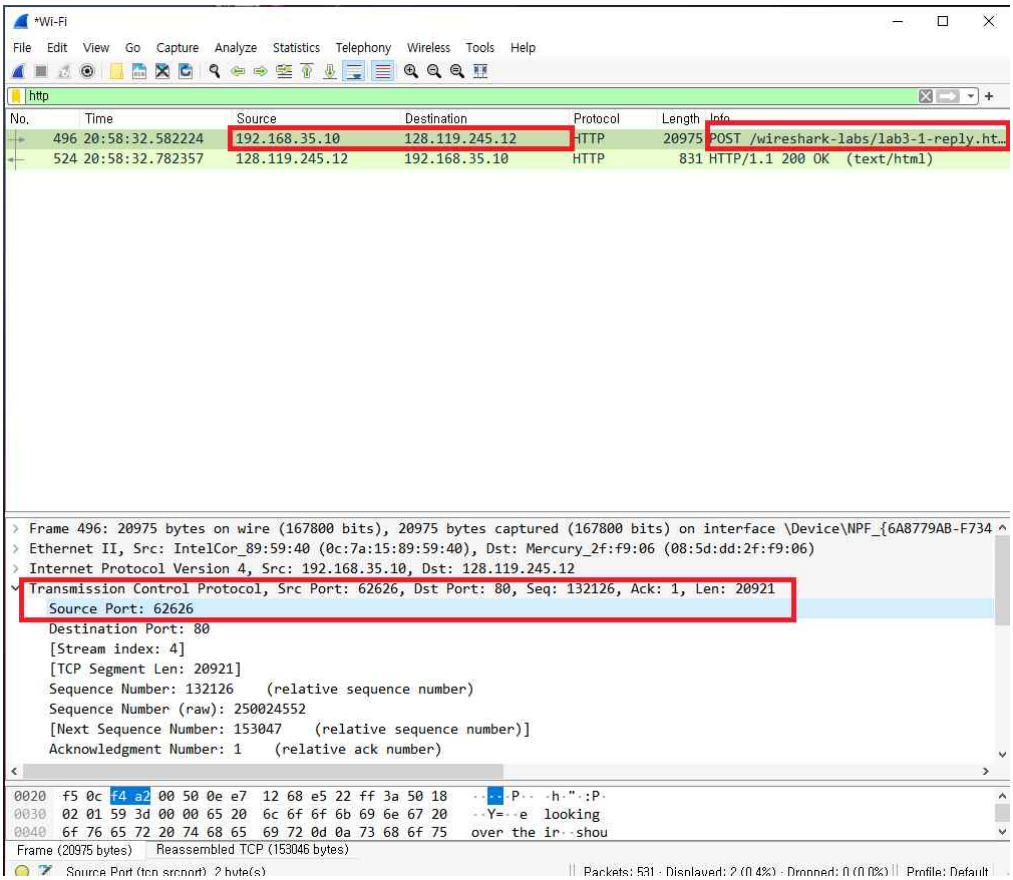
This picture shows that the result of opening 'tcp-etheral-trace-1' file downloaded from gaia.cs.umass. The http message was not found in the 'tcp-etheral-trace-1' file received from gaia.cs.umass.edu. Considering the process of exchanging Syn/Ack for TCP connection, I can guess the client's IP address and TCP port. **TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu is 1161, IP address is 192.168.1.102**

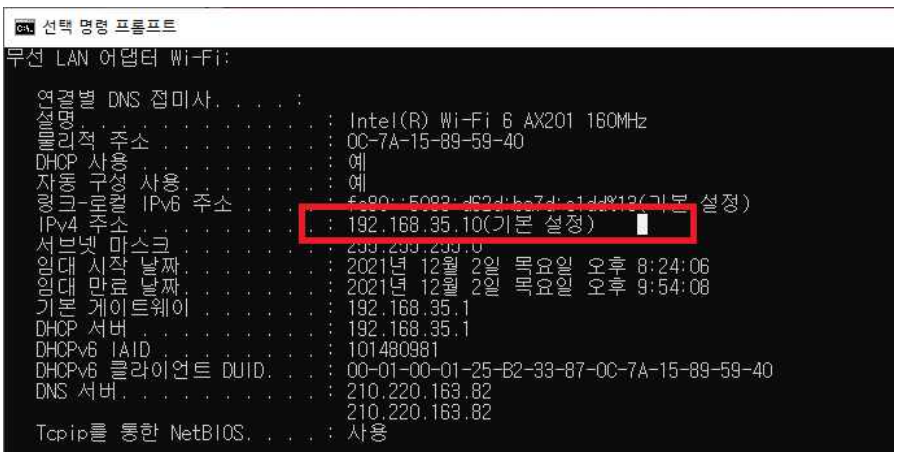
2. What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection?

- Answer

Like first question's picture, **IP address of gaia.cs.umass.edu is 128.119.245.12 and Port number 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

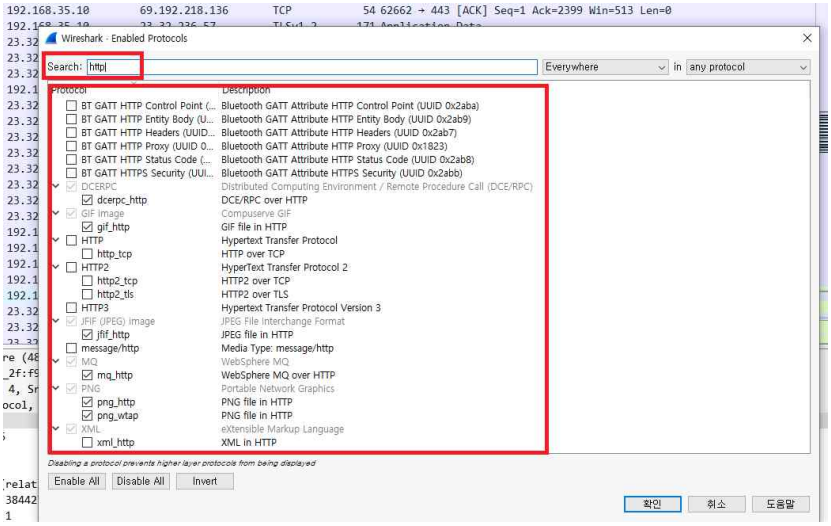
First picture show that the HTTP packet result of sending alice.txt file to gaia server in my computer. Like 1st, 2nd, 3rd, 4st Homework, TCP port number can be detected in packet header.

And this wireshark TCP lab says that sending alice.txt file uses **HTTP POST** way. So I filter the word 'HTTP' and HTTP message likes "POST /wireshark-labs can be founded.

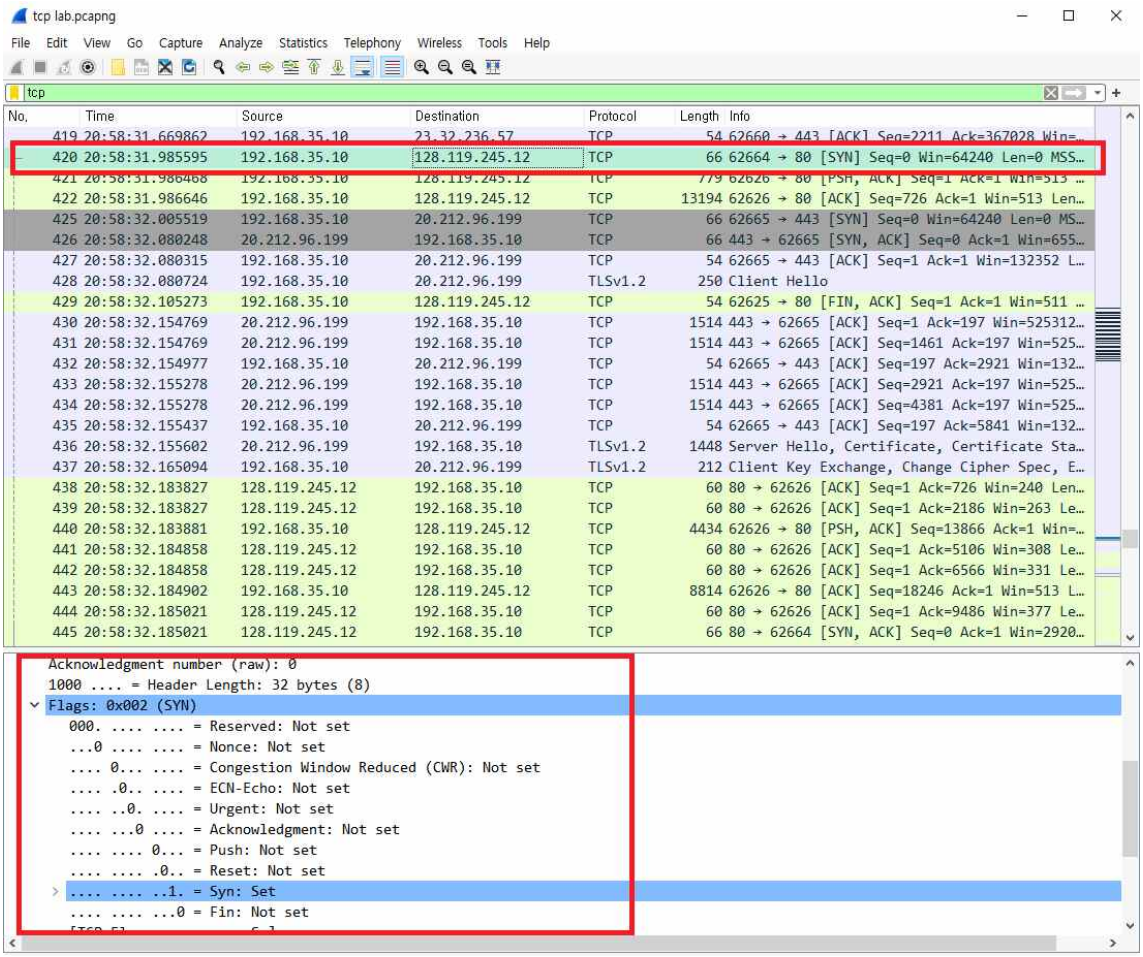
The IP address and TCP port number used by your client computer (source) to transfer the file to gaia.cs.umass.edu is 192.168.35.10 and 62626.

Second picture shows that the result of command ipconfig /all in cmd to find my PC's IP address.

● To solve the problem below, in wireshark -> Analyze->Enabled Protocols HTTP was unchecked



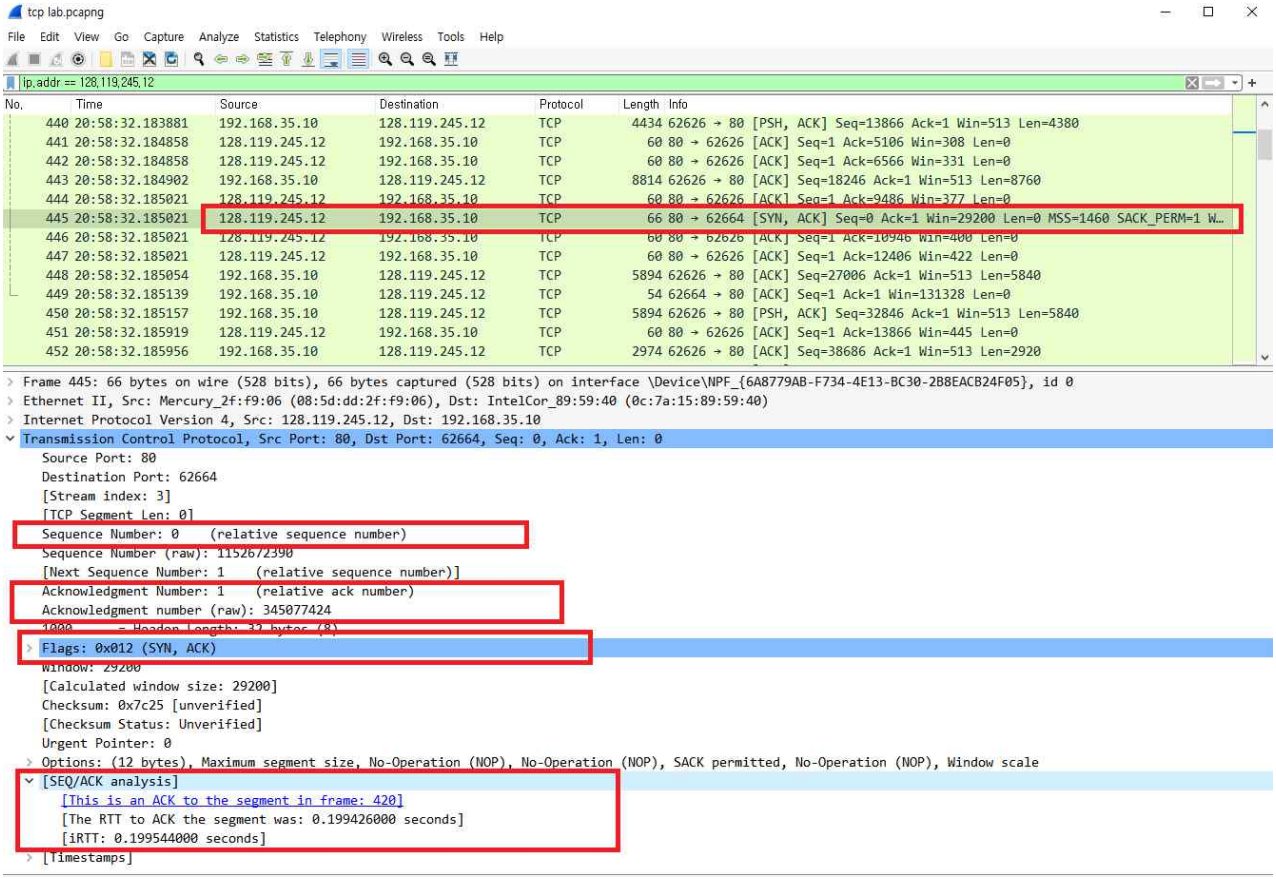
4. What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu? What is it in the segment that identifies the segment as a SYN segment?



- Answer

This picture shows that the TCP SYN segment that is used to initiate the TCP connection. To use tcp connection, the number of tcp segment sequence is 0. And in that segment, there is a information like Flags which contain Syn: Set.

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

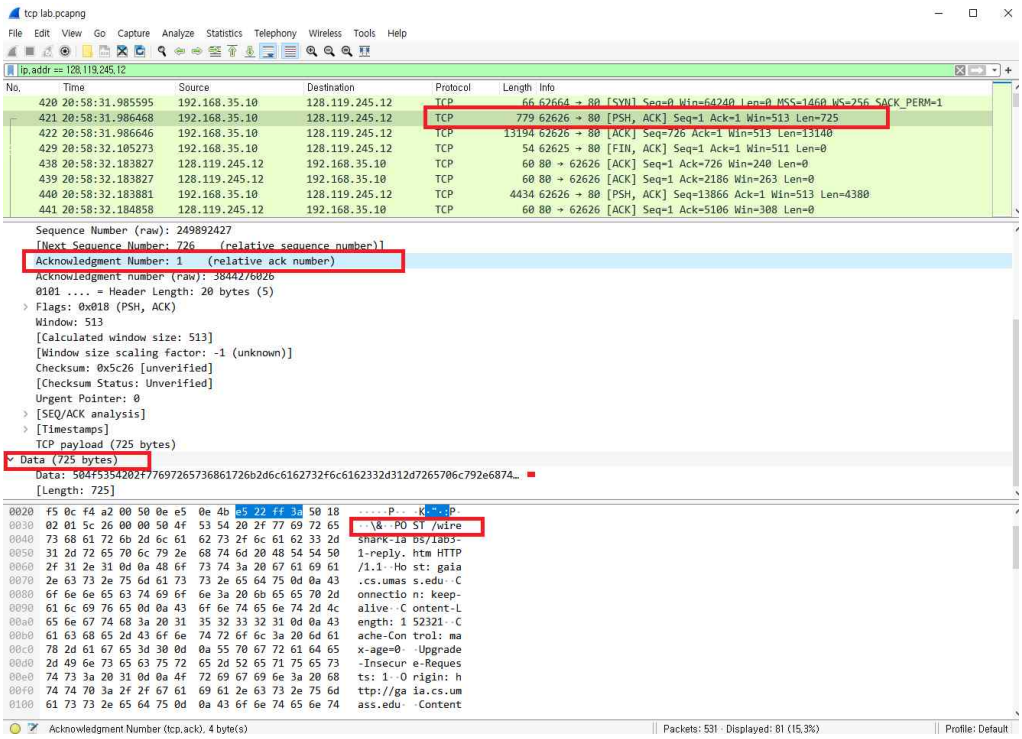
This picture shows that the sequence number of the SYNACK segment is 0.

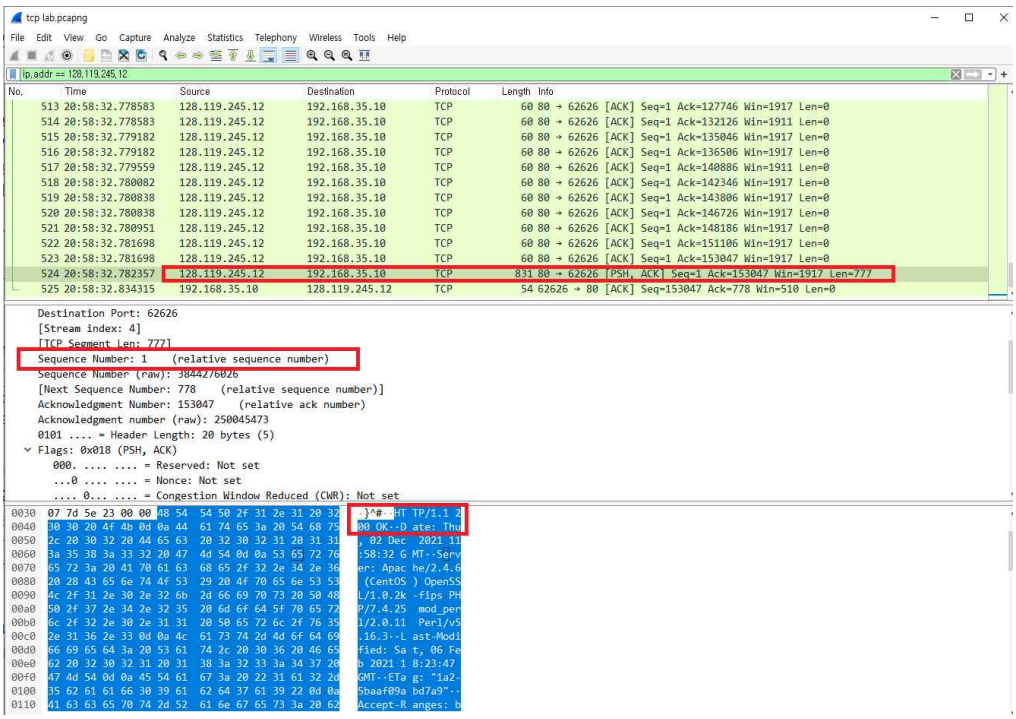
The value of the Acknowledgement field in the SYNACK segment is 1 (relative ack number).

gaia.cs.umass.edu determine that value by using initial sequence number. =>initial sequence number+1

The segment that identifies the segment as a SYNACK segment contains RTT time, Flag, iRTT time and frame number.

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

First picture show that that the contents of [PSH, ACK] packet. Its DATA field has POST /wireshark...

The sequence number of the TCP segment containing the HTTP POST command is 1.

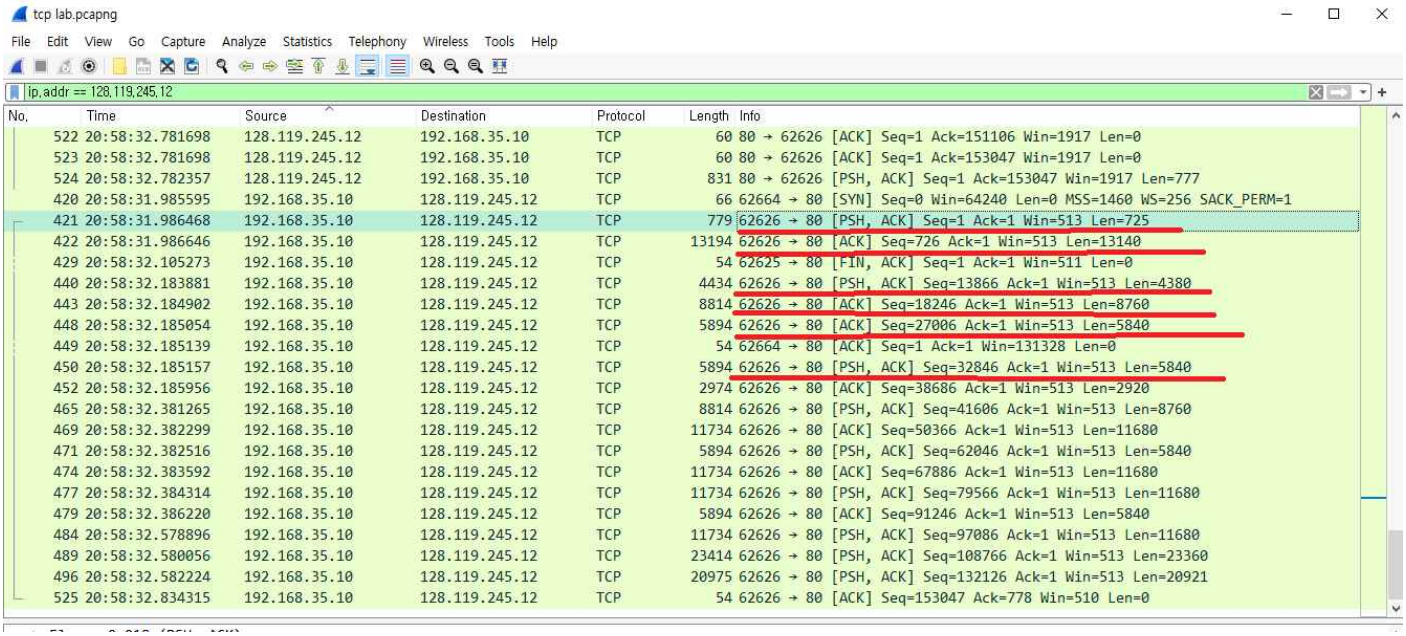
Second picture show that the contents of [PSH, ACK] packet. Its DATA field has HTTP/1.1 200 OK.

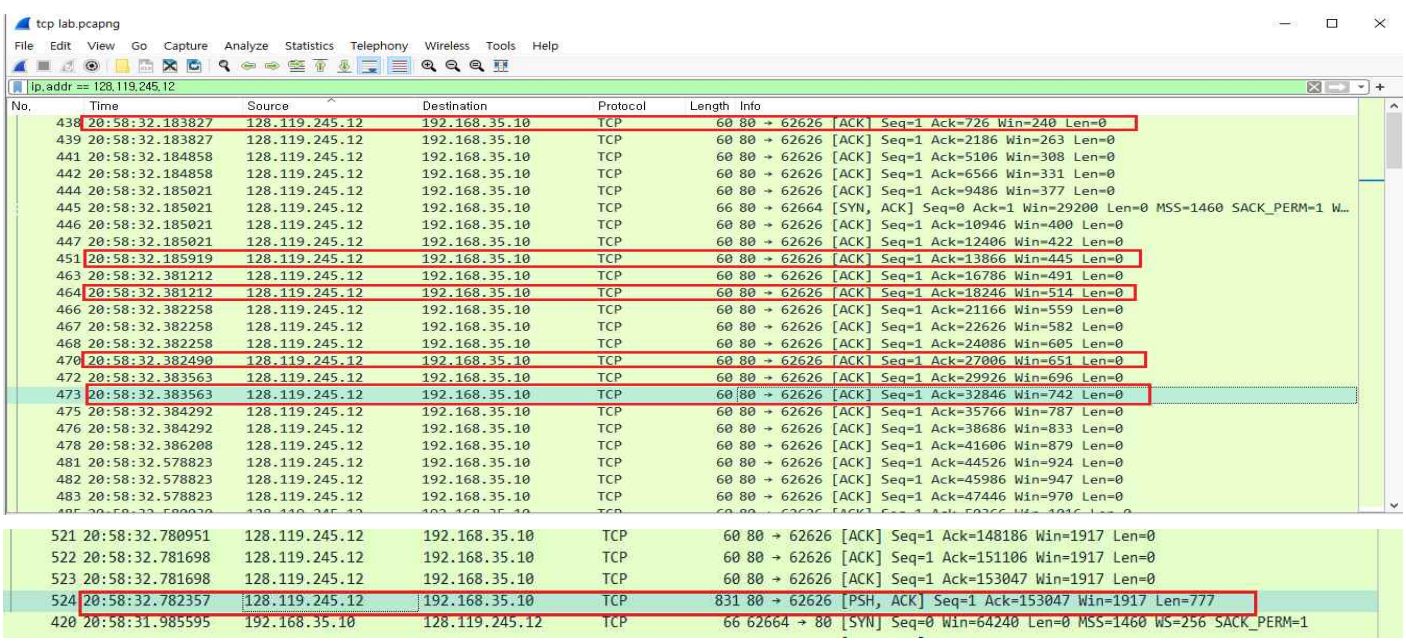
I can't find any 'POST' in DATA field. But HTTP/1.1 200 OK message is success for request(POST).

So I can guess that this is response to first TCP segment containing the HTTP POST command.

As a result **the sequence number of this TCP segment 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?





- Answer

The sequence numbers of the first six segments are 1, 726, 13866, 18426, 27006, 32846

The time of the first six segments are 20:58:31.986468, 20:58:31.986646, 20:58:32.183881,20:58:32.184902, 20:58:32.185054, 20:58:32.185157

The each ACK times are 20:58:32.183827(726), 20:58:32.185919(13866), 20:58:32.381212(18426), 20:58:32.382490(27006), 20:58:32.383563(32846), 20:58:32.782357(1)

* (#) is a number corresponding to seq number.

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

The RTT time is send time - receive time.

seq 1: 20:58:32.782357 - 20:58:31.986468 = 0.795889(s)

seq 726: 20:58:32.183827 - 20:58:31.986646 = 0.197181(s)

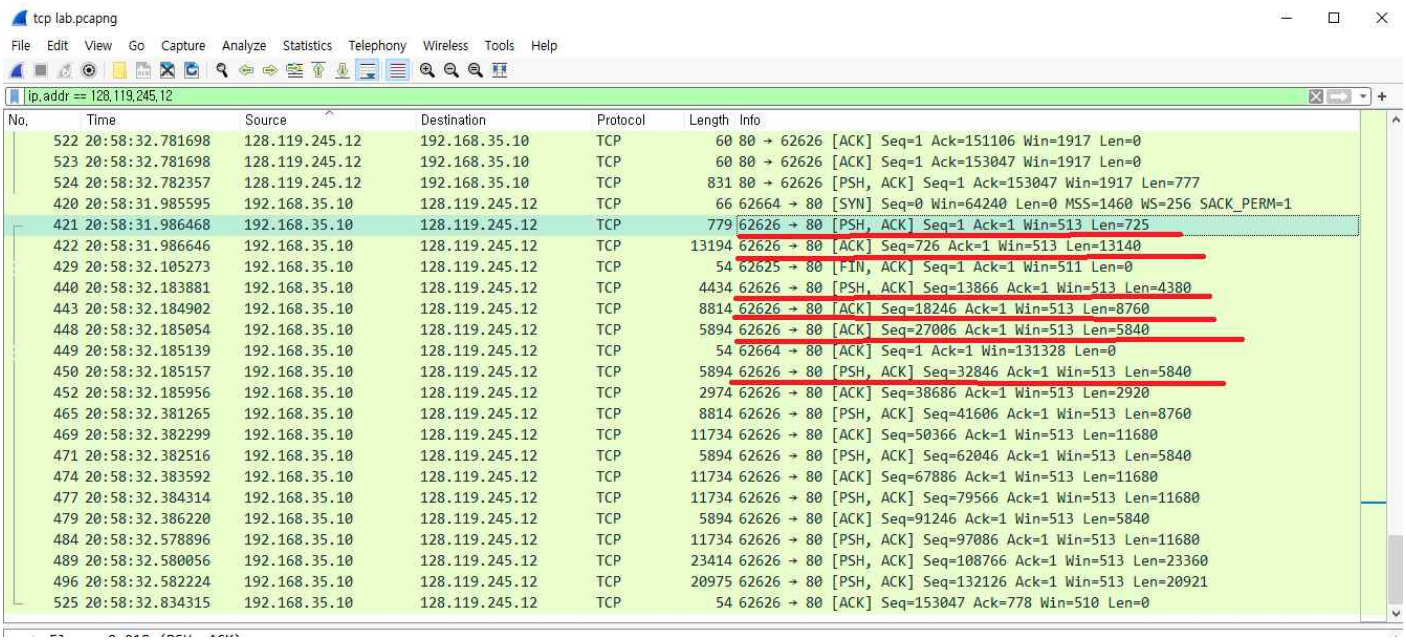
seq 13866: 20:58:32.185919 - 20:58:32.183881 = 0.002038(s)

seq 18426: 20:58:32.381212 - 20:58:32.184902 = 0.196310(s)

seq 27006: 20:58:32.382490 - 20:58:32.185054 = 0.197436(s)

seq 32846: 20:58:32.383563 - 20:58:32.185157 = 0.198406(s)

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



- Answer

In packet info, the length of each of the first six TCP segments is 'Len=#'

seq 1: 725 bytes

seq 726: 13140 bytes

seq 13866: 4380 bytes

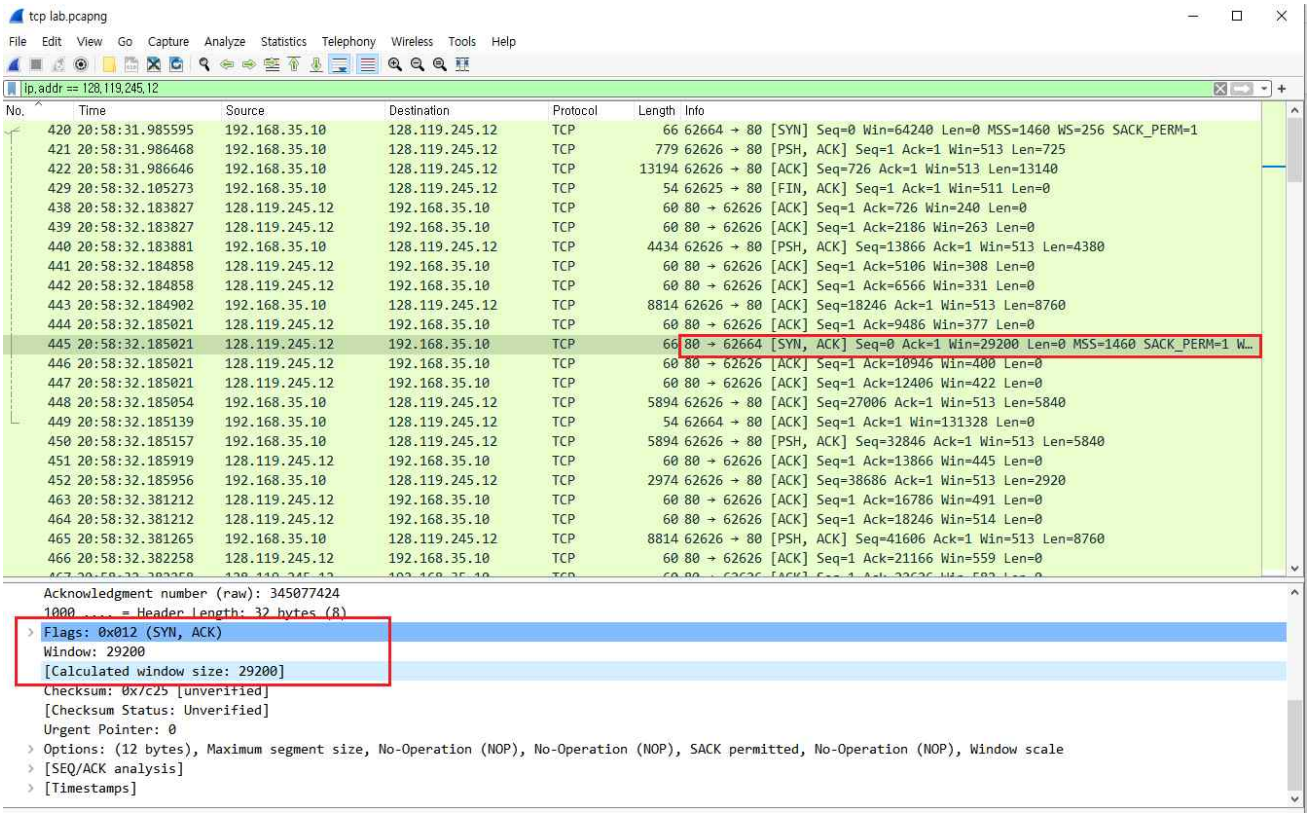
seq 18426: 8760 bytes

seq 27006: 5840 bytes

seq 32846: 5840 bytes

Exactly I know the maximum length of TCP/IP Packet is 1500 bytes. But the length of TCP/IP packets measured by the wire shark on my computer is all over 1500 bytes. I don't know why this is happened..

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?



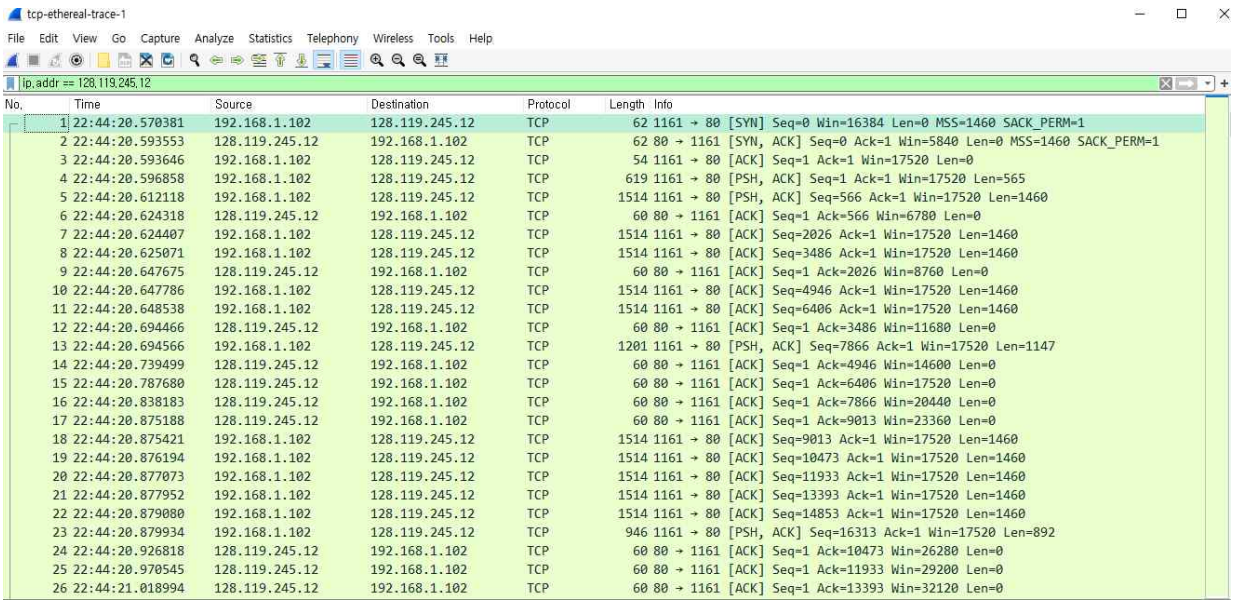
- Answer

This picture shows that each packet's info (Seq=#, Ack=#, Win=#, ….).

Available buffer space advertised at the received for the entire trace is 'Calculated window size' from gaia server with first ACK. the minimum amount of available buffer space is 29,200 bytes.

All of each packet's Win size never goes up to 29,200 bytes. So the sender never throttles receiver buffer space.

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


tcp-ethereal-trace-1

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.addr == 128.119.245.12

No.	Time	Source	Destination	Protocol	Length	Info
1	22:44:20.570381	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	22:44:20.593553	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=1
3	22:44:20.593646	192.168.1.102	128.119.245.12	TCP	54	1161 → 80 [ACK] Seq=1 Ack=1 Win=17520 Len=0
4	22:44:20.596858	192.168.1.102	128.119.245.12	TCP	619	1161 → 80 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=565
5	22:44:20.612118	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [PSH, ACK] Seq=566 Ack=1 Win=17520 Len=1460
6	22:44:20.624318	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=566 Win=6780 Len=0
7	22:44:20.624407	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=2026 Ack=1 Win=17520 Len=1460
8	22:44:20.625071	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=3486 Ack=1 Win=17520 Len=1460
9	22:44:20.647675	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=2026 Win=8760 Len=0
10	22:44:20.647786	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=4946 Ack=1 Win=17520 Len=1460
11	22:44:20.648538	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=6406 Ack=1 Win=17520 Len=1460
12	22:44:20.694466	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=3486 Win=11680 Len=0
13	22:44:20.694566	192.168.1.102	128.119.245.12	TCP	1201	1161 → 80 [PSH, ACK] Seq=7866 Ack=1 Win=17520 Len=1147
14	22:44:20.739499	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=4946 Win=14600 Len=0
15	22:44:20.787680	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=6406 Win=17520 Len=0
16	22:44:20.838183	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=7866 Win=20440 Len=0
17	22:44:20.875188	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=9013 Win=23360 Len=0
18	22:44:20.875421	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=9013 Ack=1 Win=17520 Len=1460
19	22:44:20.876194	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=10473 Ack=1 Win=17520 Len=1460
20	22:44:20.877073	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=11933 Ack=1 Win=17520 Len=1460
21	22:44:20.877952	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=13393 Ack=1 Win=17520 Len=1460
22	22:44:20.879080	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=14853 Ack=1 Win=17520 Len=1460
23	22:44:20.879934	192.168.1.102	128.119.245.12	TCP	946	1161 → 80 [PSH, ACK] Seq=16313 Ack=1 Win=17520 Len=892
24	22:44:20.926818	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=10473 Win=26280 Len=0
25	22:44:20.970545	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=11933 Win=29200 Len=0
26	22:44:21.018994	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=13393 Win=32120 Len=0

- Answer

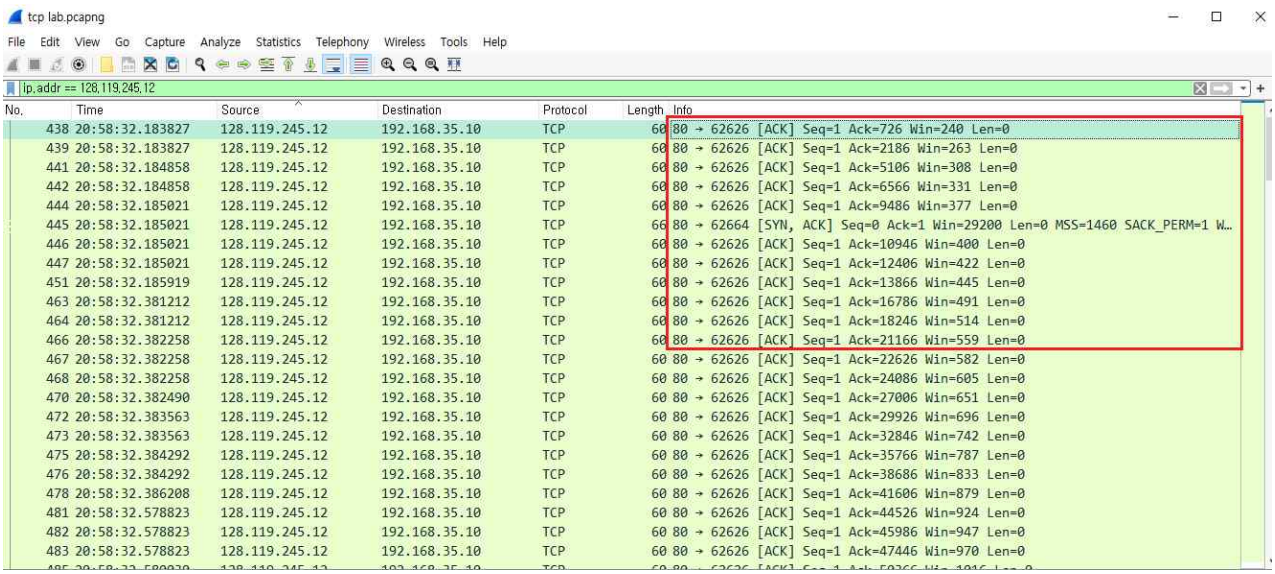
This picture shows that the packet's sequence number using 'tcp-ethereal-trace-1' downloaded in gaia server. The retransmitted segment will be sended if there is segment which failed to send.

Also this **retransmitted segment number is repeated with the number that came out already.**

According to above picture, as time goes on, there is no duplicated segment number except syn-ack.

So there is no retransmitted segments.

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).


tcp lab.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.addr == 128.119.245.12

No.	Time	Source	Destination	Protocol	Length	Info
438	20:58:32.183827	128.119.245.12	192.168.35.10	TCP	60	80 → 62626 [ACK] Seq=1 Ack=726 Win=240 Len=0
439	20:58:32.183827	128.119.245.12	192.168.35.10	TCP	60	80 → 62626 [ACK] Seq=1 Ack=2186 Win=263 Len=0
441	20:58:32.184858	128.119.245.12	192.168.35.10	TCP	60	80 → 62626 [ACK] Seq=1 Ack=5106 Win=308 Len=0
442	20:58:32.184858	128.119.245.12	192.168.35.10	TCP	60	80 → 62626 [ACK] Seq=1 Ack=6566 Win=331 Len=0
444	20:58:32.185021	128.119.245.12	192.168.35.10	TCP	60	80 → 62626 [ACK] Seq=1 Ack=9486 Win=377 Len=0
445	20:58:32.185021	128.119.245.12	192.168.35.10	TCP	66	80 → 62664 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 W…
446	20:58:32.185021	128.119.245.12	192.168.35.10	TCP	60	80 → 62626 [ACK] Seq=1 Ack=10946 Win=400 Len=0
447	20:58:32.185021	128.119.245.12	192.168.35.10	TCP	60	80 → 62626 [ACK] Seq=1 Ack=12406 Win=422 Len=0
451	20:58:32.185919	128.119.245.12	192.168.35.10	TCP	60	80 → 62626 [ACK] Seq=1 Ack=13866 Win=445 Len=0
463	20:58:32.381212	128.119.245.12	192.168.35.10	TCP	60	80 → 62626 [ACK] Seq=1 Ack=16786 Win=491 Len=0
464	20:58:32.381212	128.119.245.12	192.168.35.10	TCP	60	80 → 62626 [ACK] Seq=1 Ack=18246 Win=514 Len=0
466	20:58:32.382258	128.119.245.12	192.168.35.10	TCP	60	80 → 62626 [ACK] Seq=1 Ack=21166 Win=559 Len=0
467	20:58:32.382258	128.119.245.12	192.168.35.10	TCP	60	80 → 62626 [ACK] Seq=1 Ack=22626 Win=582 Len=0
468	20:58:32.382258	128.119.245.12	192.168.35.10	TCP	60	80 → 62626 [ACK] Seq=1 Ack=24086 Win=605 Len=0
470	20:58:32.382490	128.119.245.12	192.168.35.10	TCP	60	80 → 62626 [ACK] Seq=1 Ack=27006 Win=651 Len=0
472	20:58:32.383563	128.119.245.12	192.168.35.10	TCP	60	80 → 62626 [ACK] Seq=1 Ack=29926 Win=696 Len=0
473	20:58:32.383563	128.119.245.12	192.168.35.10	TCP	60	80 → 62626 [ACK] Seq=1 Ack=32846 Win=742 Len=0
475	20:58:32.384292	128.119.245.12	192.168.35.10	TCP	60	80 → 62626 [ACK] Seq=1 Ack=35766 Win=787 Len=0
476	20:58:32.384292	128.119.245.12	192.168.35.10	TCP	60	80 → 62626 [ACK] Seq=1 Ack=38686 Win=833 Len=0
478	20:58:32.386208	128.119.245.12	192.168.35.10	TCP	60	80 → 62626 [ACK] Seq=1 Ack=41606 Win=879 Len=0
481	20:58:32.578823	128.119.245.12	192.168.35.10	TCP	60	80 → 62626 [ACK] Seq=1 Ack=44526 Win=924 Len=0
482	20:58:32.578823	128.119.245.12	192.168.35.10	TCP	60	80 → 62626 [ACK] Seq=1 Ack=45986 Win=947 Len=0
483	20:58:32.578823	128.119.245.12	192.168.35.10	TCP	60	80 → 62626 [ACK] Seq=1 Ack=47446 Win=970 Len=0

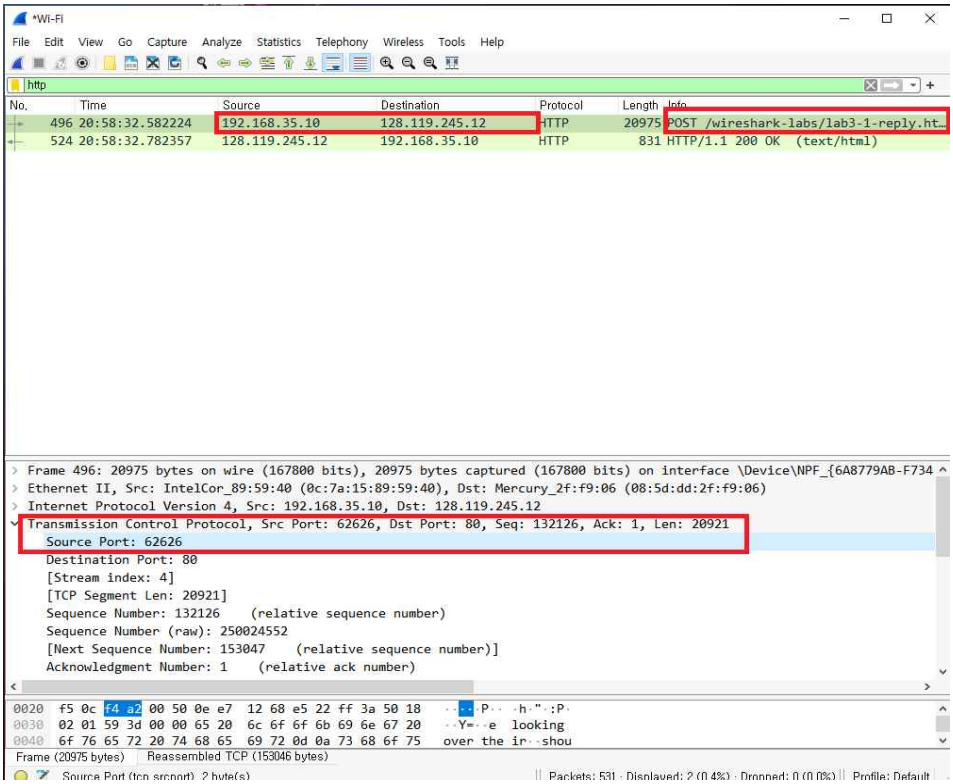
- Answer

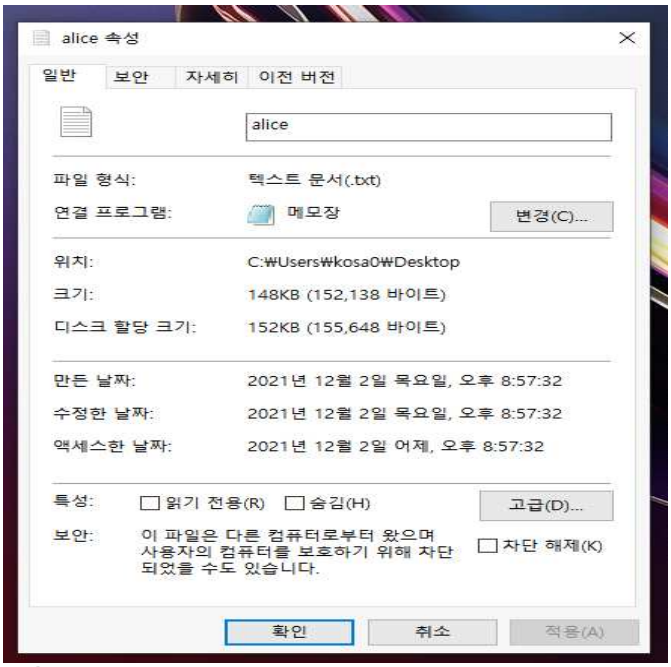
This picture shows that the ACK packet's information. Looking at the ACK information, the difference between ACK number and ACK number is 1460 and 2920(1460 x 2). For example, 2186 - 726 = 1460, 5106 - 2186 = 2920, 6566 - 5106 =1460 and so on. (I think maybe 2920 means wireshark can't catch some packetes)

So the receiver typically acknowledge 1460 bytes which is maximum data size.

1460 bytes means that maximum packet size 1500bytes - TCP/IP header size 40 bytes = 1460 bytes

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





- Answer

This picture shows that the time difference between HTTP Post and HTTP OK. That is, this time difference is the transmitting time the data 'alice txt'. alice.txt file size is 152,138 bytes

Bandwidth is the size of data transmitted per time.

Throughput is the size of data which is transmitted successfully per time.

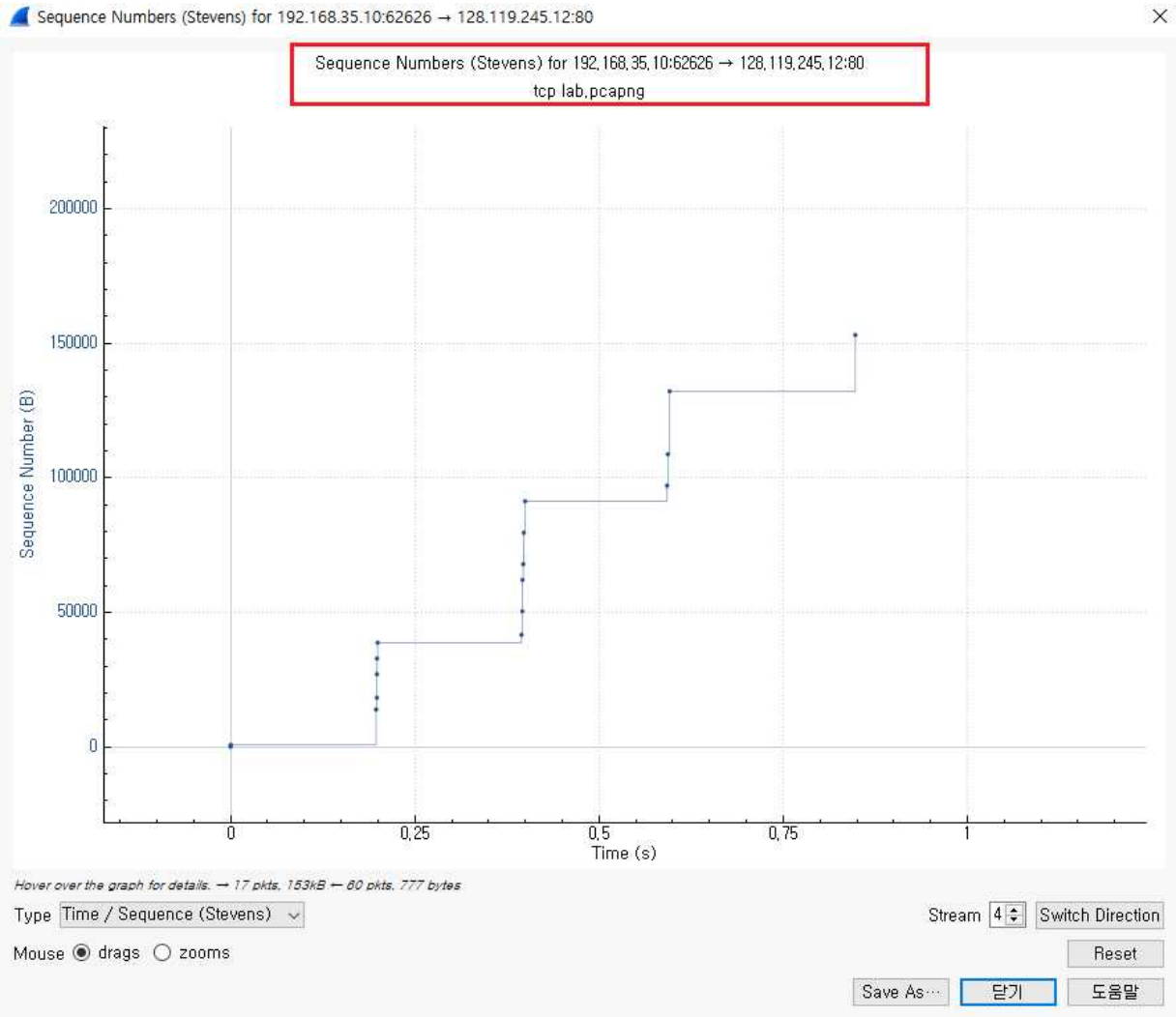
In this packet transmittion, there is no retransmitted data. So there is no fail transmitted data.

Starting time is 20:58:32.584224 and Ending time is 20:58:32.782357.

The time difference is 0.198133s.

As a conclusion 152,138 / 0.198133 = approximately 797,857.954 bytes/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

This picture shows that sequence number graph for my computer(192.168.35.10) to gaia server.

I think the TCP's slowstart phase begins and ends is each 0s and approximately 0.2 s. Because at the end of 0.2s, packet exchanges are started. And I think congestion occurred between 0.4 s and 0.6 s.. Because the rate at which the sequence number increases is slightly higher between 0.4 s and 0.6 s, but is not larger than the rest.

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

Same as problem 13's answer !