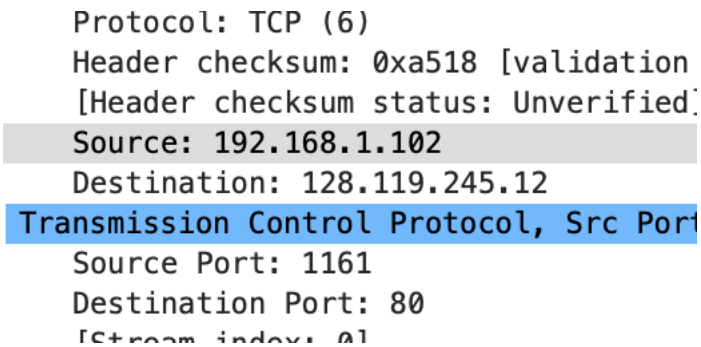
20132927_정재명_HW5

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.

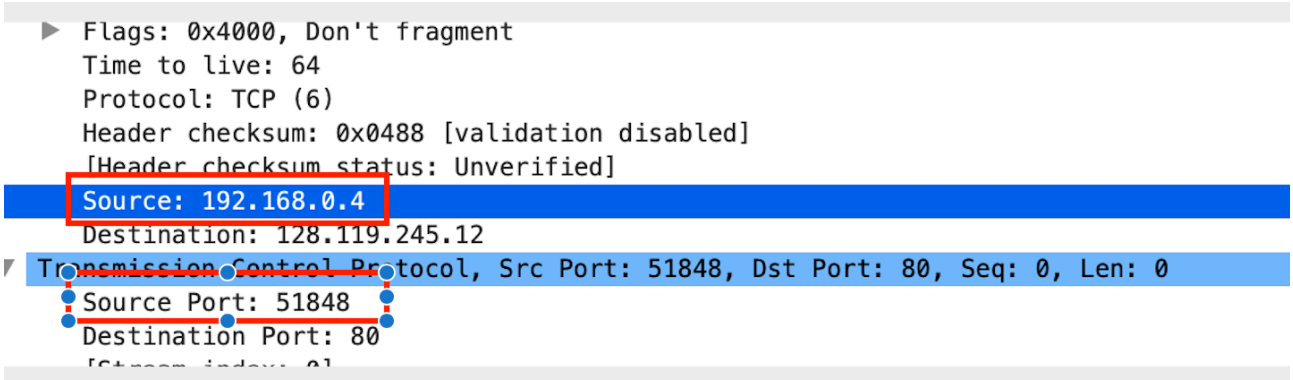


My Ip : 192.168.0.102
Port : 1161

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?

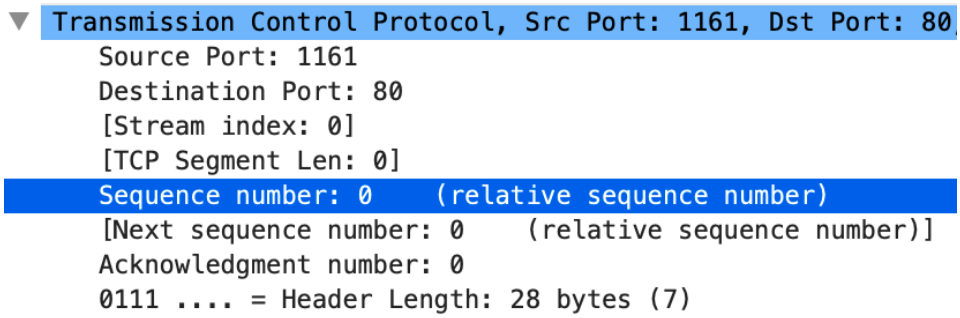
Ip : 128.119.245.12
Port : 80

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?

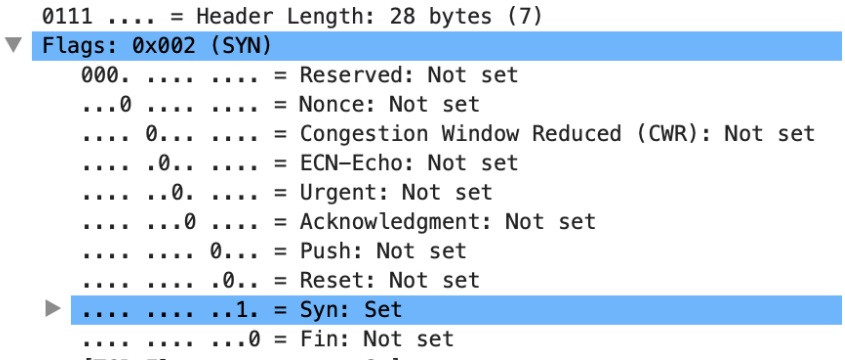


Ip : 192.168.0.4
Port : 51848

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?



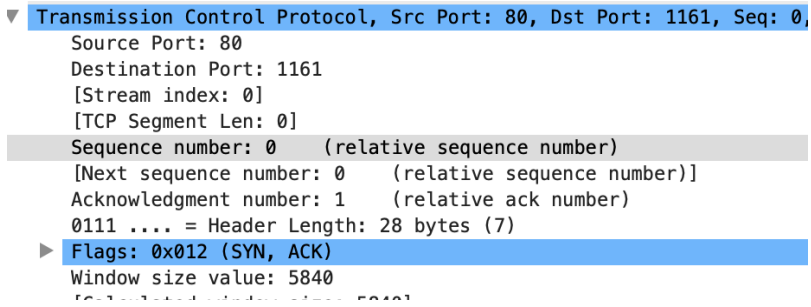
-> 0번이다.



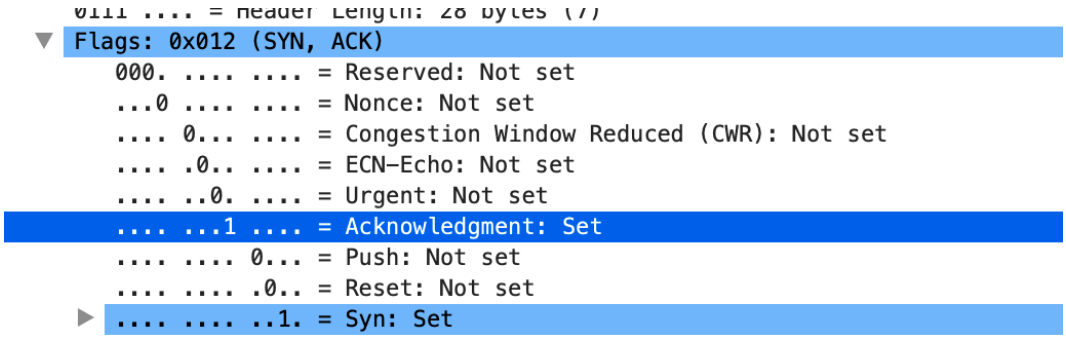
-> 세그멘트 헤더의 flags 필드에서 Syn비트가 set되어 있다.

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?

-> number of the SYNACK segment : 0이다.

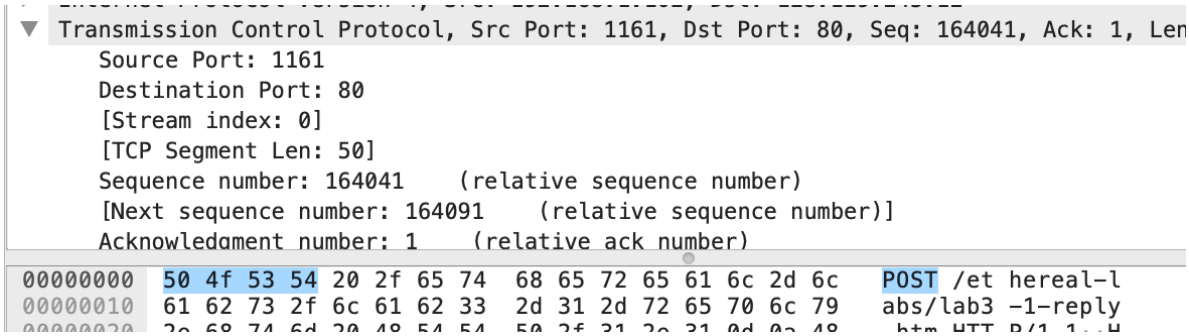


-> the value of the Acknowledgement field : 아래와 같이 1로 세팅되어 있다.



-> tcp 3 way handshaking 방식에 따라, 두번째 SYNACK 세그먼트의 ack는 첫번째로 받은 패킷의 seq넘버에 1을 더한 값을 set해서 보내게 된다.
-> 위 사진에서 Syn과 Acknowledgment 비트가 모두 Set되어 있는데, 이것이 SYNACK 세그먼트임을 알려준다.

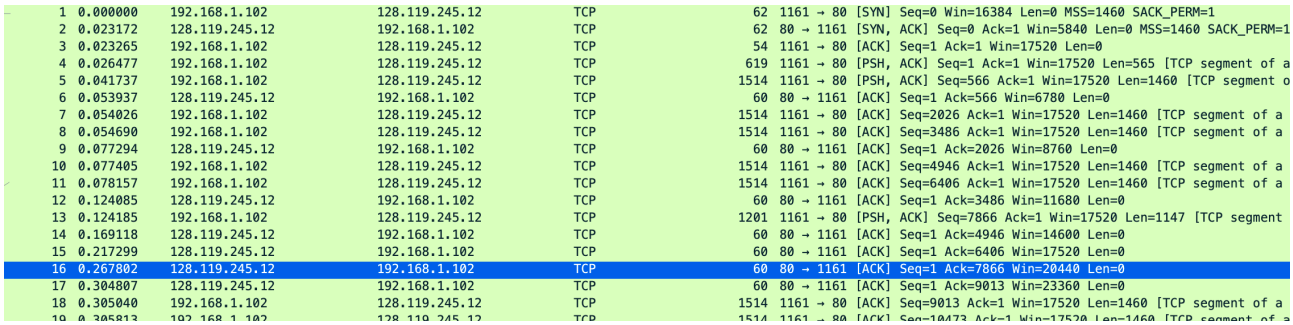
6. What is the sequence number of the TCP segment containing the HTTP POST command? Note that in order to find the POST command, you'll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with a "POST" within its DATA field.



164041

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)?

-> 1, 566, 2026, 3486, 4946, 6406



At what time was each segment sent? When was the ACK for each segment received?
-> 1(0.026477 -> 0.053937), 566(0.041737 -> 0.077293), 2026(0.054026 -> 0.124585), 3486(0.054690 -> 0.169118), 4946(0.077405 -> 0.217299), 6406(0.078157 -> 0.267802)

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?

1. 0.02746
2. 0.035556
3. 0.070559
4.0.114428
5.0.139894
6.0.189645

What is the `EstimatedRTT` value (see Section 3.5.3, page 242 in text) after the receipt of each ACK?

1. 0.02746
2. 0.028472
3. 0.033732
4. 0.043819
5. 0.055828
6. 0.069763

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*.

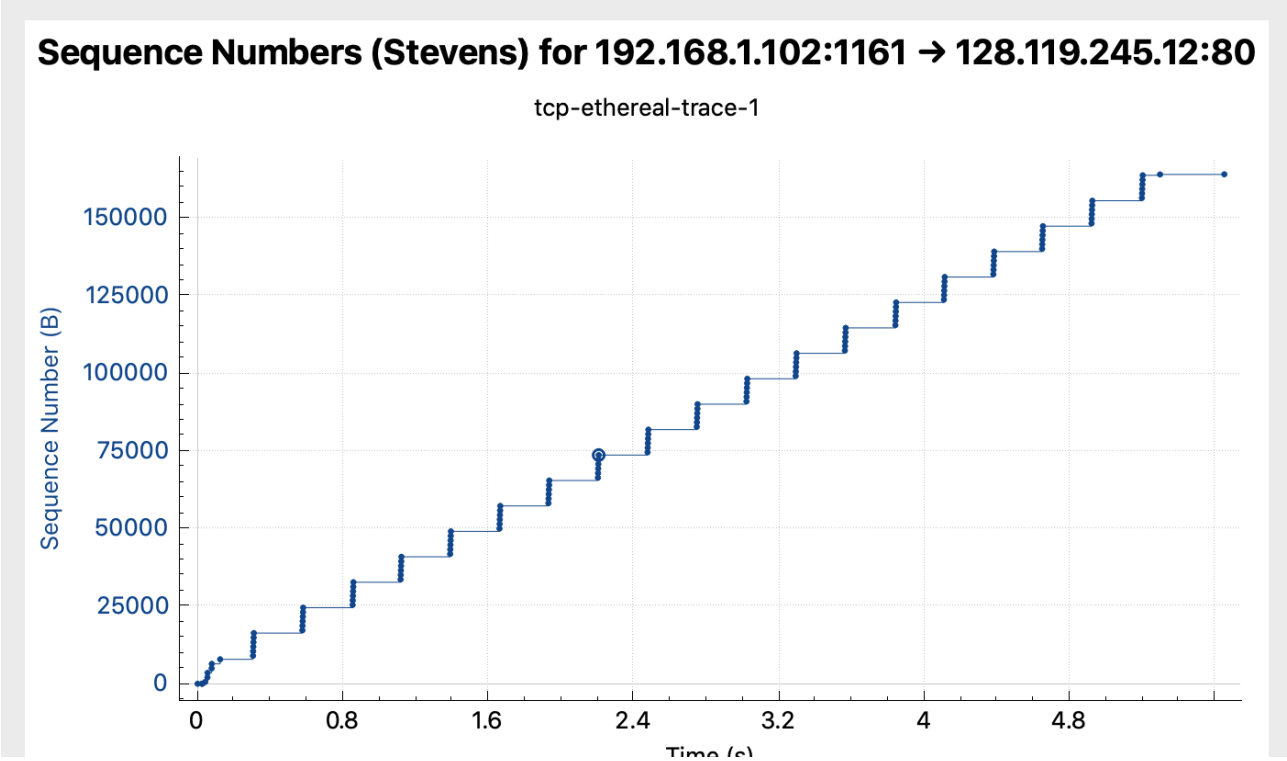
8.What is the length of each of the first six TCP segments?
-> 565, 1460, 1460, 1460, 1460, 1460

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?

-> 5840

```
    62  1161 → 80 [SYN] Seq=0 Win=16384 Len=0 MSS=146
    62  80 → 1161 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len
    54  1161 → 80 [ACK] Seq=1 Ack=1 Win=17520 Len=0
   619  1161 → 80 [PSH, ACK] Seq=1 Ack=1 Win=17520 Le
  1514  1161 → 80 [PSH, ACK] Seq=566 Ack=1 Win=17520
```

10. Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question?
-> 재전송되었다는 것은 윈도우의 크기가 줄어들었다는 것을 의미한다. 윈도우가 감소한 위치가 재전송된 곳이라고 볼 수 있다. 다음 표를 보면, 타임아웃에 의해 재전송 된 구간을 확인할 수 있다.



11. How much data does the receiver typically acknowledge in an ACK?

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 a re
5	0.041737	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [PSH, ACK] Seq=566 Ack=1 Win=17520 Len=1460 [TCP segment of a
6	0.053937	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=566 Win=6780 Len=0
7	0.054026	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=2026 Ack=1 Win=17520 Len=1460 [TCP segment of a rea

-> 처음에는 565바이트로 시작하고 이후 쭉 1460바이트의 데이터를 보낸다. 마지막에는 272바이트를 보낸다.

190	5.125019	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=154117 Win=62780 Len=0
191	5.197286	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=156469 Win=62780 Len=0
192	5.197508	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=156469 Ack=1 Win=17520 Len=1460 [TCP segr
193	5.198388	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=157929 Ack=1 Win=17520 Len=1460 [TCP segr
194	5.199275	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=159389 Ack=1 Win=17520 Len=1460 [TCP segr
195	5.200252	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=160849 Ack=1 Win=17520 Len=1460 [TCP segr
196	5.201150	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=162309 Ack=1 Win=17520 Len=1460 [TCP segr
197	5.202024	192.168.1.102	128.119.245.12	TCP	326	1161 → 80 [PSH, ACK] Seq=163769 Ack=1 Win=17520 Len=272 [TCP

Can you identify cases where the receiver is ACKing every other received segment (see Table 3.2 on page 250 in the text).

-> 찾아볼 수 없다.

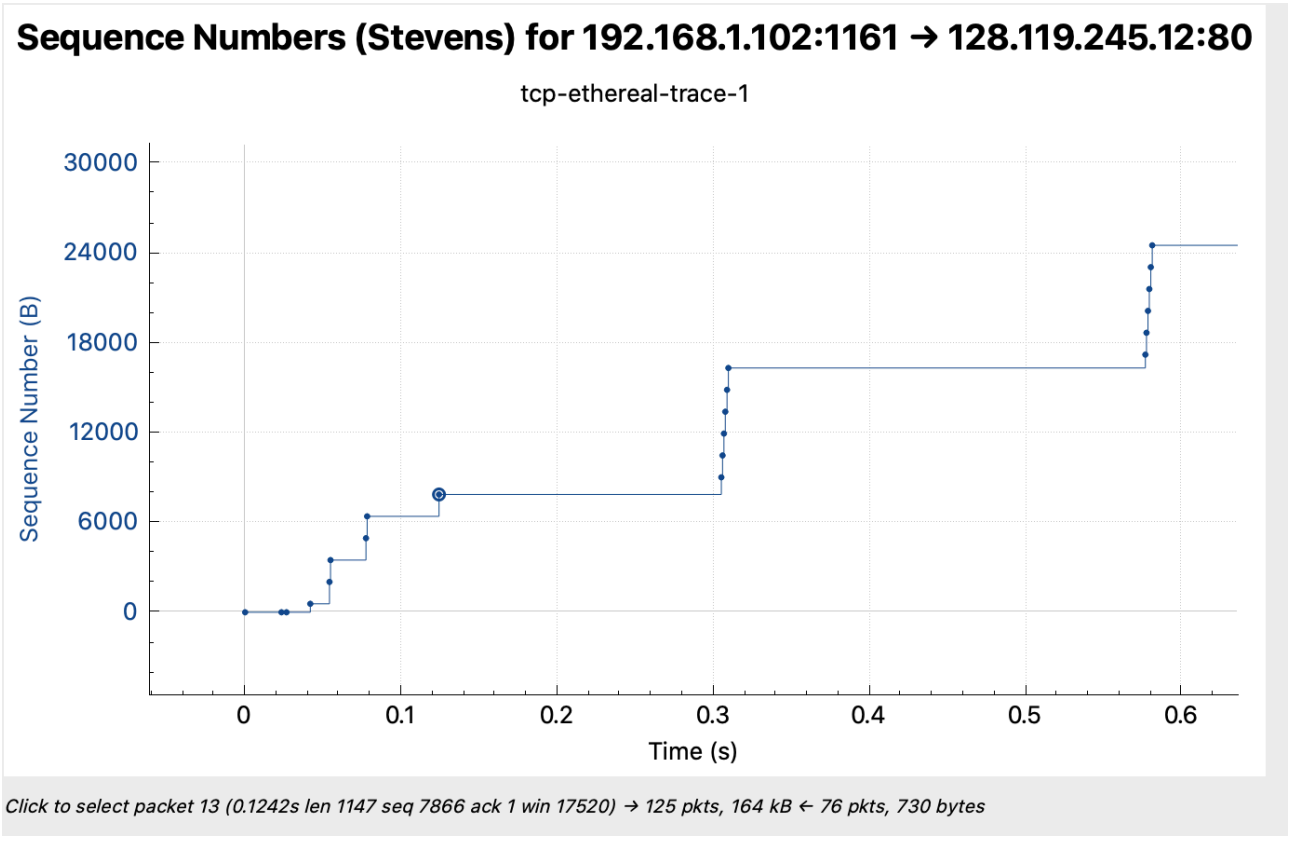
12. What is the throughput (bytes transferred per unit time) for the TCP connection?
Explain how you calculated this value.

-> 단위 시간당 바이트 처리량을 구하면 된다. 총 164091바이트의 데이터가 전송되었다. 첫번째 데이터가 도착한 시간은 0.023172, 마지막은 5.45830이다. 이 시간의 차이는 5.435128초이다.
답 : 164091 / 5.435128 byte/sec

212	7.103780	192.168.1.100	192.168.1.1	SSDP	175	M-SEARCH * HTTP/1.1
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
6	0.053937	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=566 Win=6780 Len=0
9	0.077294	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=2026 Win=8760 Len=0
12	0.124085	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=3486 Win=11680 Len=0
14	0.169118	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=4946 Win=14600 Len=0

186	5.019189	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=151197 Win=62780 Len=0
190	5.125019	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=154117 Win=62780 Len=0
191	5.197286	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=156469 Win=62780 Len=0
198	5.297257	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=159389 Win=62780 Len=0
200	5.389471	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=162309 Win=62780 Len=0
201	5.447887	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=164041 Win=62780 Len=0
202	5.455830	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=164091 Win=62780 Len=0
203	5.461175	128.119.245.12	192.168.1.102	HTTP	784	HTTP/1.1 200 OK (text/html)
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_PE

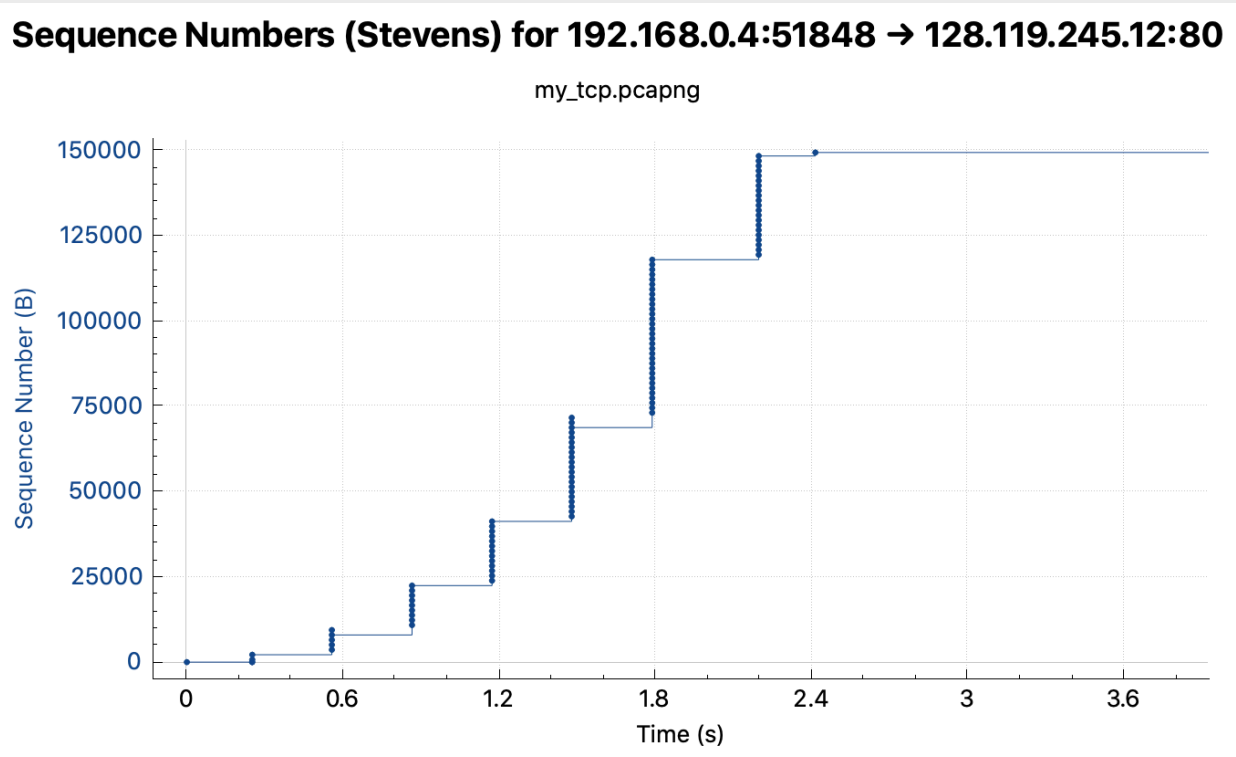
13. Use the *Time-Sequence-Graph(Stevens)* plotting tool to view the sequence number versus time plot of segments being sent from the client to the gaia.cs.umass.edu server. Can you identify where TCP's slowstart phase begins and ends, and where congestion avoidance takes over?



-> 그림을 보면 0.3초 이후부터는 동일한 윈도우 크기를 가지고 있고 그 이전까지 윈도우 크기의 지수적인 성장이 나타나고 있음을 알 수 있다. 그림에서 처음 두 개의 패킷은 3 ways handshaking을 위한 패킷이므로, 슬로우 스타트의 시작은 그림에서 세번째 패킷 부터다. 그리고 0.3초 이후의 윈도우 크기가 계속 같게 유지되고 있는데, 이것만 보고 슬로우 스타트가 끝났다고 할 수는 없다. 그래서 슬로우 스타트의 끝은 알 수 없다.

Comment on ways in which the measured data differs from the idealized behavior of TCP that we've studied in the text.
-> 위에서 언급했듯이, 0.3초 이후에 계속 동일한 윈도우 사이즈를 가지고 있다. 이것은 책에서 공부했던 슬로우 스타트, 혼잡 회피, 빠른 회복 그 어디에도 속하지 않은 경우다.

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



-> 처음 시작부터 계속 윈도우 크기가 2배수로 커지고 있다. 약 2.2초 이후로는 패킷이 더이상 전송되지 않아 그 이후에 다른 상태로 빠지게 될지는 알 수 없다. 그래서 끝점은 이 그림만으로는 알 수가 없다.
-> 딱히 이상한 점이 보이지는 않는다.