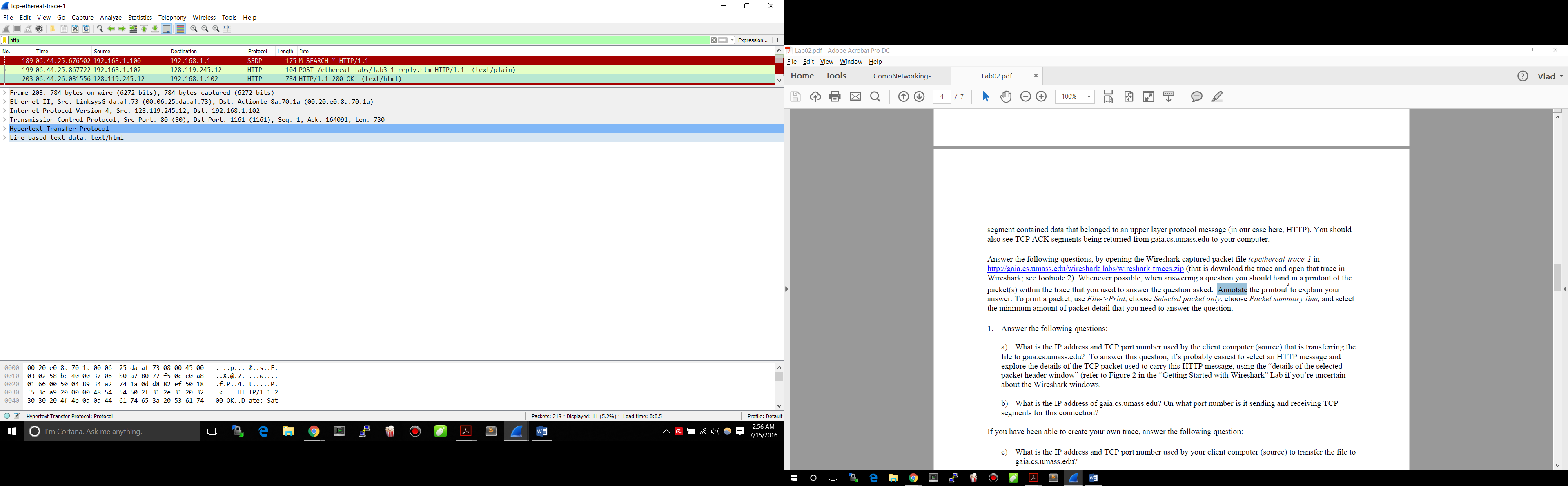
Vlad Predovic

Due 5/24/2016

CS 372

Lab 2

1. Answer the following questions:

a) 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. 

IP Address: 192.168.1.102 Port Number: 1161

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

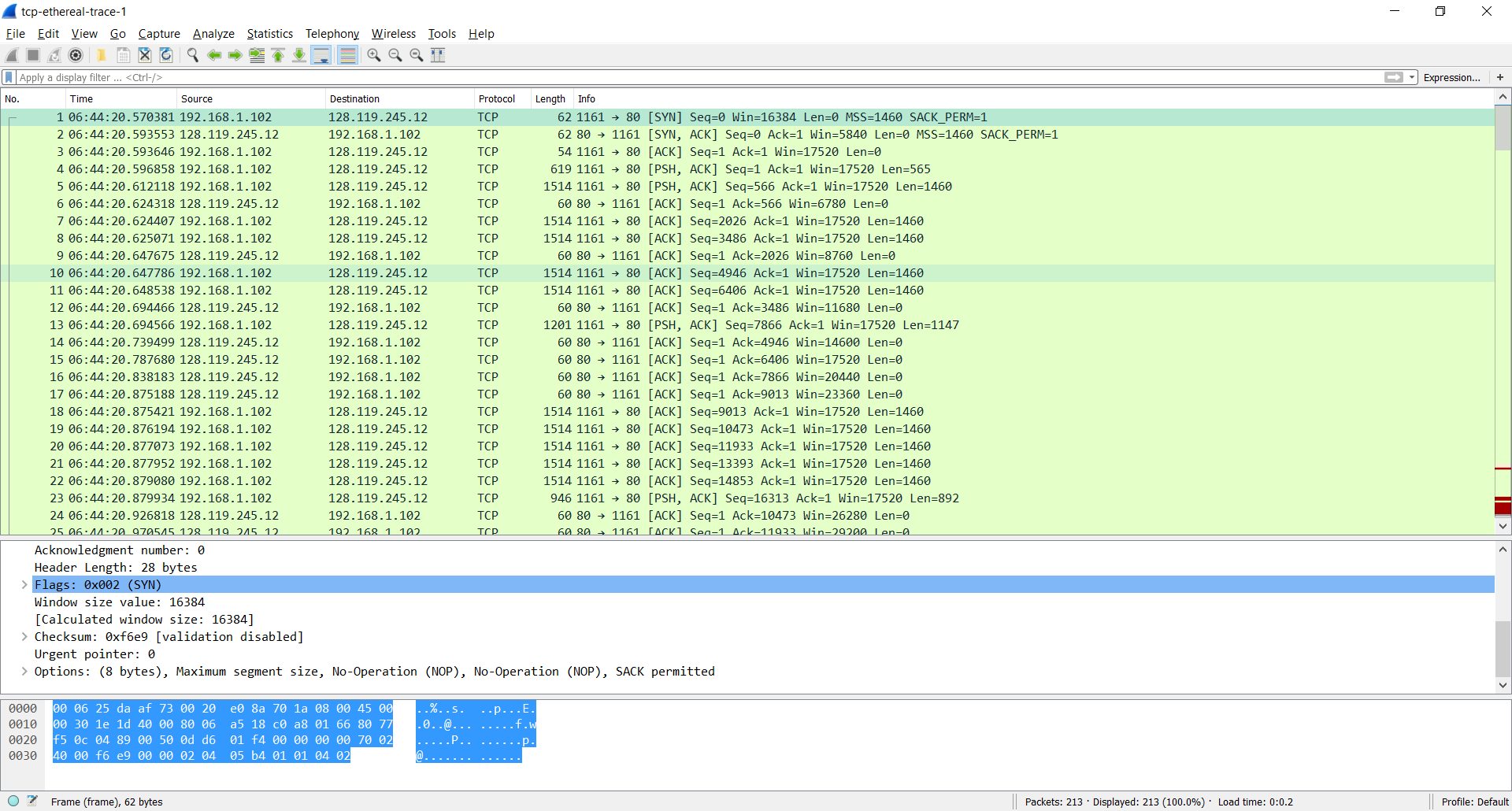
IP Address: 128.119.245.12 Port Number: 80

If you have been able to create your own trace, answer the following question:

c) 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 Address: 10.0.0.244 Port Number: 55847

2. 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?

The sequence number is 0. In wireshark, the SYN flag is set raised (set to 1), identifying it as a SYN segment. 

3. 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?

The sequence number is set to 0. The aacknowledgement field is set to 1. This is because the acknowledgement field is set to the client\_isn + 1. In this case the client\_isn(sequence number) was 0.

The SYN and ACK flags are both raised. This is why it constitutes as the SYNACK segment.

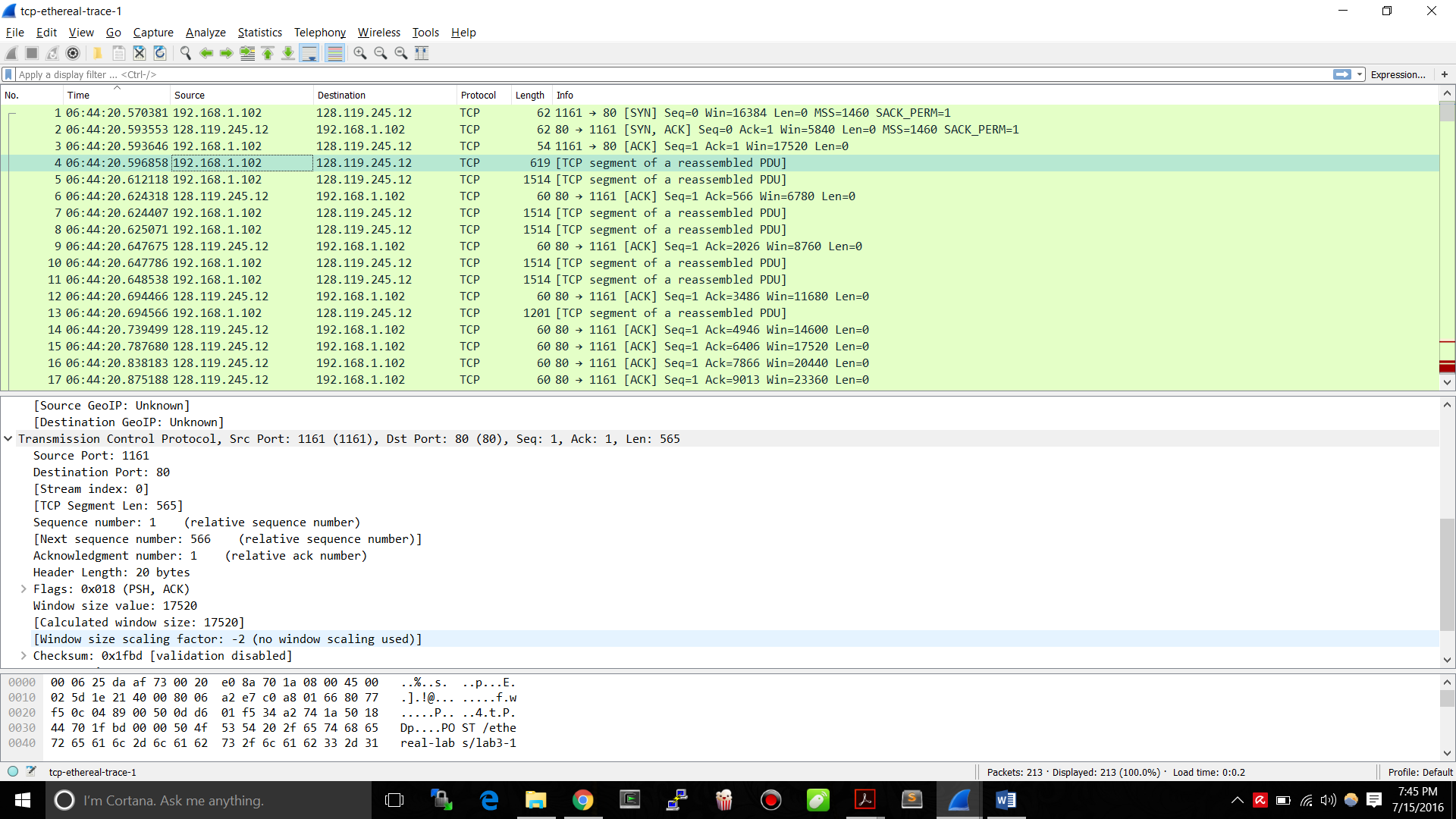
4. What is the sequence number of the TCP segment containing the HTTP POST command? Note that in order to find the POST command, you’ll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with a “POST” within its DATA field.

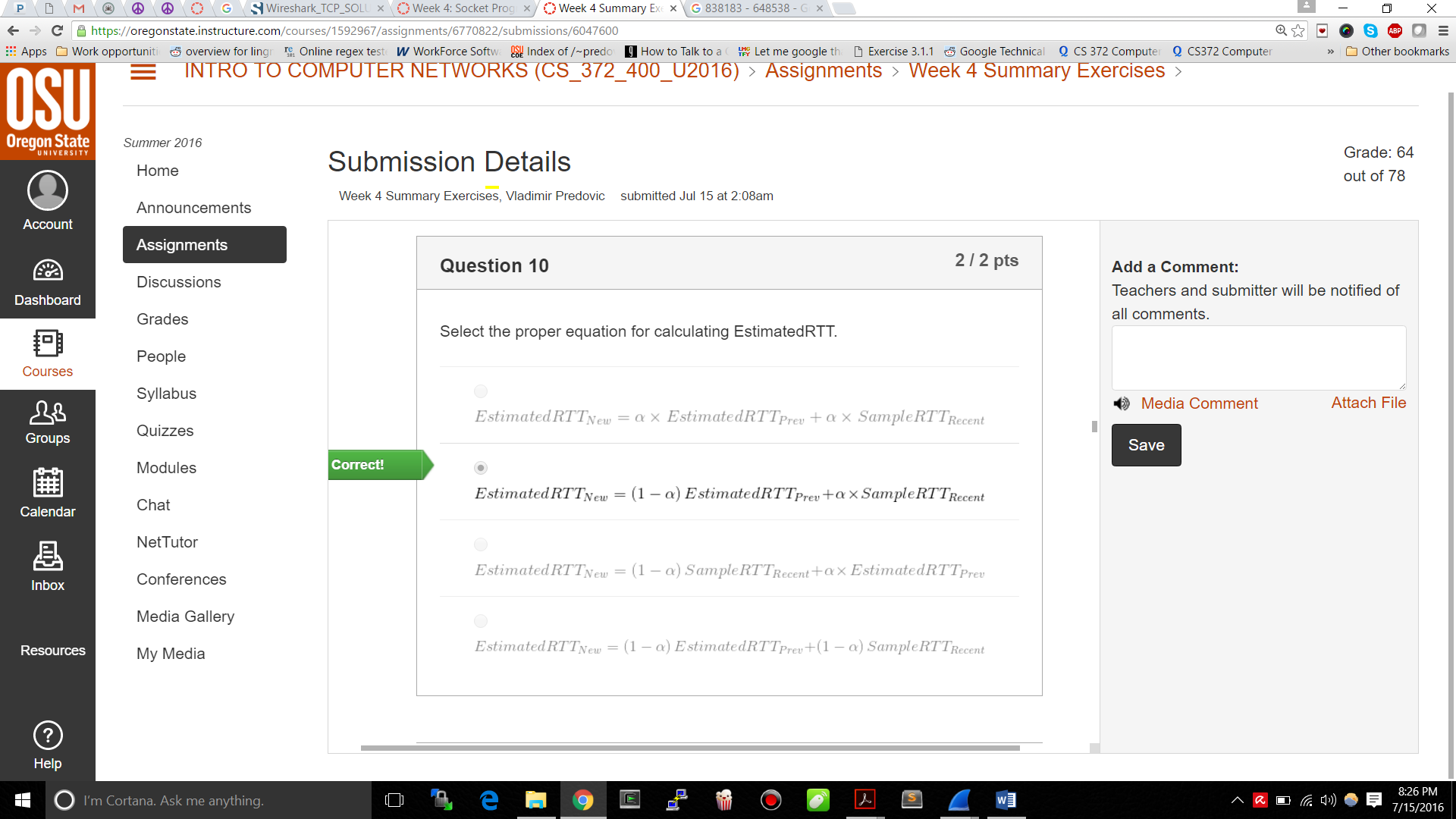
Sequence Number: 1



5.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Sent Time | Time ACK  received | RTT  seconds | Est RTT (after ACK rcpt)  In seconds |
| 1) Seq #: 1 | 06:44:20.596858 | 06:44:20.624318 | 0.02746 | 0.02746 |
| 2) Seq #: 566 | 06:44:20.612118 | 06:44:20.647675 | 0.035557 | 0.0285 |
| 3) Seq #: 2026 | 06:44:20.624407 | 06:44:20.694466 | 0.070059 | 0.0337 |
| 4) Seq #: 3486 | 06:44:20.625071 | 06:44:20.739499 | 0.114428 | 0.0438 |
| 5) Seq #: 4946 | 06:44:20.647786 | 06:44:20.787680 | 0.139894 | 0.0558 |
| 6) Seq #: 6406 | 06:44:20.648538 | 06:44:20.838183 | 0.189645 | 0.0725 |





Assuming 1st estRTT = 1st sampleRTT α = .125

EstimatedRTT1: 0.875 \* .02746 + 0.125 \* .02746 = .02746

EstimatedRTT 2: 0.875 \* .02746+ 0.125 \* 0.035557 = 0.0285

EstimatedRTT 3: 0.875 \* 0.035557+ 0.125 \* 0.070059 = 0.0337

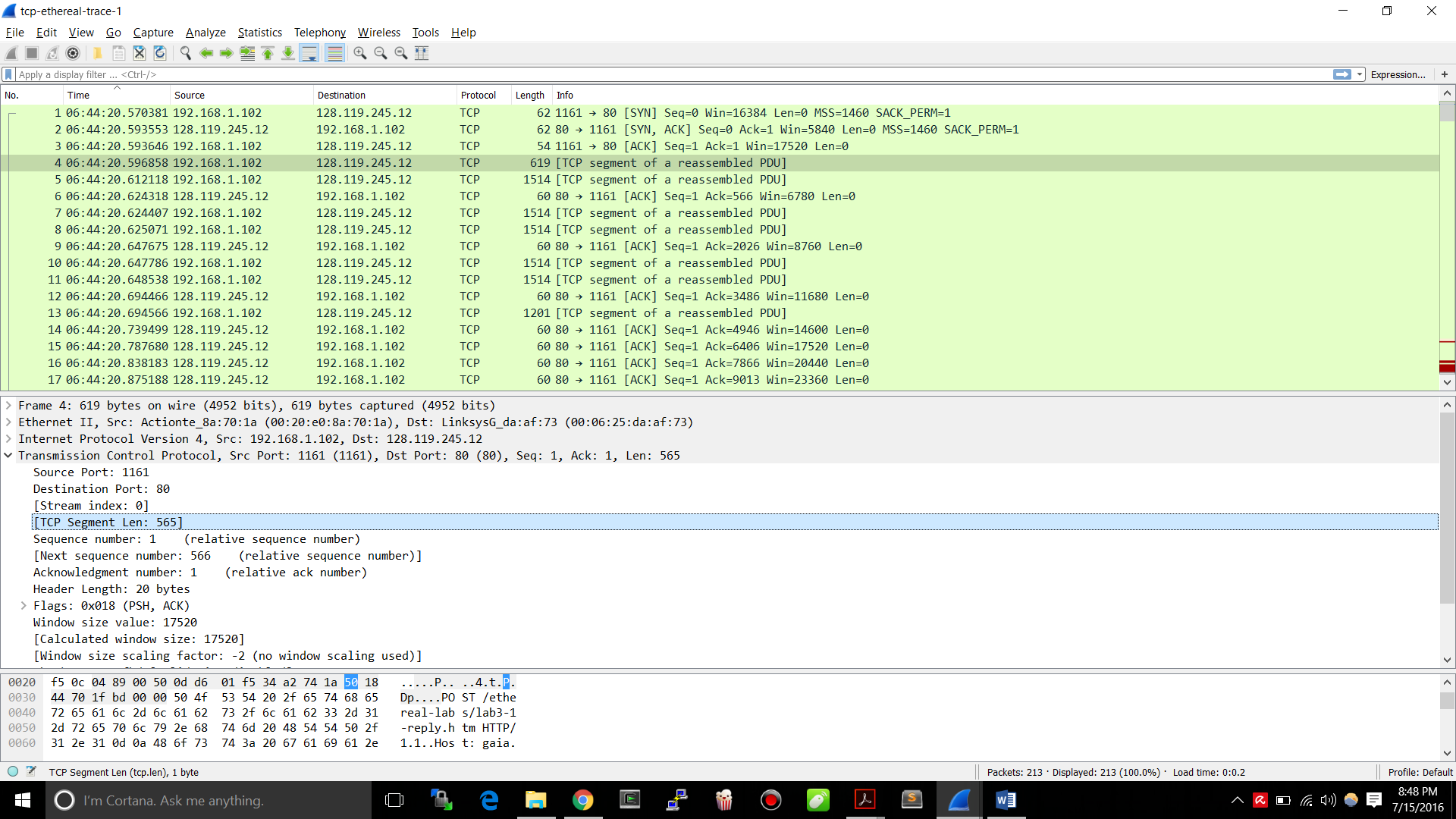
EstimatedRTT 4: 0.875 \* 0.070059+ 0.125 \* 0.114428 = 0.0438

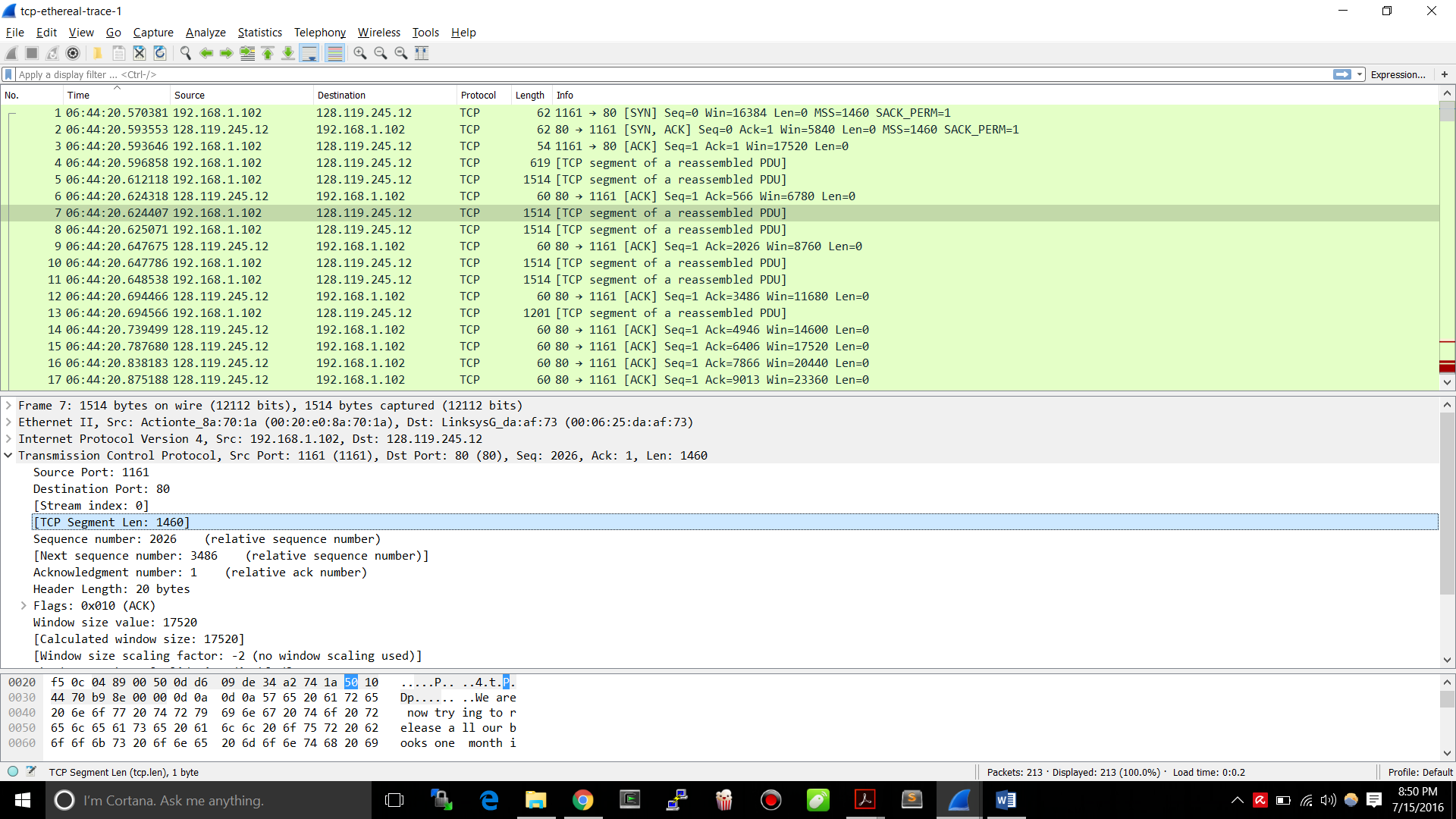
EstimatedRTT 5: 0.875 \* 0.114428+ 0.125 \* 0.139894 = 0.0558

EstimatedRTT 6: 0.875 \* 0.139894+ 0.125 \* 0.189645 = 0.0725

6. What is the length of each of the first six TCP segments?4

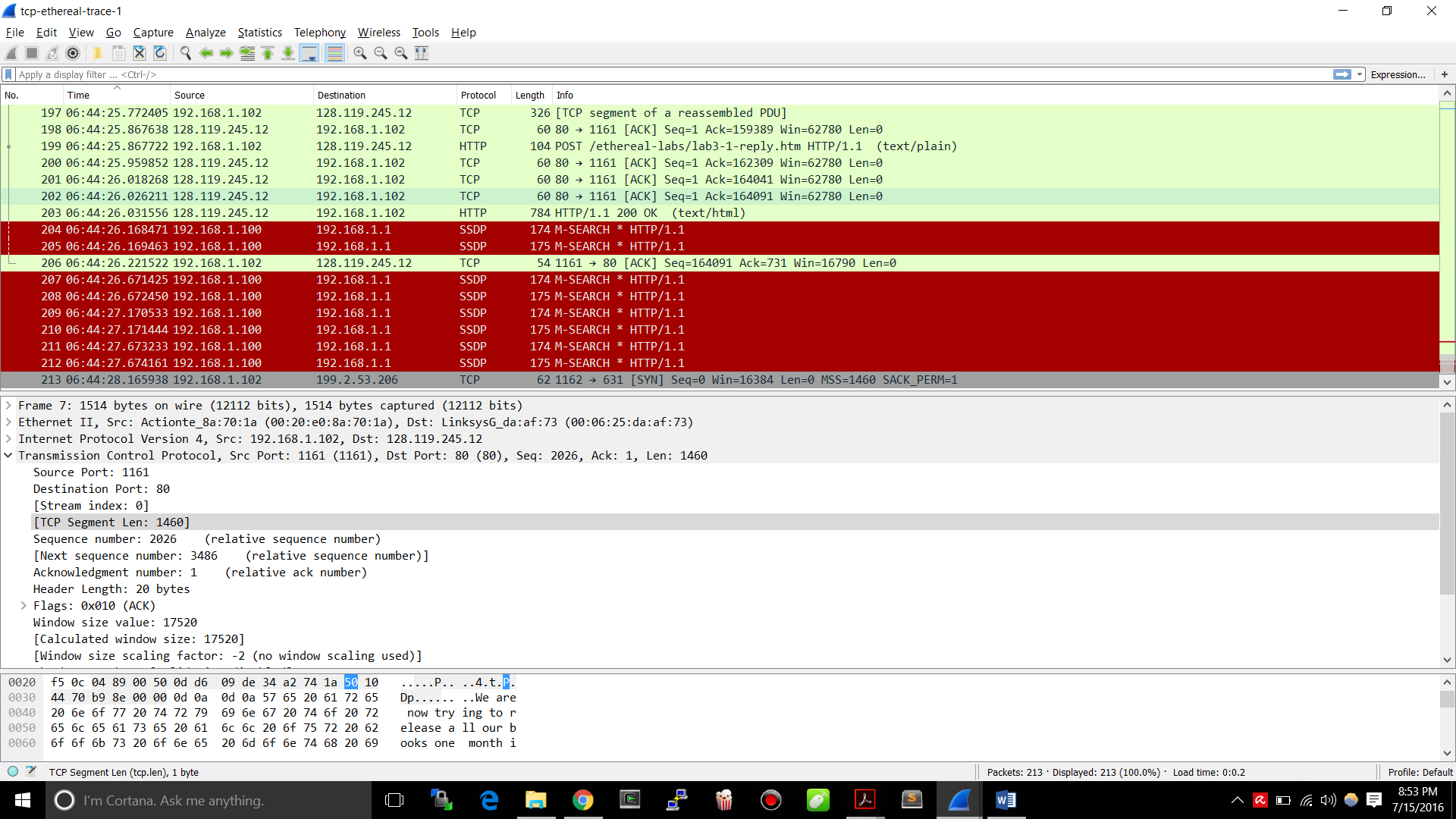
1: 565 bytes 2 through 6: 1460 bytes





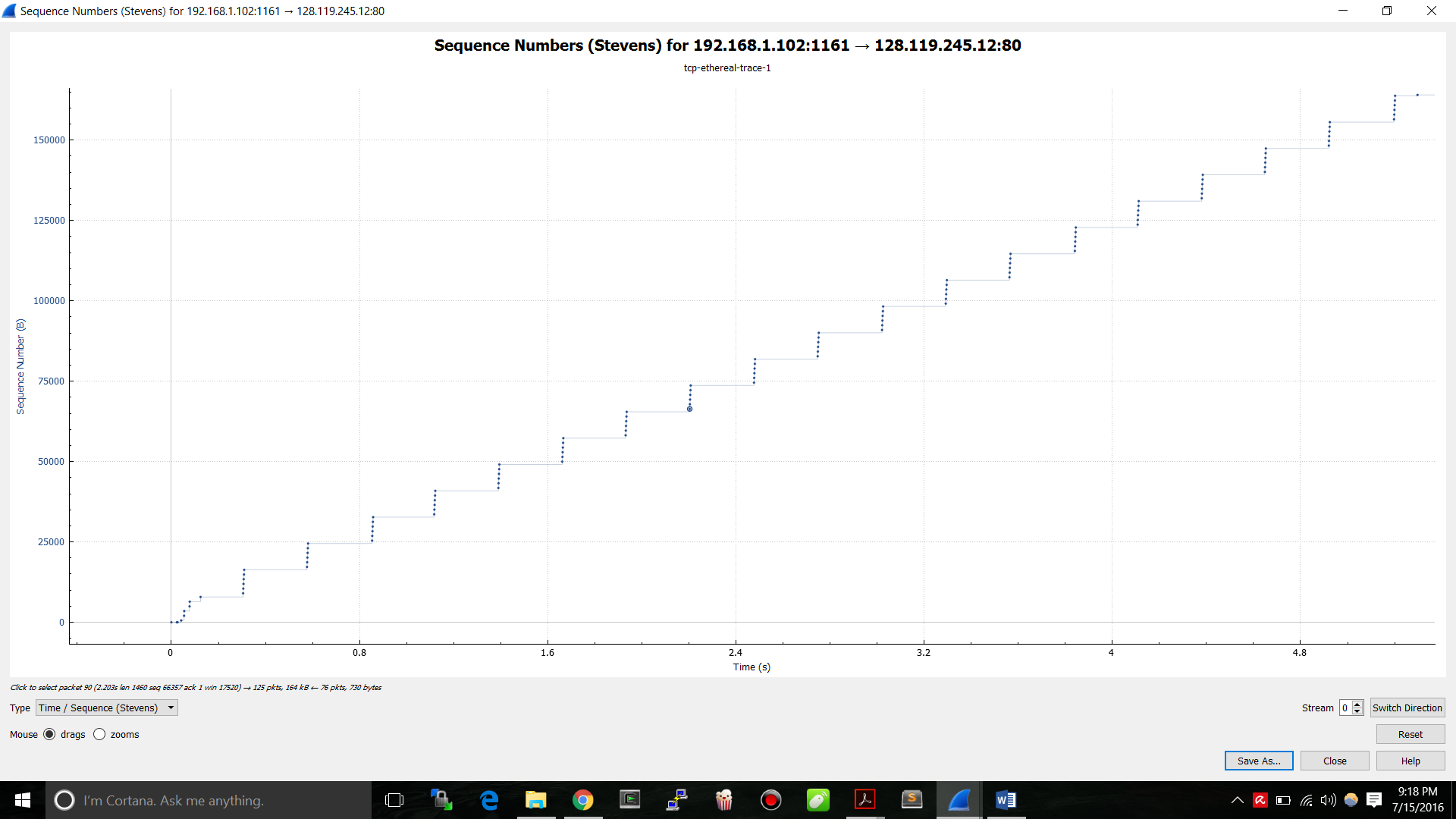
7. What is the minimum amount of available buffer space advertised at the received for the entire trace? Does the lack of receiver buffer space ever throttle the sender?

Min amount of buffer space is 5840. Can be seen in the first picture at the top right of question 6. Increases to 62780 for the entinre trace. Sender is never throttled.



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

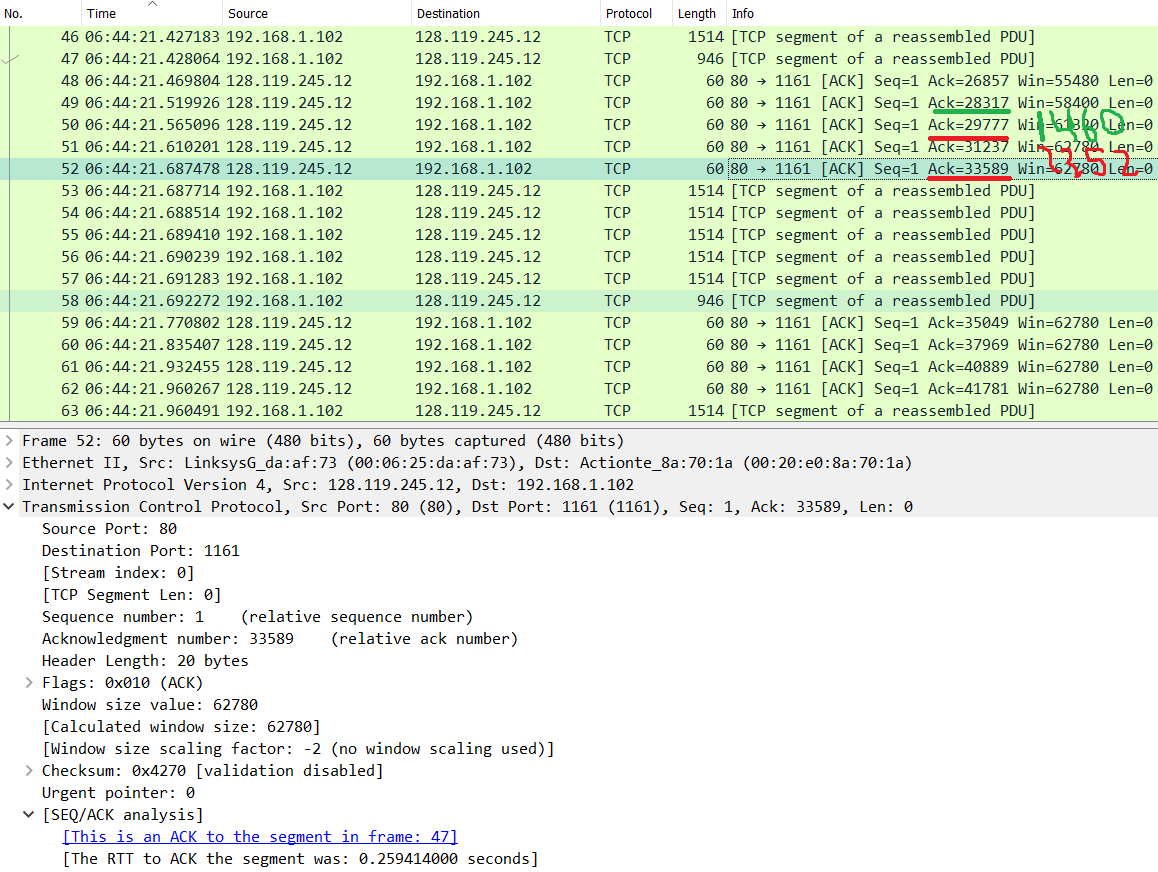
No, I checked the datils of the HTTP segment and also the sequence numbers. If a segment was retransmitted (due to getting lost for example), it’s sequence number woud be lower than the ones before and after it (due to arriving out of order). In the following graph, you can clearly tell because all the sequence numbers increase with respect to time.



9. 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 247 in the text)?

In the first segment sent 566 bytes are acknowledged. For most of the others directly responded to 1460 bytes are acknowledged up until item 47.

However, at item number 47 the behaviour seems to change as the acknowledgements now come once every two packets and the ACK numbers increment by 2352 as opposed to 1460. This trend continues until the end of transmission.



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

The last ACK represents the next byte number expected. The first byte had sequence number 1. Therefore 164090 bytes were sent during the time period the sniffer ran. Dividing this by the amount of time ran would give the result. The time ran is calculated by subtracting the time of this last packet from the first one (item 4 from item 206). 6:44:26.221522 – 6:44:20.596858 = 5.624664 seconds

As a result: 164090 bytes / 5.624664 sec = 29173.298 = 29.17 KB/sec

