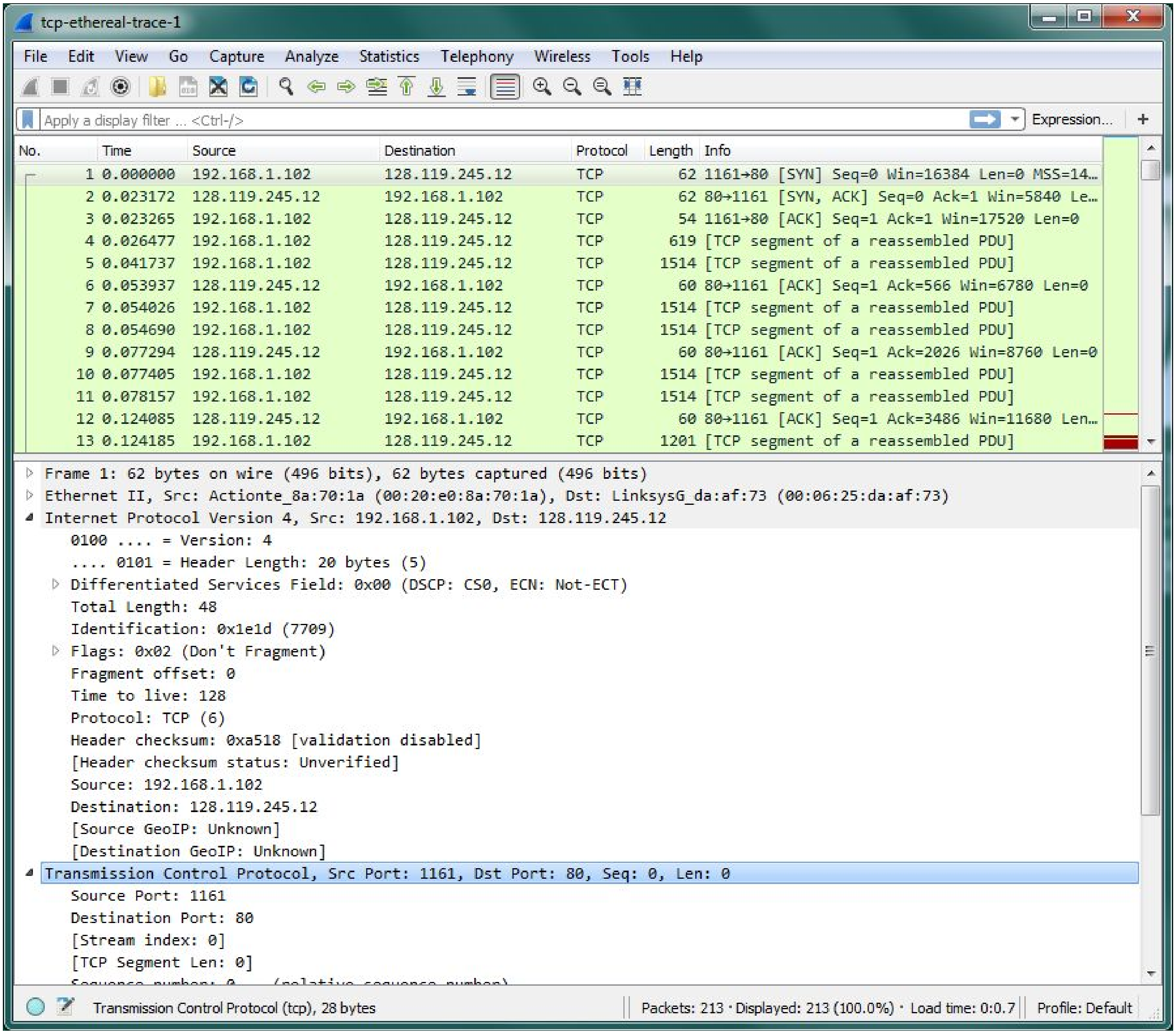
Sterling Price - U49743231
Matthew Kramer - U20891900
CNT 4004.001 - Wireshark Lab 4



Source and destination IP addresses and TCP port numbers

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?

The client computer has the IP address **`192.168.1.102`** and TCP port number **`1161`**.

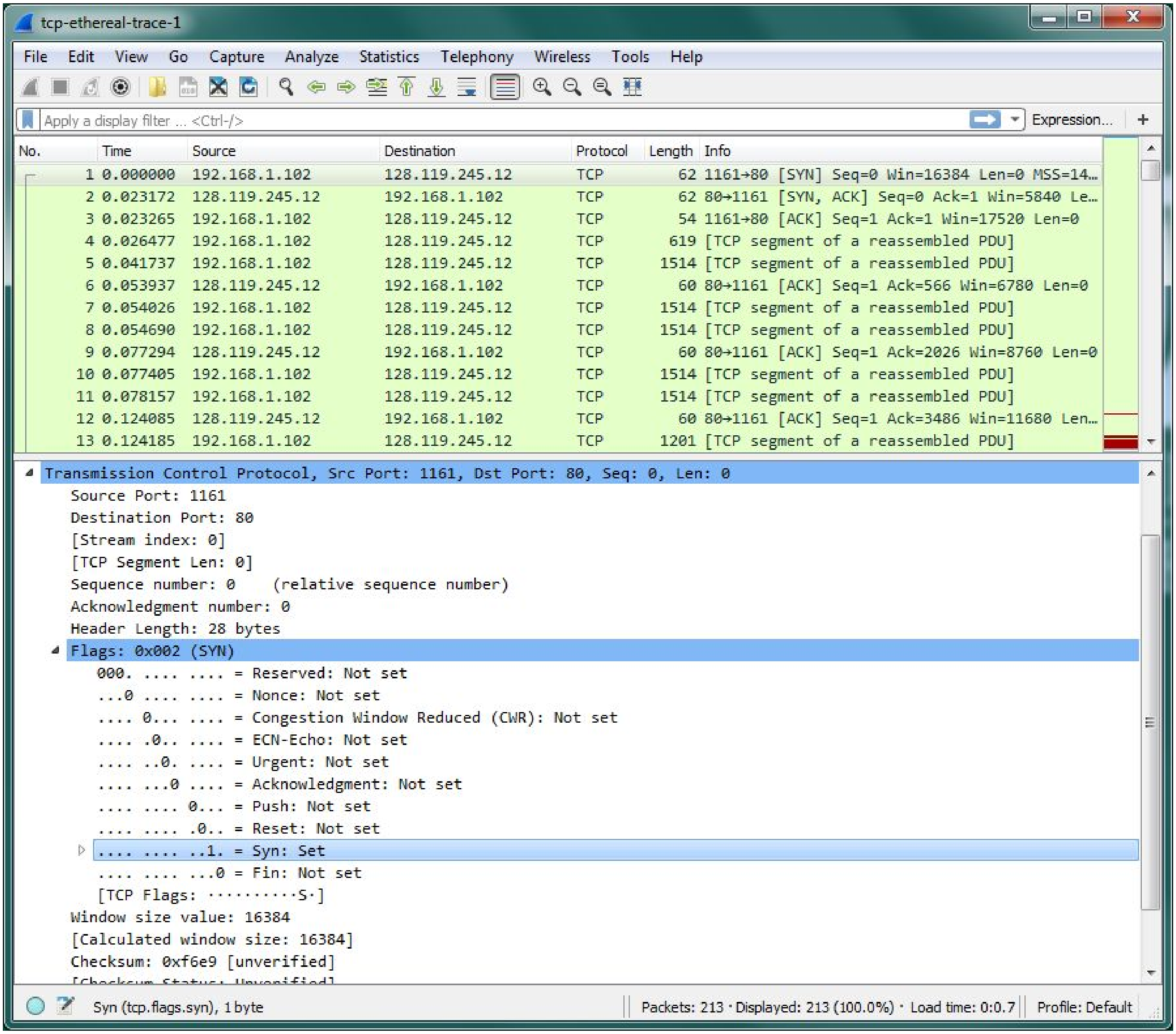
2. What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection?

The gaia.cs.umass.edu server has the IP address **`128.119.245.12`** and TCP port number **`80`**.

Sterling Price - U49743231
Matthew Kramer - U20891900
CNT 4004.001 - Wireshark Lab 4

3. If you have been able to create your own trace, what is the IP address and TCP port number used by your client computer (source) to transfer the file to gaia.cs.umass.edu?

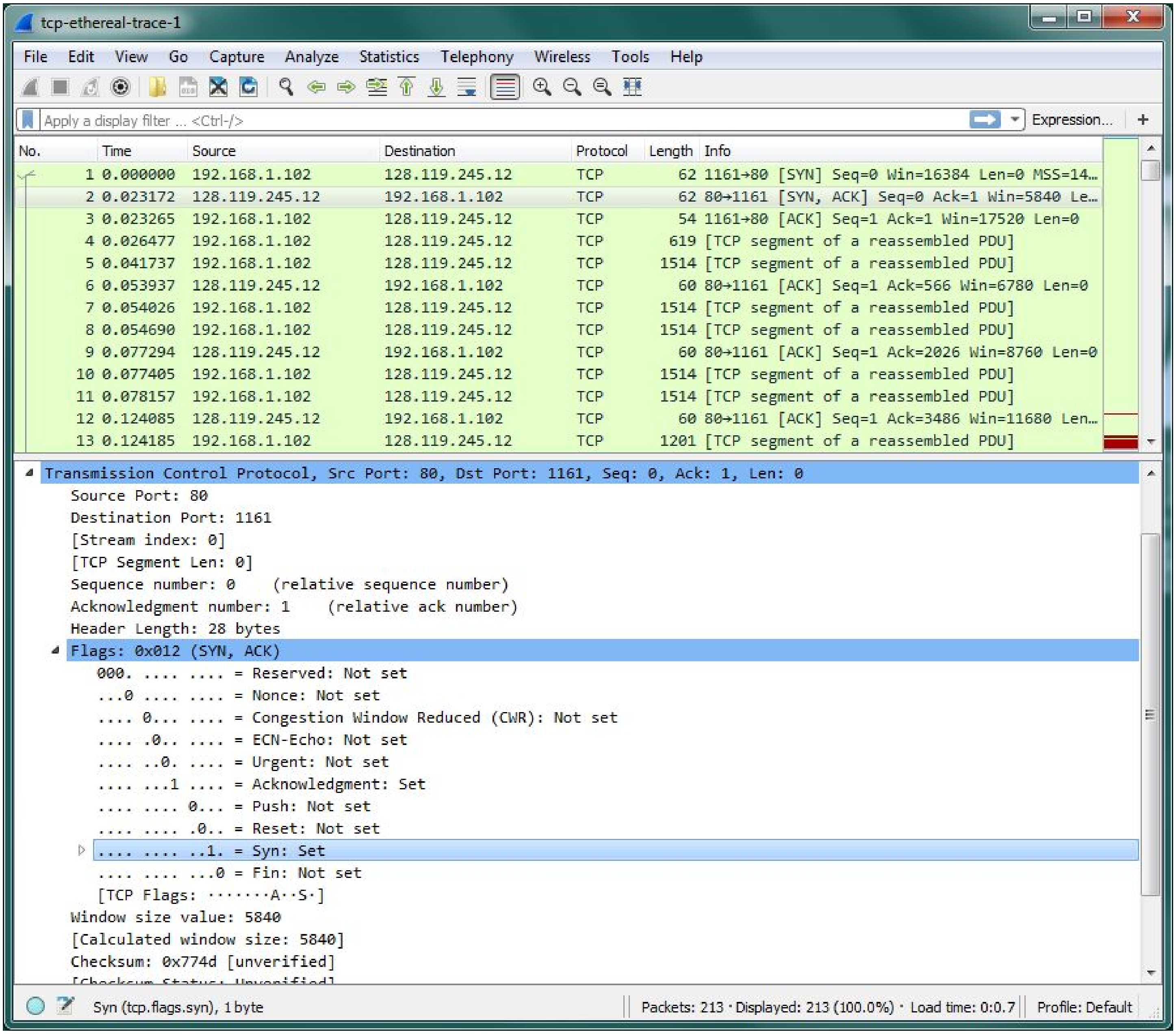
We have used the trace found at http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip.



TCP SYN segment

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?

The sequence number **`0`** is used to initiate the TCP connection between the client and server. The **SYN** flag is set (value of **1**), indicating that this is a **SYN** segment.

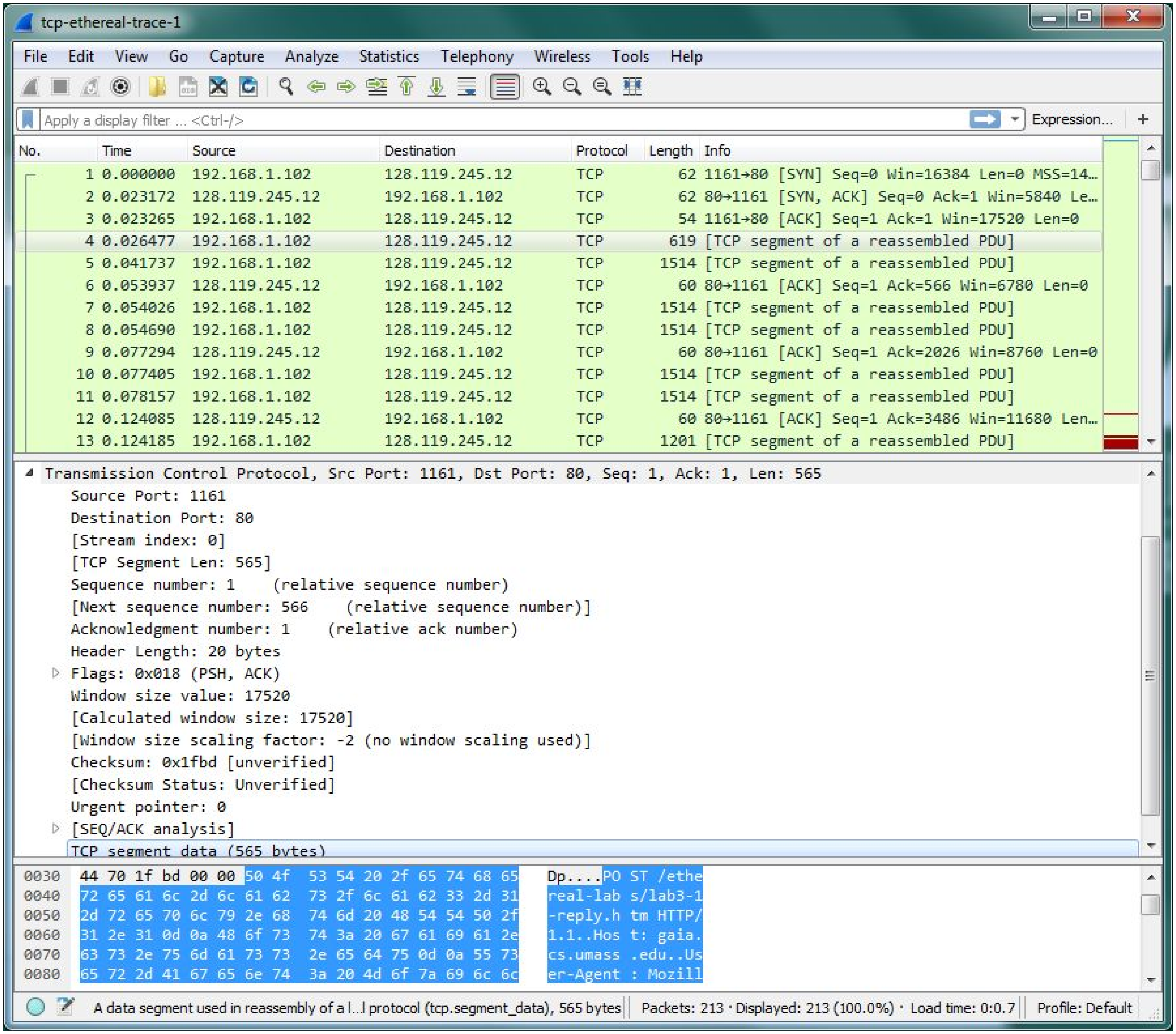


TCP SYNACK segment

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?

> The **SYNACK** segment has sequence number **`0`** (like the **SYN** segment above) and an acknowledgement value of **`1`**. This value is determined by incrementing the sequence number from the **SYN** segment. The **SYN** and **`Acknowledgement`** flags identify this as a **SYNACK** segment.

Sterling Price - U49743231
Matthew Kramer - U20891900
CNT 4004.001 - Wireshark Lab 4



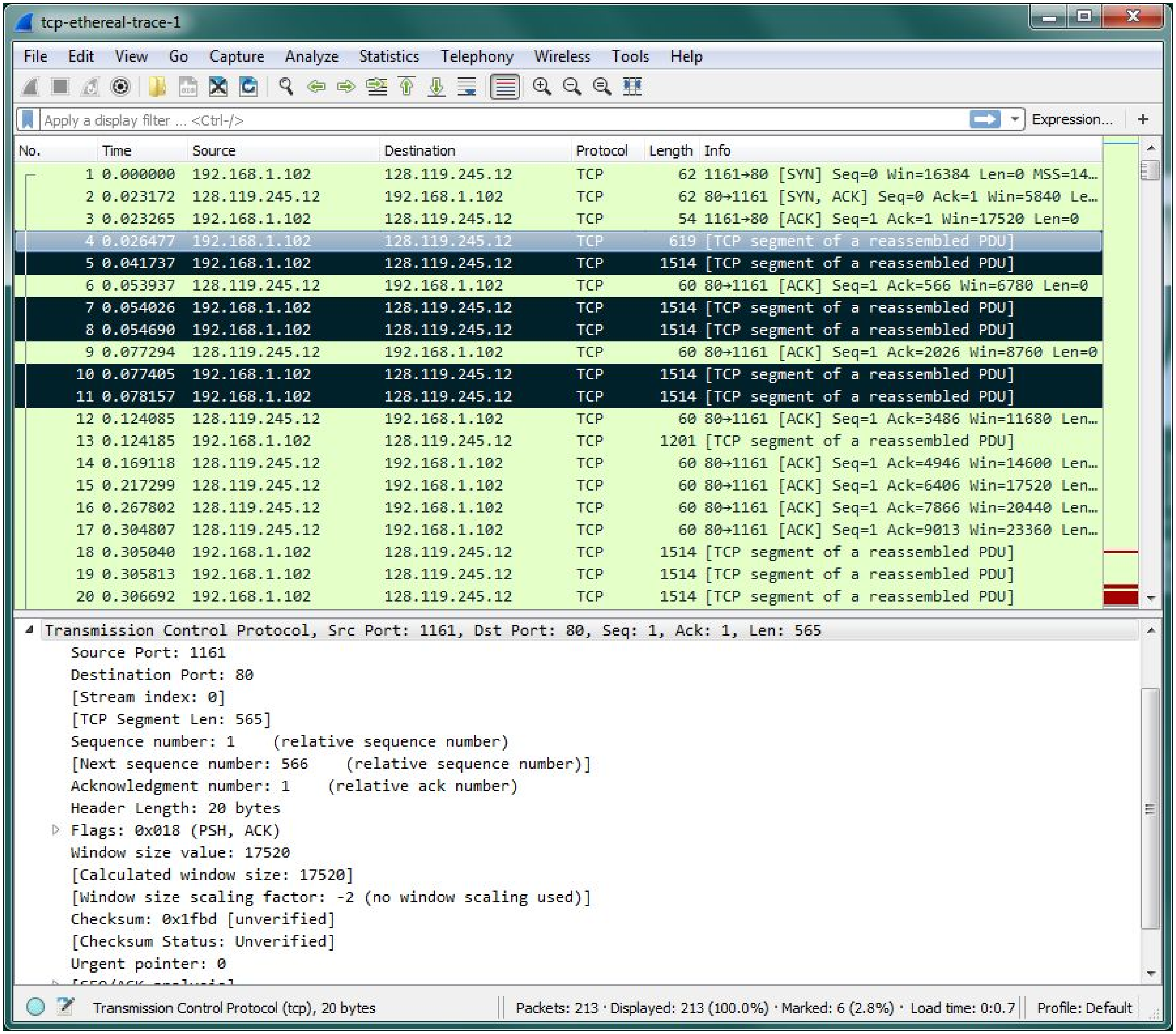
Segment containing HTTP POST command

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.

The TCP segment containing the HTTP POST command has sequence number **1**.

Sterling Price - U49743231
Matthew Kramer - U20891900

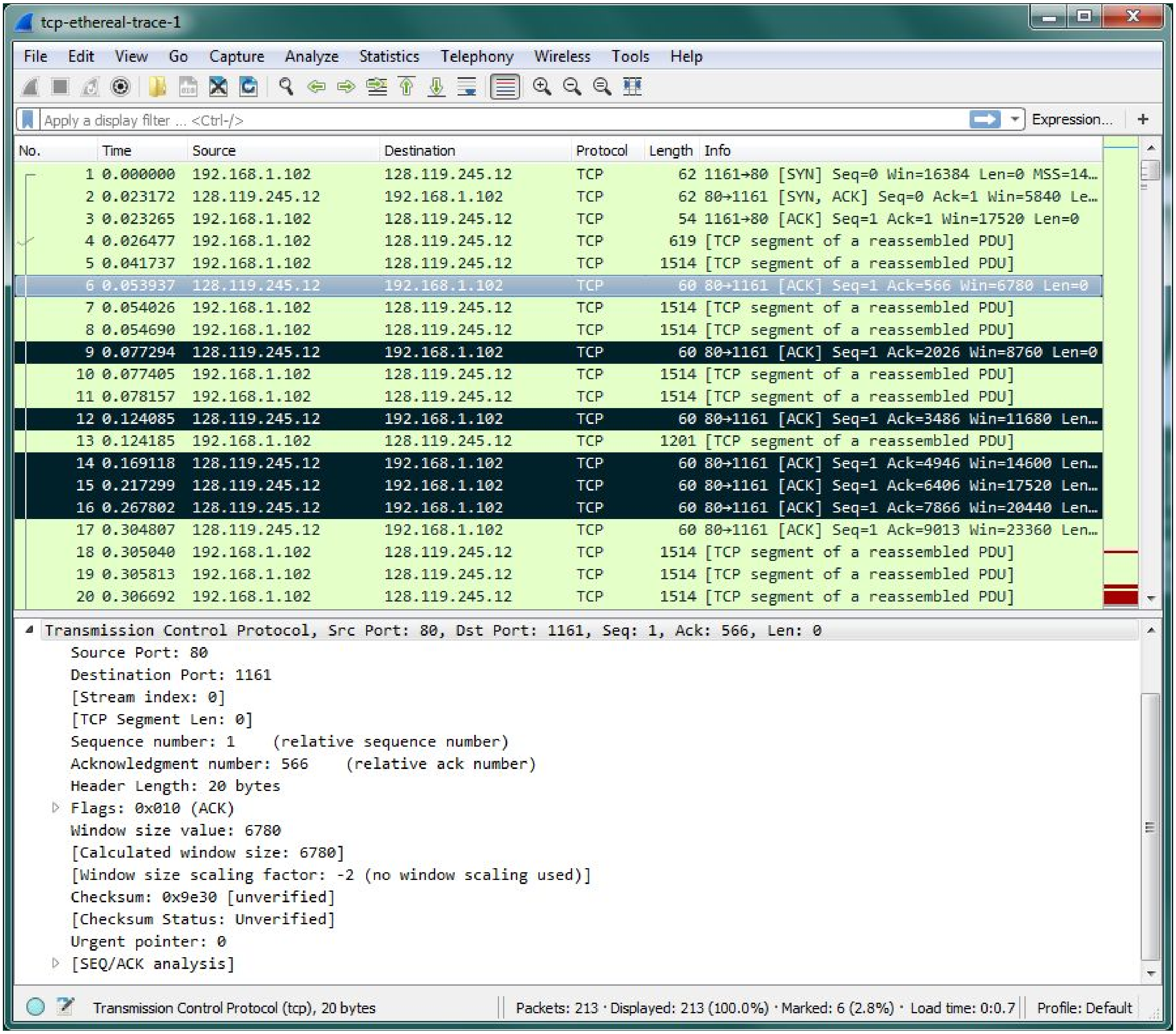
CNT 4004.001 - Wireshark Lab 4



Segments 1 through 6

Sterling Price - U49743231
Matthew Kramer - U20891900

CNT 4004.001 - Wireshark Lab 4



Acknowledgements of segments 1 through 6

Sterling Price - U49743231
Matthew Kramer - U20891900

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 239 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 239 for all subsequent segments.

Number	Packet Number		Sequence Number		Time Segment Sent	Time ACK Received	RTT
	Segment	ACK	Segment	ACK			
1	4	6	1	566	0.026477	0.053937	0.027460
2	5	9	566	2026	0.041737	0.077294	0.035557
3	7	12	2026	3486	0.054026	0.124085	0.070059
4	8	14	3486	4946	0.054690	0.169118	0.114430
5	10	15	4946	6406	0.077405	0.217299	0.139890
6	11	16	6406	7866	0.078157	0.267802	0.189640

$$EstimatedRTT = (0.875 \times EstimatedRTT) + (0.125 \times SampleRTT)$$

EstimatedRTT after **ACK 1**
$= RTT\ for\ Segment\ 1 = 0.02746\ seconds$

EstimatedRTT after **ACK 2**
$= (0.875 \times 0.02746) + (0.125 \times 0.035557) = 0.0285\ seconds$

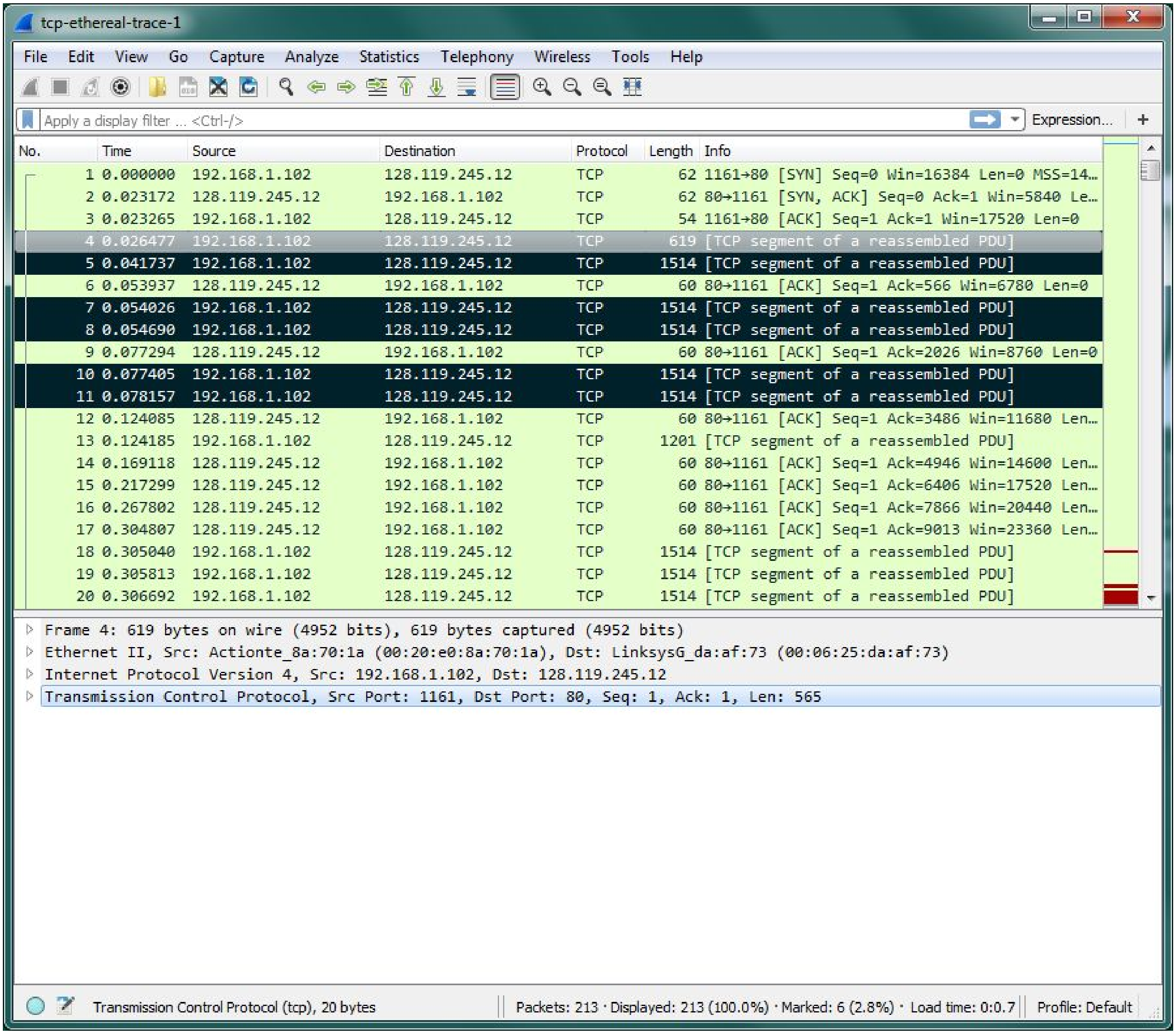
EstimatedRTT after **ACK 3**
$= (0.875 \times 0.0285) + (0.125 \times 0.070059) = 0.0337\ seconds$

EstimatedRTT after **ACK 4**
$= (0.875 \times 0.0337) + (0.125 \times 0.11443) = 0.0438\ seconds$

EstimatedRTT after **ACK 5**
$= (0.875 \times 0.0438) + (0.125 \times 0.13989) = 0.0558\ seconds$

EstimatedRTT after **ACK 6**
$= (0.875 \times 0.0558) + (0.125 \times 0.18964) = 0.0725\ seconds$

Sterling Price - U49743231
Matthew Kramer - U20891900
CNT 4004.001 - Wireshark Lab 4

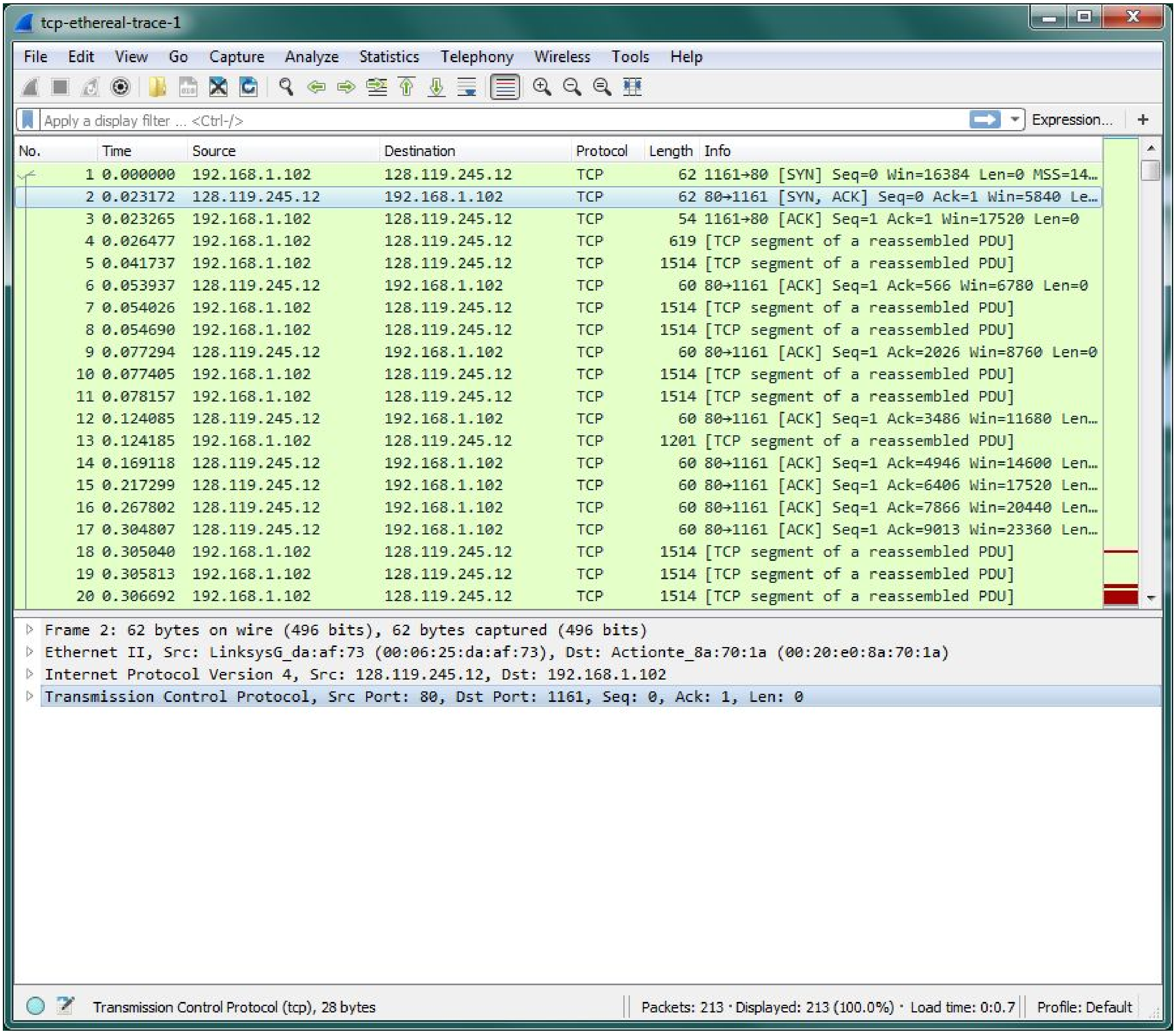


Length of first TCP segment

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

Segment 1 is 565 bytes long while segments 2 through 6 are 1460 bytes long.

Sterling Price - U49743231
Matthew Kramer - U20891900
CNT 4004.001 - Wireshark Lab 4

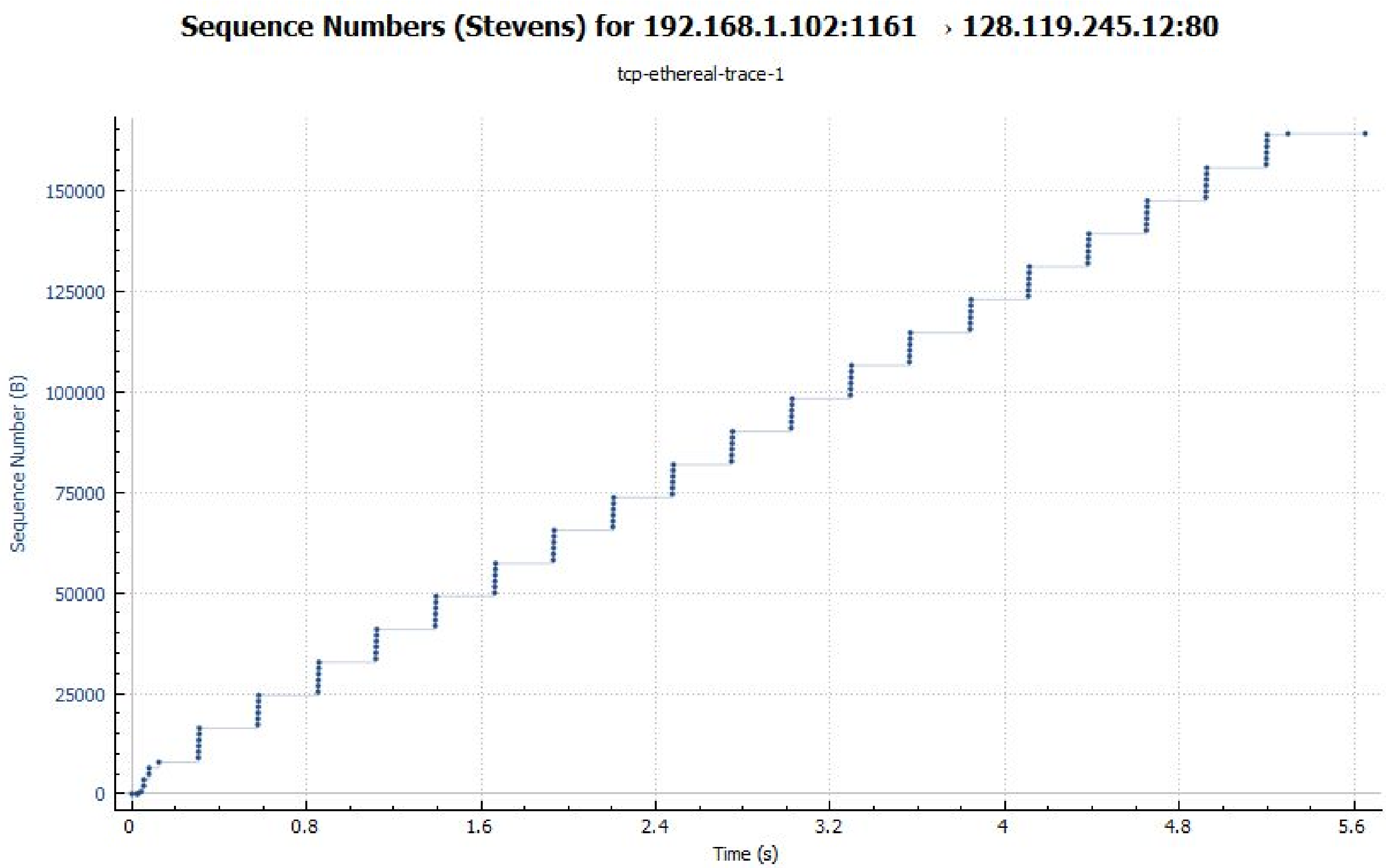


Minimum buffer space

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?

The minimum amount of available buffer space is advertised in the first acknowledgement from the server as 5840 bytes. The sender is never throttled by the lack of receiver buffer space.

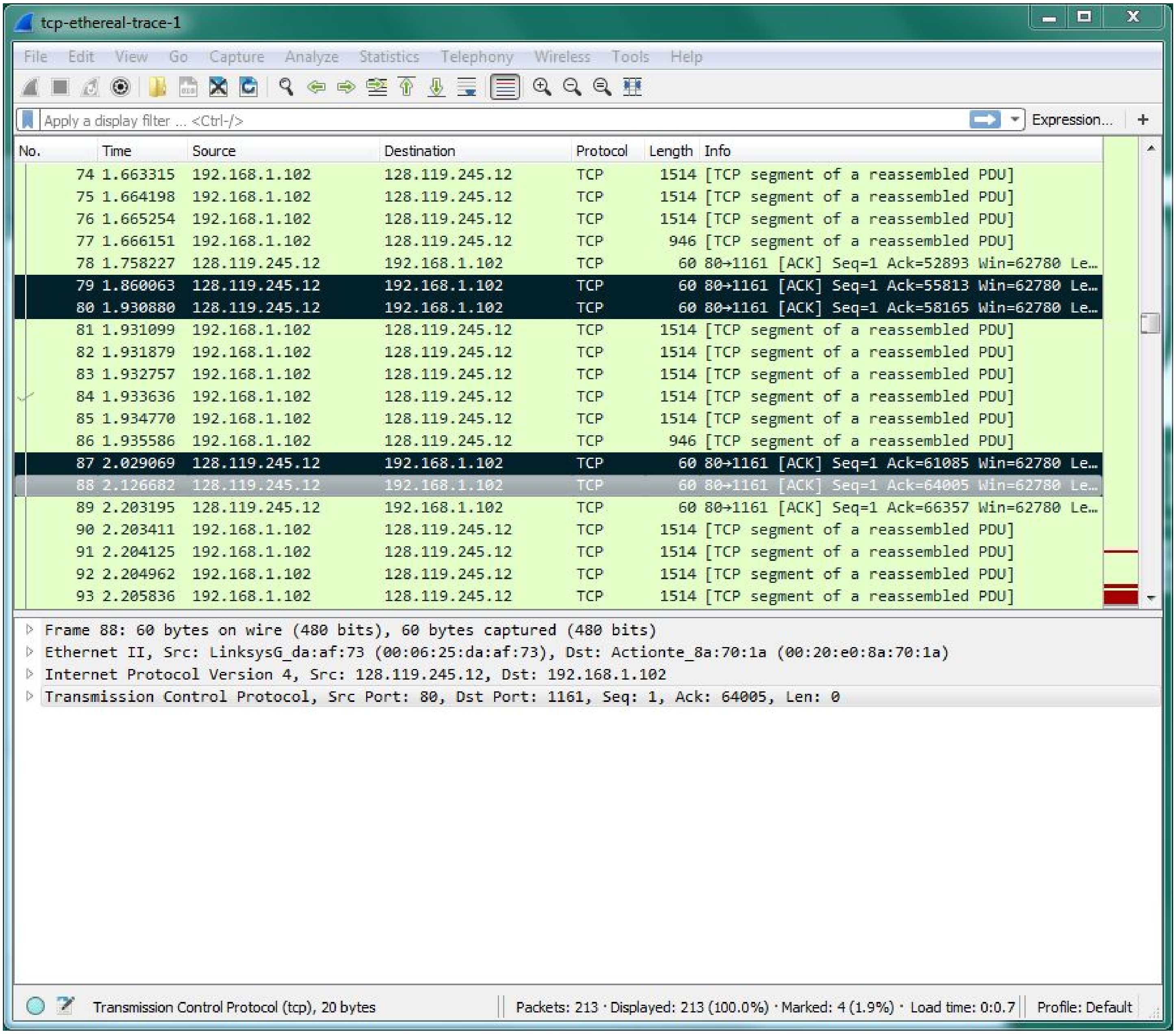
Sterling Price - U49743231
Matthew Kramer - U20891900
CNT 4004.001 - Wireshark Lab 4



Time-sequence graph for the trace

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

> There are no retransmitted segments in the trace file. The time-sequence graph reveals that each successive sequence number is greater than those before it, meaning that no segment was ever sent again.



ACKing every other segment

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 247 in the text).

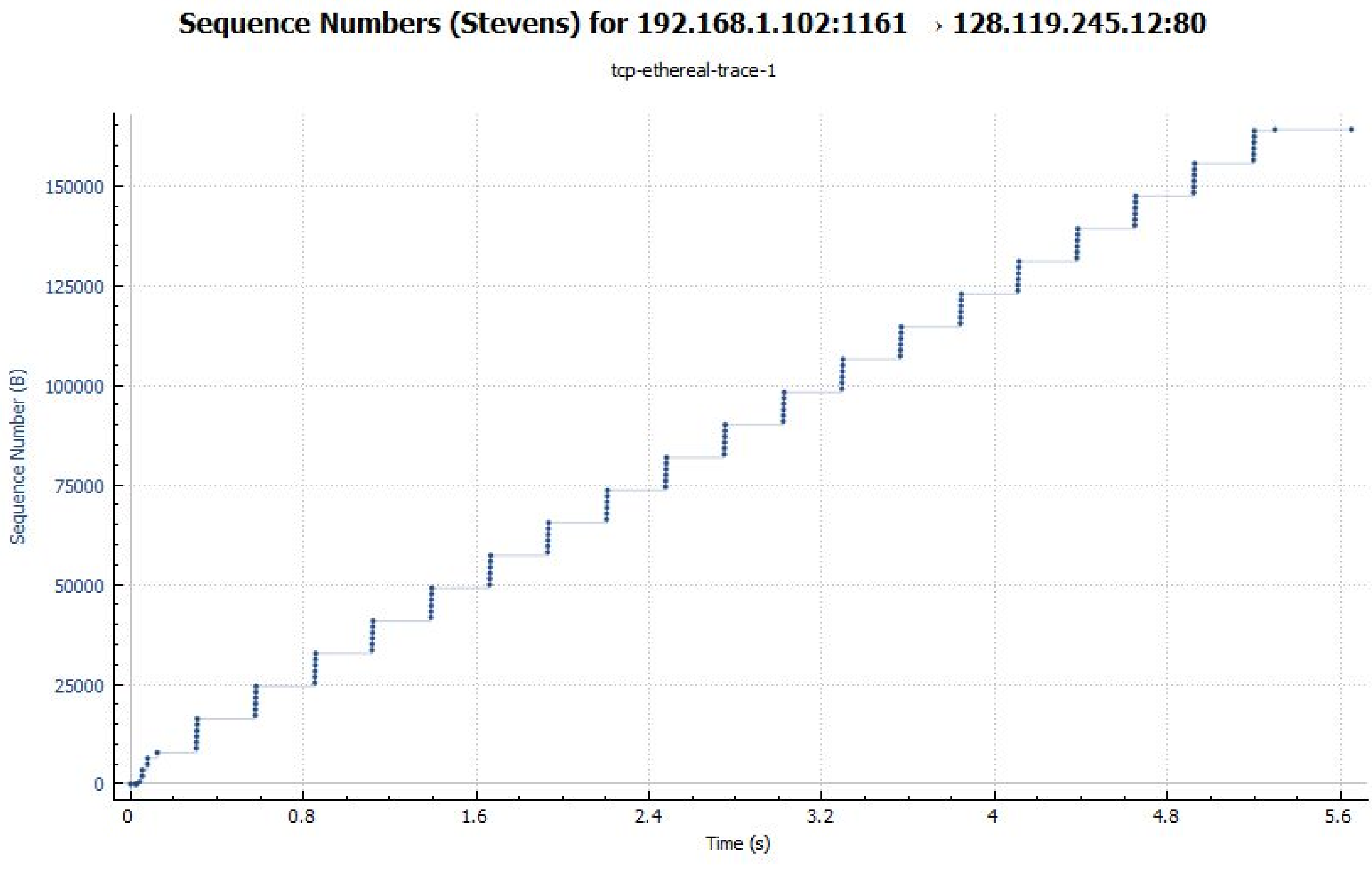
> The receiver typically acknowledges **`1460`** bytes of data. **`10`** out of **`12`** acknowledgements were of this size; of the remaining two, one acknowledged **`566`** bytes and the other **`1147`**. If you inspect the acknowledged data of each ACK, there are cases where the receiver is ACKing every other segment. For example, segment **`80`** acknowledges **`2920`** bytes of data, or **`2 × 1460`** bytes.

Sterling Price - U49743231
Matthew Kramer - U20891900

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

The throughput of the TCP connection may be calculated by taking the total amount of data transferred and dividing it by the total time the connection was open. In doing this, we can calculate an average throughput for the duration of the connection. The total amount of data transferred can be calculated by finding the difference between the sequence number of the first segment (1 byte) and last ACK (164091 bytes). The total amount of time can be calculated by finding the difference between the time the first segment was sent (0.026477 seconds) and the time the last ACK was received (5.455830 seconds).

$$Throughput = \frac{total\ bytes}{total\ time} = \frac{164091 - 1}{5.455830 - 0.026477} = \frac{164090}{5.4294} = 30.222\ bytes/second$$



Time-Sequence graph for the trace

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 slow start 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.

TCP's slow start phase begins when the connection starts, when the HTTP POST segment is sent. According to the graph, the slow start ends at approximately time 0.18.