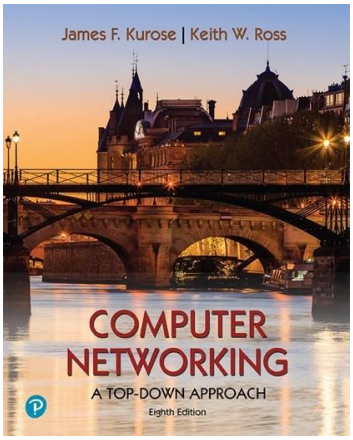
Wireshark Lab: TCP v8.0

Supplement to *Computer Networking: A Top-Down Approach, 8th ed.*, J.F. Kurose and K.W. Ross



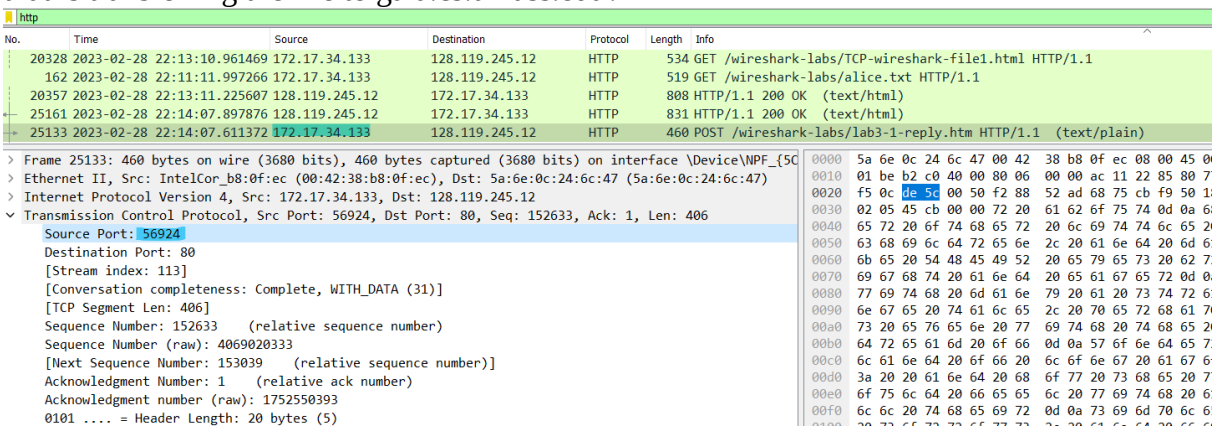
"Tell me and I forget. Show me and I remember. Involve me and I understand." Chinese proverb

---

Student Name : Nguyễn Hữu Hiếu

Student ID : 2013149

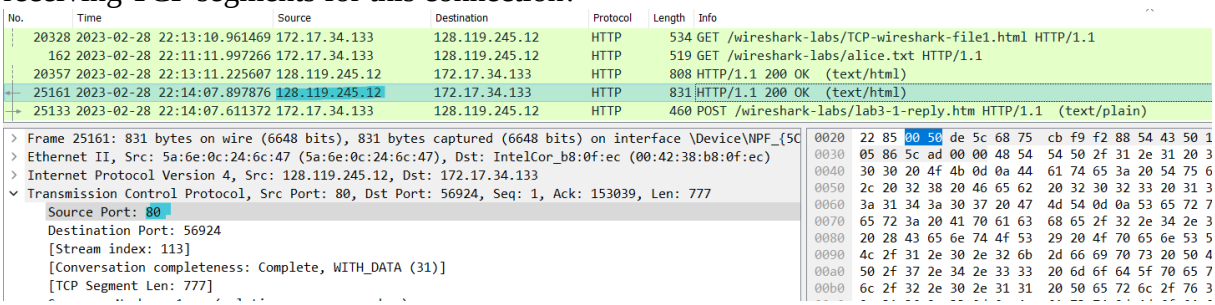
1. What is the IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu?



The IP address used by the client computer that is transferring the file to gaia.cs.umass.edu is 172.17.34.133

The TCP port is 56924

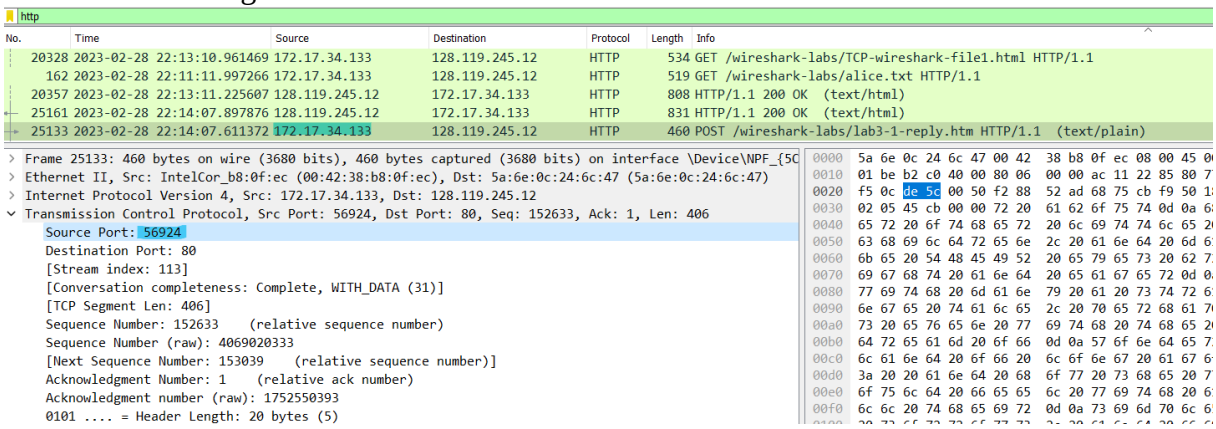
2. What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection?



The IP address of gaia.cs.umass.edu is 128.119.245.12

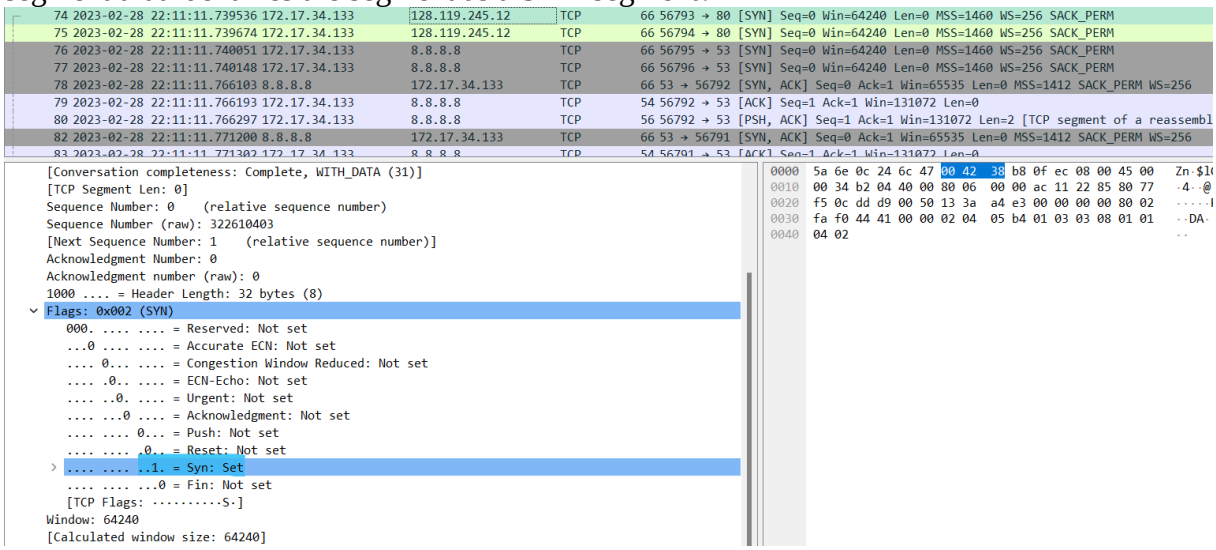
The TCP port is 80

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?



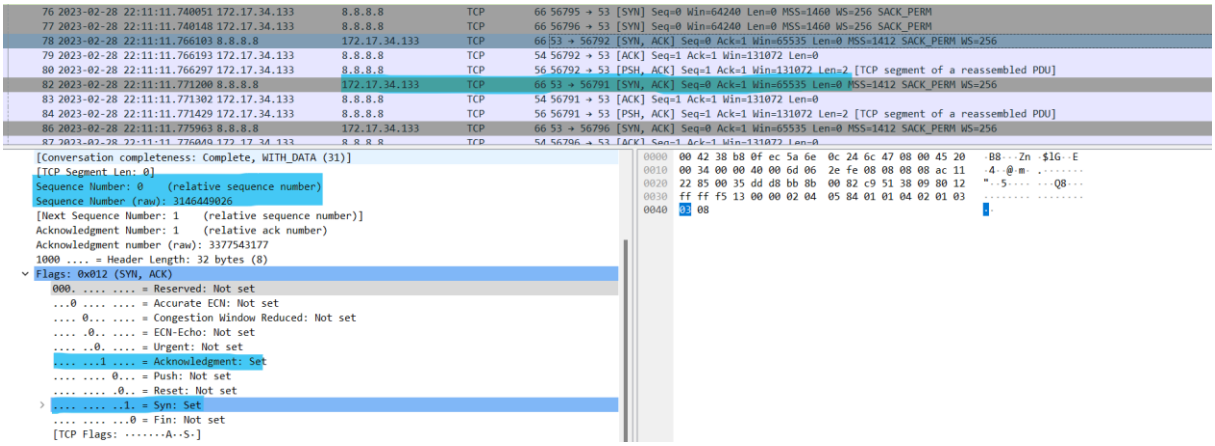
The IP address used by the client computer that is transferring the file to gaia.cs.umass.edu is 172.17.34.133
The TCP port is 56924

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 of the TCP SYN segment is 0 since it is used to imitate the TCP connection between the client computer and gaia.cs.umass.edu. According to the screenshot below, in the Flags section, the SYN flag is set to 1 which indicates that this segment is 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?



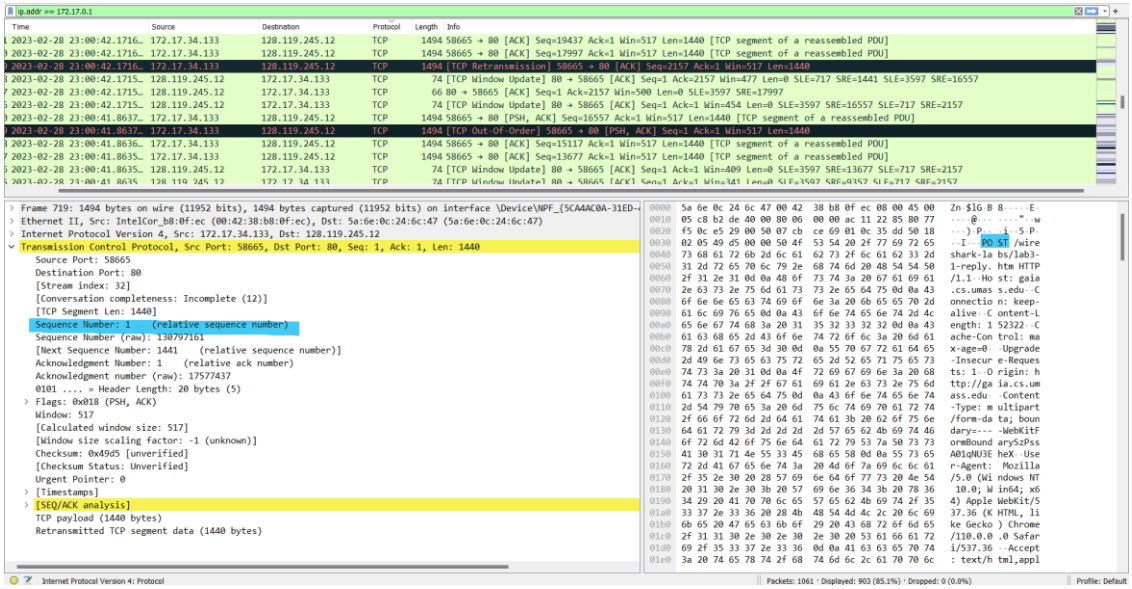
The sequence number of the SYN_ACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN is 0.

The value of the acknowledgement field in the SYN_ACK segment is determined by the server gaia.cs.umass.edu.

The server adds 1 to the initial sequence number of the SYN segment from the client computer.

For this case, the initial sequence number of the SYN segment from the client computer is 0, thus the value of the acknowledgement field in the SYN_ACK segment is 1. A segment will be identified as a SYN_ACK segment if both SYN flag and ACKnowledgement flag in the segment are set to 1.

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.

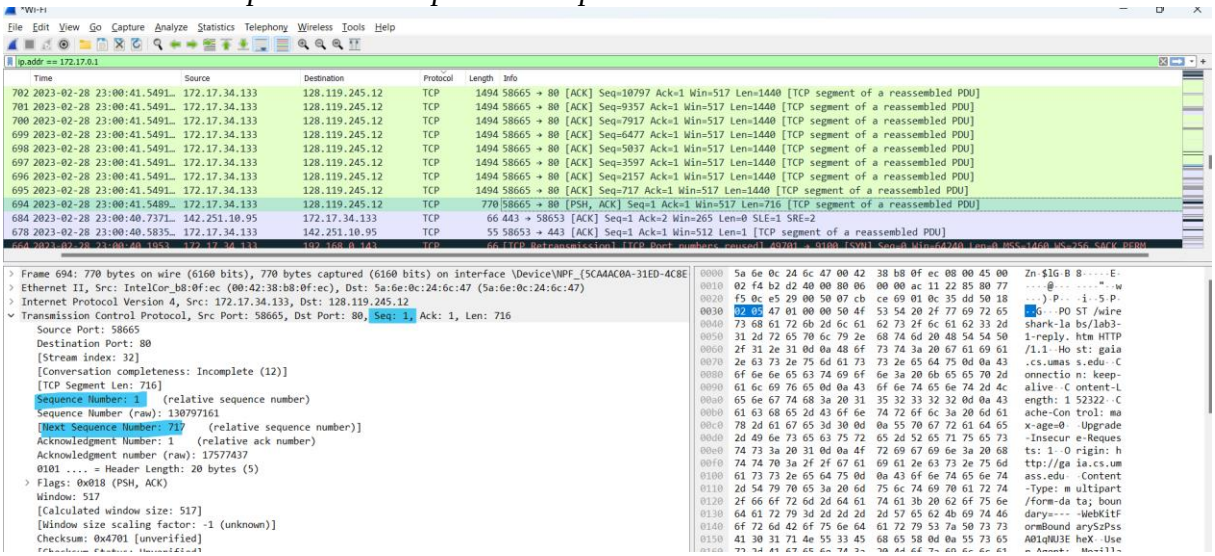


The sequence number of the TCP segment containing the HTTP Post command is 1.

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.*

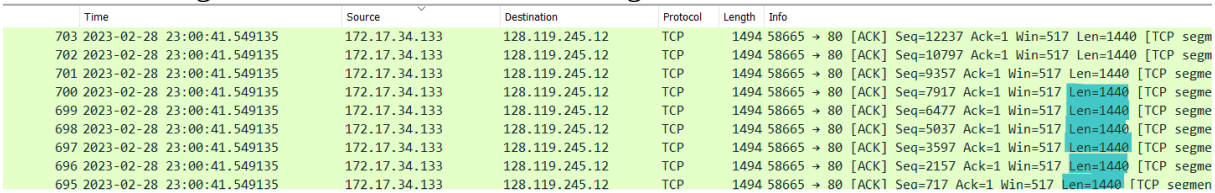


Sequence number for segment 1 is 1, sequence number for segment 2 is 717.

Seg1: Arrival Time: Feb 28, 2023 23:00:41.548950000 SE Asia Standard Time

Seg2: Arrival Time: Feb 28, 2023 23:00:41.549135000 SE Asia Standard Time

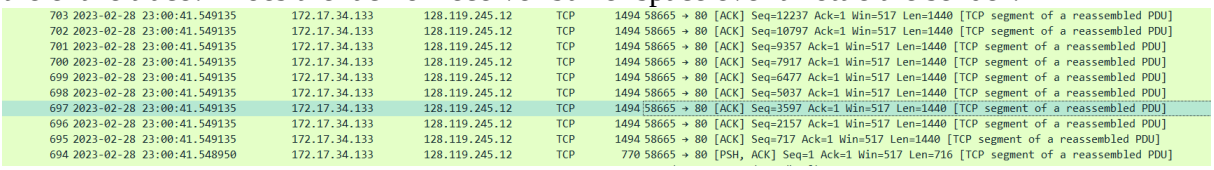
8. What is the length of each of the first six TCP segments?[1]



	Time	Source	Destination	Protocol	Length	Info
703	2023-02-28 23:00:41.549135	172.17.34.133	128.119.245.12	TCP	1494	58665 → 80 [ACK] Seq=12237 Ack=1 Win=517 Len=1440 [TCP segm
702	2023-02-28 23:00:41.549135	172.17.34.133	128.119.245.12	TCP	1494	58665 → 80 [ACK] Seq=10797 Ack=1 Win=517 Len=1440 [TCP segm
701	2023-02-28 23:00:41.549135	172.17.34.133	128.119.245.12	TCP	1494	58665 → 80 [ACK] Seq=9357 Ack=1 Win=517 Len=1440 [TCP segme
700	2023-02-28 23:00:41.549135	172.17.34.133	128.119.245.12	TCP	1494	58665 → 80 [ACK] Seq=7917 Ack=1 Win=517 Len=1440 [TCP segme
699	2023-02-28 23:00:41.549135	172.17.34.133	128.119.245.12	TCP	1494	58665 → 80 [ACK] Seq=6477 Ack=1 Win=517 Len=1440 [TCP segme
698	2023-02-28 23:00:41.549135	172.17.34.133	128.119.245.12	TCP	1494	58665 → 80 [ACK] Seq=5037 Ack=1 Win=517 Len=1440 [TCP segme
697	2023-02-28 23:00:41.549135	172.17.34.133	128.119.245.12	TCP	1494	58665 → 80 [ACK] Seq=3597 Ack=1 Win=517 Len=1440 [TCP segme
696	2023-02-28 23:00:41.549135	172.17.34.133	128.119.245.12	TCP	1494	58665 → 80 [ACK] Seq=2157 Ack=1 Win=517 Len=1440 [TCP segme
695	2023-02-28 23:00:41.549135	172.17.34.133	128.119.245.12	TCP	1494	58665 → 80 [ACK] Seq=717 Ack=1 Win=517 Len=1440 [TCP segmen

the length of each of the first six TCP segments is 1440

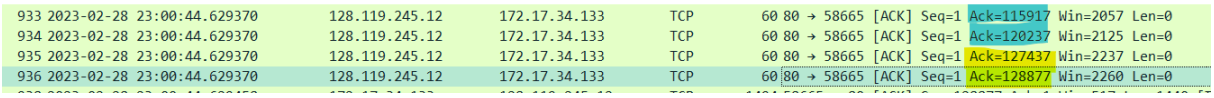
9. What is the minimum amount of available buffer space advertised at the received for the entire trace? Does the lack of receiver buffer space ever throttle the sender?



703	2023-02-28 23:00:41.549135	172.17.34.133	128.119.245.12	TCP	1494	58665 → 80 [ACK] Seq=12237 Ack=1 Win=517 Len=1440 [TCP segment of a reassembled PDU]
702	2023-02-28 23:00:41.549135	172.17.34.133	128.119.245.12	TCP	1494	58665 → 80 [ACK] Seq=10797 Ack=1 Win=517 Len=1440 [TCP segment of a reassembled PDU]
701	2023-02-28 23:00:41.549135	172.17.34.133	128.119.245.12	TCP	1494	58665 → 80 [ACK] Seq=9357 Ack=1 Win=517 Len=1440 [TCP segment of a reassembled PDU]
700	2023-02-28 23:00:41.549135	172.17.34.133	128.119.245.12	TCP	1494	58665 → 80 [ACK] Seq=7917 Ack=1 Win=517 Len=1440 [TCP segment of a reassembled PDU]
699	2023-02-28 23:00:41.549135	172.17.34.133	128.119.245.12	TCP	1494	58665 → 80 [ACK] Seq=6477 Ack=1 Win=517 Len=1440 [TCP segment of a reassembled PDU]
698	2023-02-28 23:00:41.549135	172.17.34.133	128.119.245.12	TCP	1494	58665 → 80 [ACK] Seq=5037 Ack=1 Win=517 Len=1440 [TCP segment of a reassembled PDU]
697	2023-02-28 23:00:41.549135	172.17.34.133	128.119.245.12	TCP	1494	58665 → 80 [ACK] Seq=3597 Ack=1 Win=517 Len=1440 [TCP segment of a reassembled PDU]
696	2023-02-28 23:00:41.549135	172.17.34.133	128.119.245.12	TCP	1494	58665 → 80 [ACK] Seq=2157 Ack=1 Win=517 Len=1440 [TCP segment of a reassembled PDU]
695	2023-02-28 23:00:41.549135	172.17.34.133	128.119.245.12	TCP	1494	58665 → 80 [ACK] Seq=717 Ack=1 Win=517 Len=1440 [TCP segment of a reassembled PDU]
694	2023-02-28 23:00:41.548950	172.17.34.133	128.119.245.12	TCP	770	58665 → 80 [PSH, ACK] Seq=1 Ack=1 Win=517 Len=716 [TCP segment of a reassembled PDU]

the minimum amount of available buffer space advertised at the received for the entire trace is 517

10. Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question?
    No there is no retransmitted segments in the trace file
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).
    According to the screenshot below, we can see that the ACK numbers increase in the sequence of 1401, 2801, 4201, and so on. The ACK numbers increases by 1400 each time, indicating that the receiver is acknowledging 1400 bytes



933	2023-02-28 23:00:44.629370	128.119.245.12	172.17.34.133	TCP	60	80 → 58665 [ACK] Seq=1 Ack=115917 Win=2057 Len=0
934	2023-02-28 23:00:44.629370	128.119.245.12	172.17.34.133	TCP	60	80 → 58665 [ACK] Seq=1 Ack=120237 Win=2125 Len=0
935	2023-02-28 23:00:44.629370	128.119.245.12	172.17.34.133	TCP	60	80 → 58665 [ACK] Seq=1 Ack=127437 Win=2237 Len=0
936	2023-02-28 23:00:44.629370	128.119.245.12	172.17.34.133	TCP	60	80 → 58665 [ACK] Seq=1 Ack=128877 Win=2260 Len=0

[1] The TCP segments in the tcp-ethereal-trace-1 trace file are all less that 1460 bytes. This is because the computer on which the trace was gathered has an Ethernet card that limits the length of the maximum IP packet to 1500 bytes (40 bytes of TCP/IP header data and 1460 bytes of TCP payload). This 1500 byte value is the standard maximum length allowed by Ethernet. If your trace indicates a TCP length greater than 1500 bytes, and your computer is using an Ethernet connection, then Wireshark is reporting the wrong TCP segment length; it will likely also show only one large TCP segment rather than multiple smaller segments. Your computer is indeed probably sending multiple smaller segments, as indicated by the ACKs it receives. This inconsistency in reported segment lengths is due to the interaction between the Ethernet driver and the Wireshark software. We recommend that if you have this inconsistency, that you perform this lab using the provided trace file.