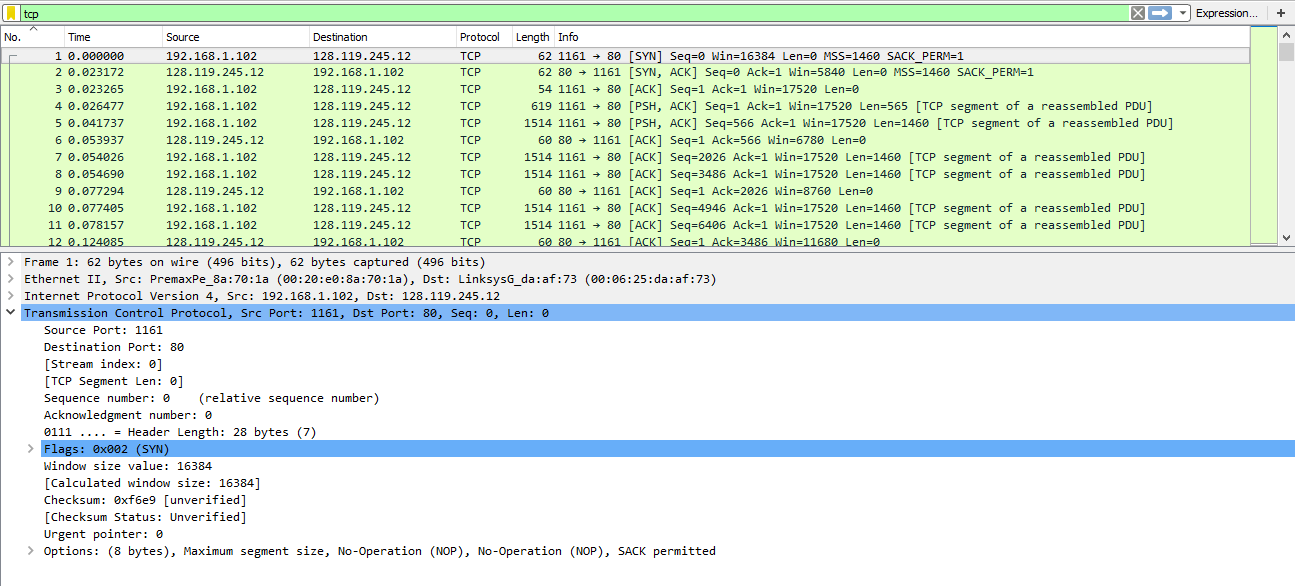
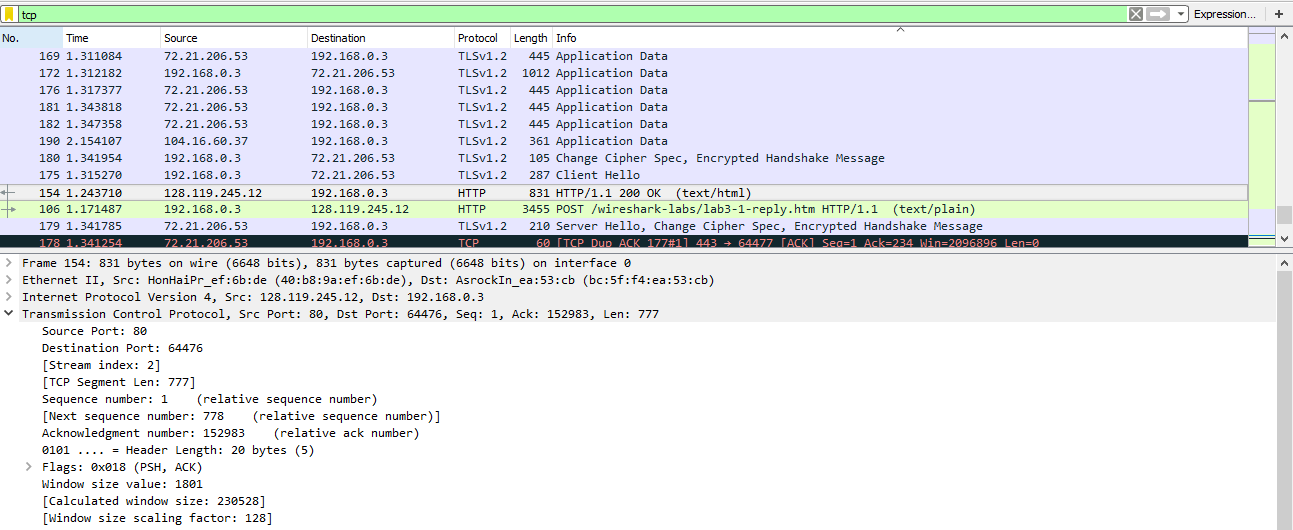
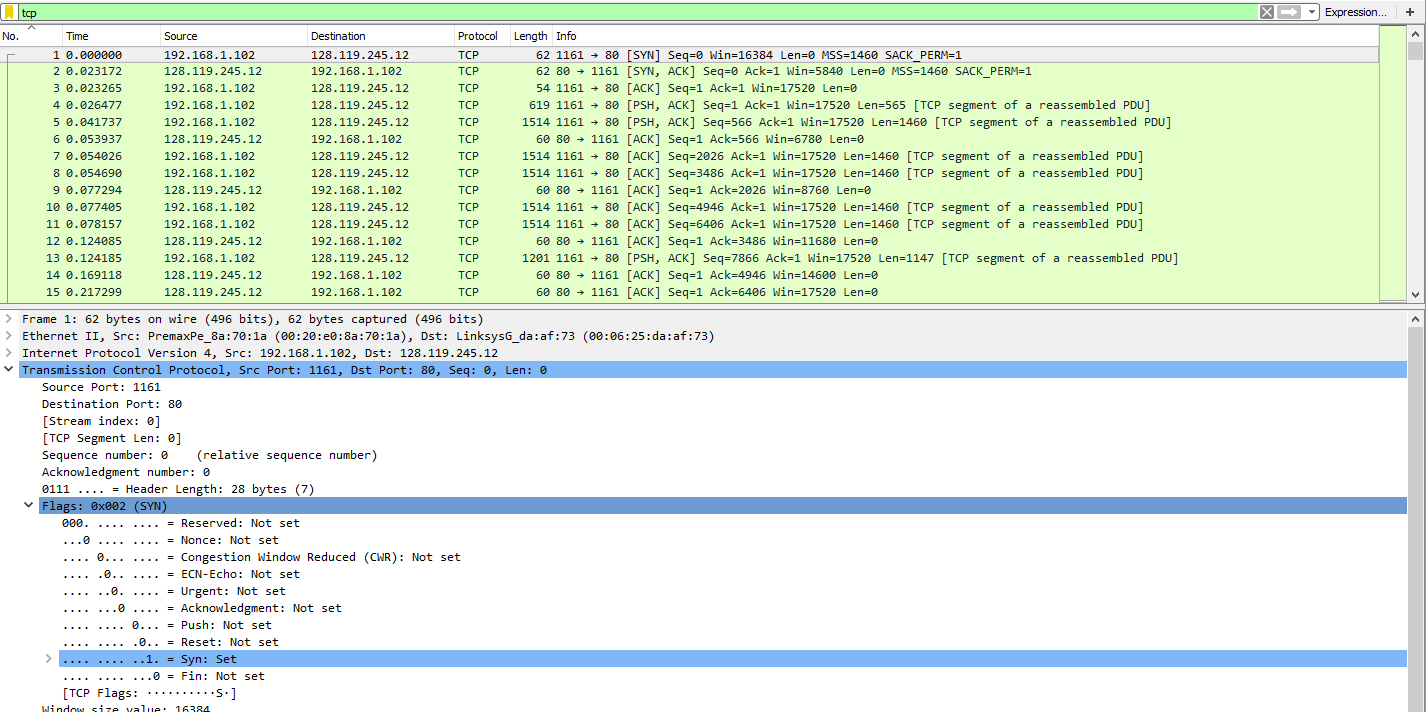
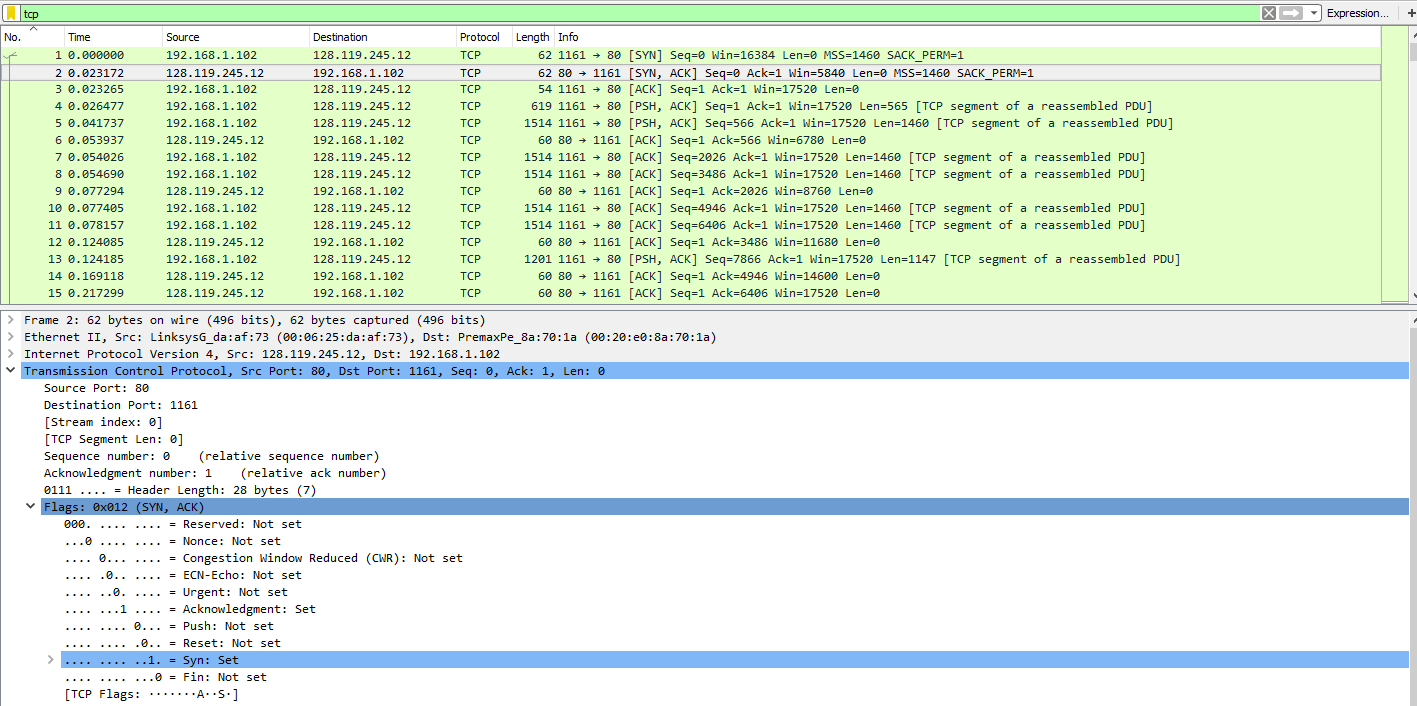
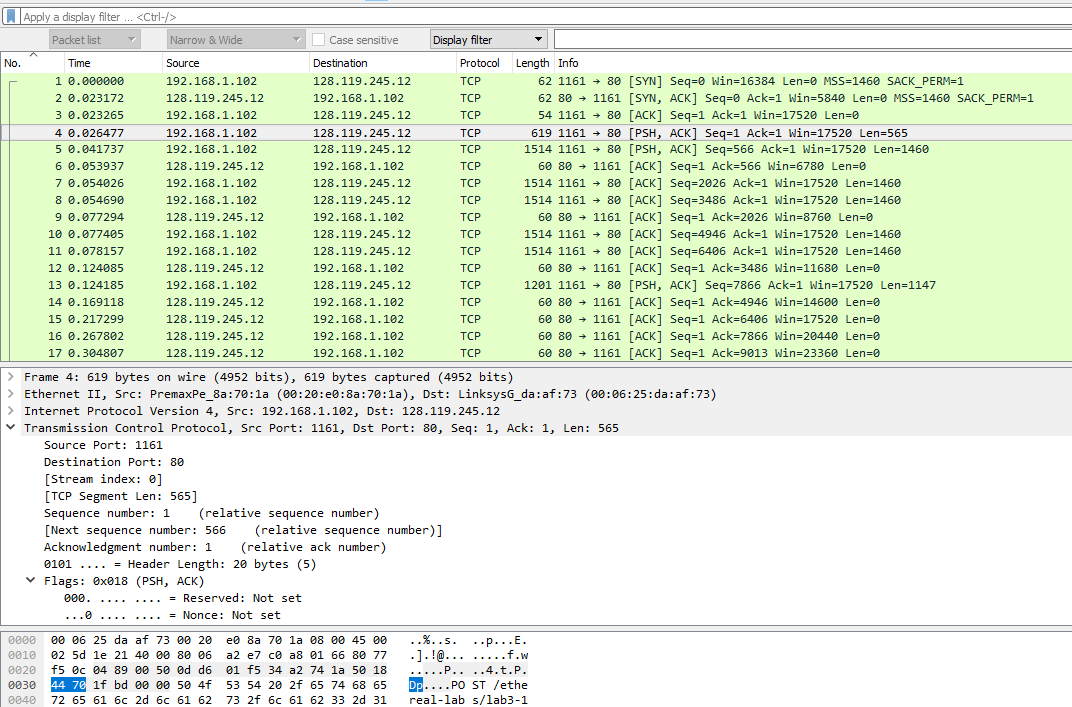
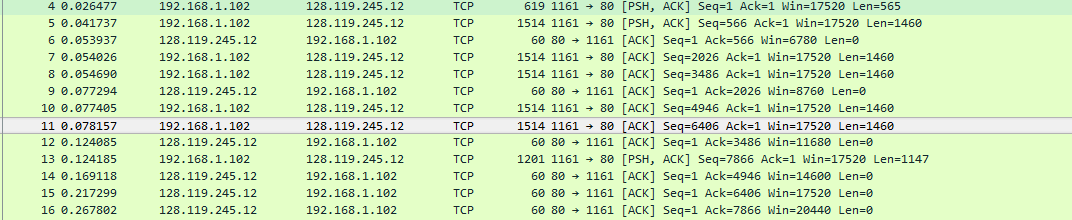
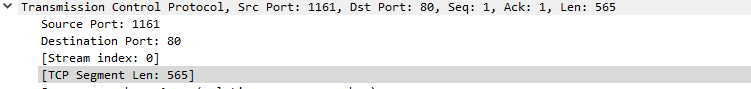
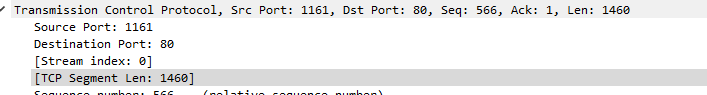
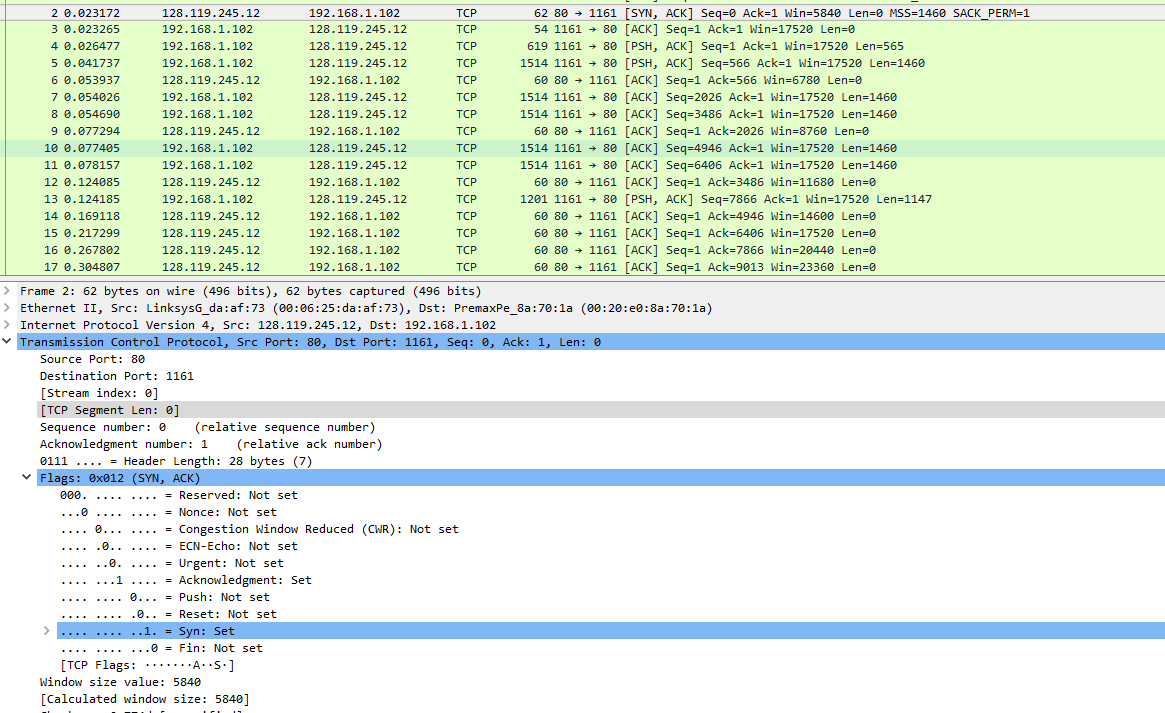
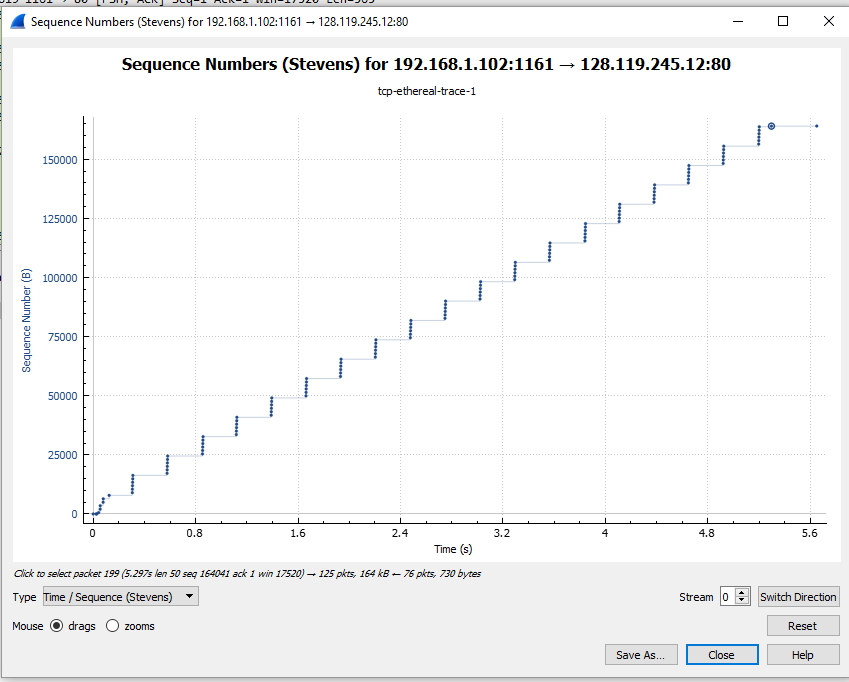
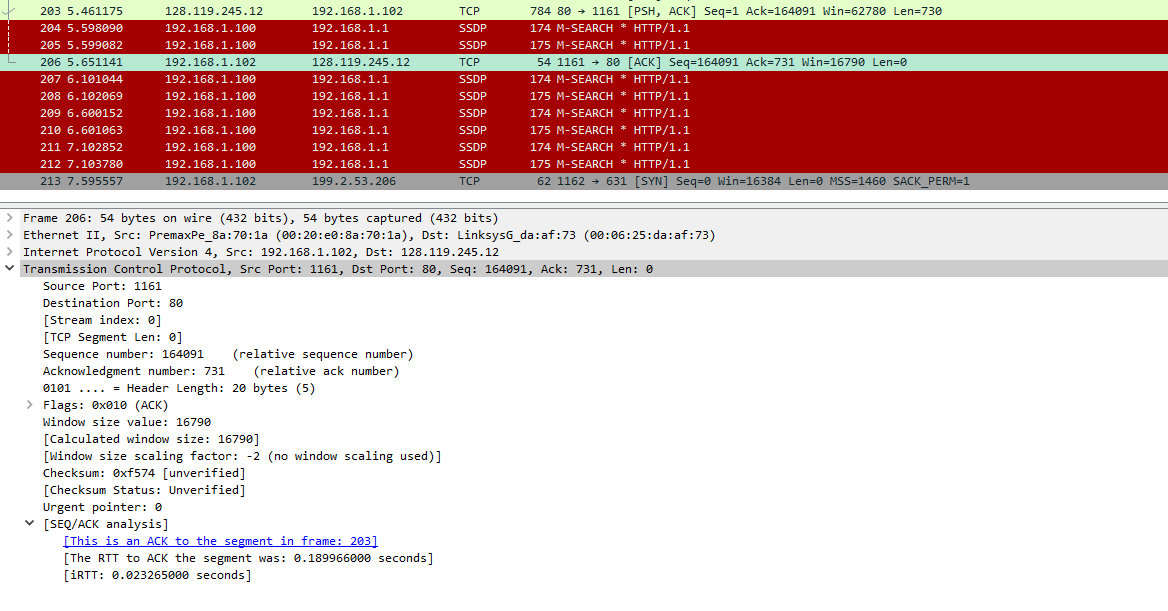
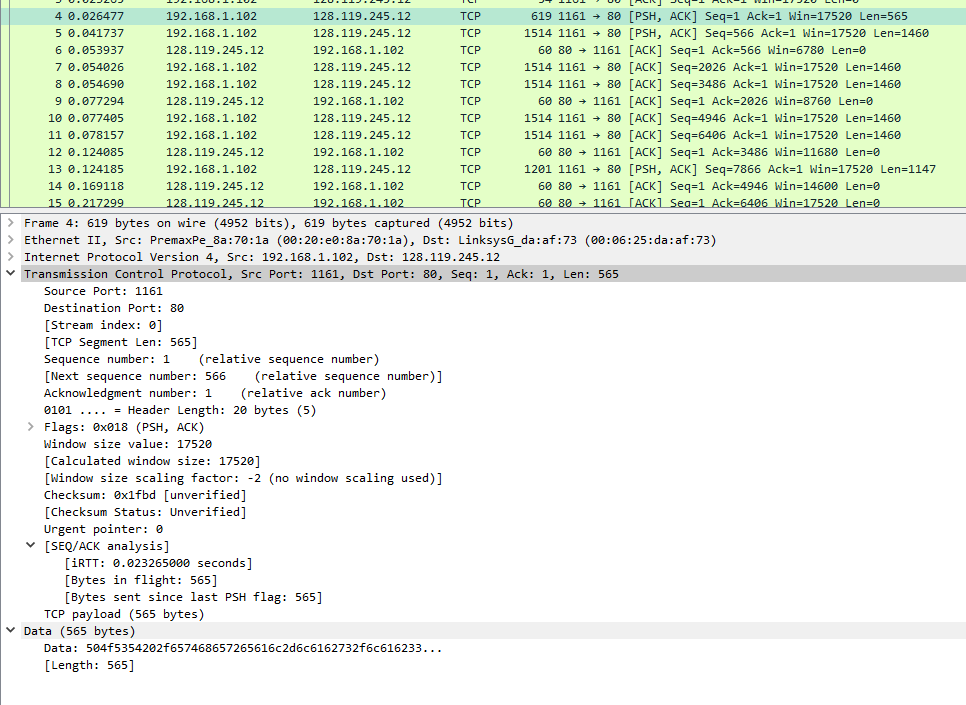
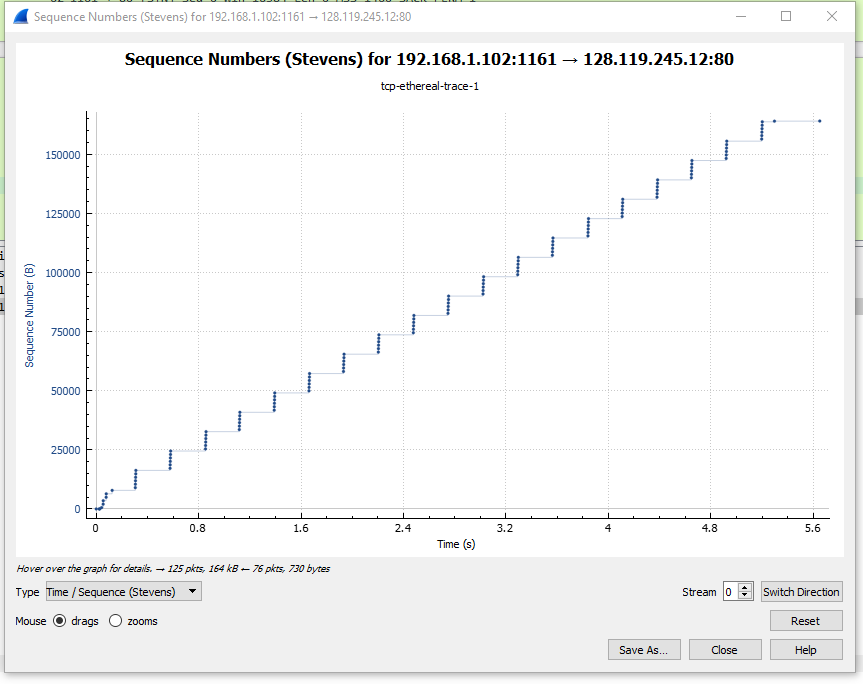
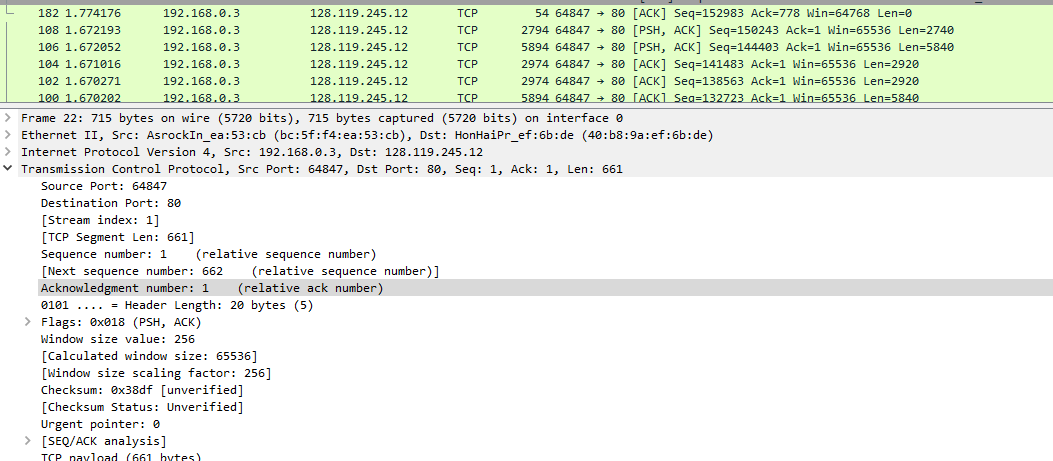
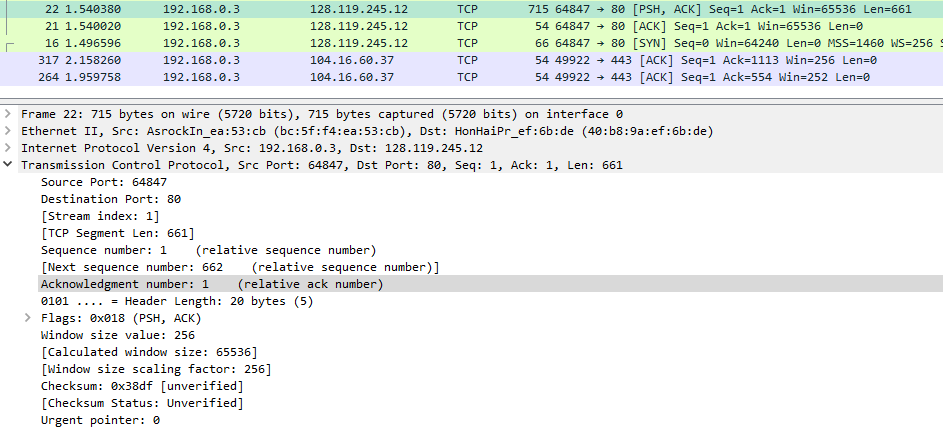
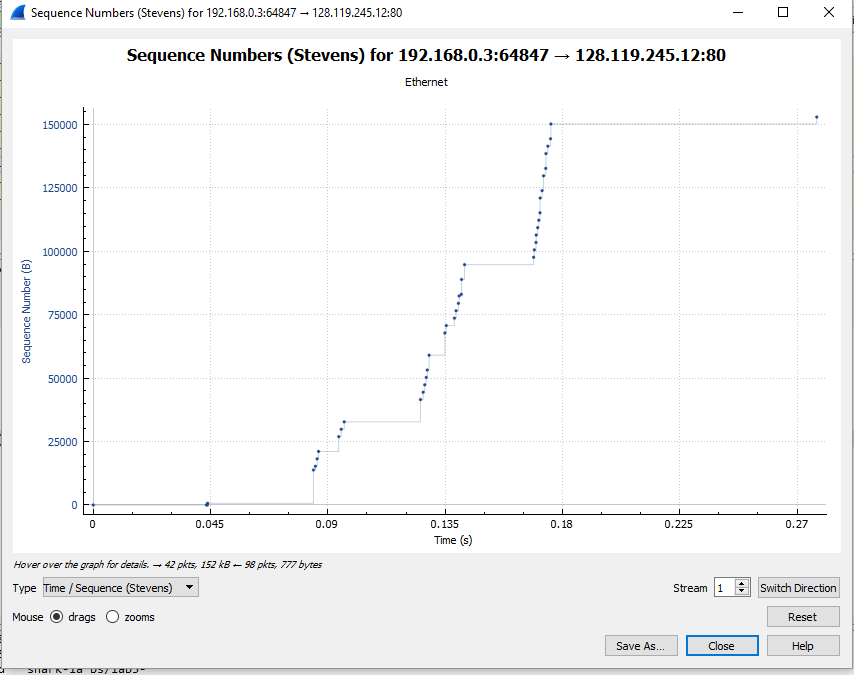
1. **What is the IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu? To answer this question, it’s probably easiest to select an HTTP message and explore the details of the TCP packet used to carry this HTTP message, using the “details 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.**

The IP Address is : 192.168.1.102, and the port is 1161

1. **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 for gaia.cs.umass.edu is: 128.119.245.12, and the port is 80
2. **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 is 192.168.0.3 and the port is 64476
3. **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. The [SYN] flag in the information field identifies the segment as a SYN segment.
4. **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 for the SYNACK segment is 0. The value for the Acknowledgement field is 1. The gaia.cs.umass.edu knew the acknowledgement would be 1 because it is a relative ack number. The [SYN,ACK] flag in the Info field identifies this segment as a SYNACK segment.
5. **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 packet number 4 contains the HTTP POST command, and the sequence number of the HTTP POST command is 1.
6. **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? 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?**  
     
   Segment 1: Sent time: 0.026477; Ack time: 0.053937; RTT: 0.053937-0.026477 = 0.02746  
   Segment 2: Sent time: 0.041737; Ack time: 0.077294; RTT: 0.077294-0.041737 = 0.035557  
   Segment3: Sent time: 0.054026; Ack time: 0.124085; RTT: : 0.124085-0.054026 = 0.070059  
   Segment4: Sent time: 0.054690; Ack time: 0.169118; RTT: 0.169118-0.054690 = 0.114428  
   Segment5: Sent time: 0.077405; Ack time: 0.217299: RTT: 0.217299-0.077405 = 0.139894  
   Segment6: Sent time: 0.078157; Ack time: 0.267802; RTT: 0.267802-0.078157 = 0.189645  
     
     
     
     
     
     
     
    **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.**  
     
   The estimated RTT is calculated by : 0.875 \* Estimated RTT(previous) + 0.125 \* SampleRTT  
     
   Segment1: 0.02746  
   Segment2: 0.875\*0.02746 + 0.125\*0.035557 = 0.02847  
   Segment3: 0.875\*0.0285 + 0.125\*0.070059 = 0.03369  
   Segment4: 0.875\*0.0337 + 0.125\*0.0114428 = 0.0438  
   Segment5: 0.875\*0.0438 + 0.125\*0.139894 = 0.0558  
   Segment6: 0.875\*0.0558 + 0.125\*0.189645 = 0.0725
7. **What is the length of each of the first six TCP segments?**  
     
     
     
     
     
     
   Segment1: 565 bytes  
   Segment2: 1460  
   Segment3: 1460  
   Segment4: 1460  
   Segment5: 1460  
   Segment6: 1460
8. **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 5640 bytes. The sender is never throttled.
9. **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. Analyzing the Time-Sequence-Graph(Stevens), we can determine that no sequence is out of place, everything is increased in both the x and y direction with no backtracking, thus there were no retransmitted segments.
10. **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).**  
      
    In Package 6, the acknowledge sequence number is 566.  
    In Package 9, the acknowledge sequence number is 2026.  
    In Package 12, acknowledge sequence number is 3486.  
    In Package 14, acknowledge sequence number is 4946.  
      
    The receiver typically acknowledges 1460 bytes of data.
11. **What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.**  
      
    164090 / (5.651141 – 0.026477) = 29173.298 bytes per second  
      
    The final packet acknowledgement is packet 206, which occurred at time 5.651141. The final sequence number is 164091. The first packet occurred at time 0.026477. I used the calculation above to determine the throughput.
12. **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.**  
      
      
      
      
    The slow start phase begins at time 0, and lasts until rough time 0.2. The congestion avoidance starts around 0.3, and lasts until the end of the packets.
13. **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**
14. **What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.**  
      
    152983 / (1.774176 – 1.540380) = 654343.95798 bytes per second  
    The first packet #22 occurred at time 1.540380. The final packet #182 occurred at 1.774176, and the final sequence number was 152983.
15. **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.  
      
      
      
    The slow start segment occurs around the 0.045 time, ending around the 0.05 time.The congestion avoidance takes over around the 0.08 time.