

California State University, Monterey Bay

Week 5 – Lab 5

Group 10

Clarence Mitchell

CST311

Introduction to Computer Networks

SUMMER 2015

Instructor: Dr. Anand Seetharam

CAPTURED TRACE

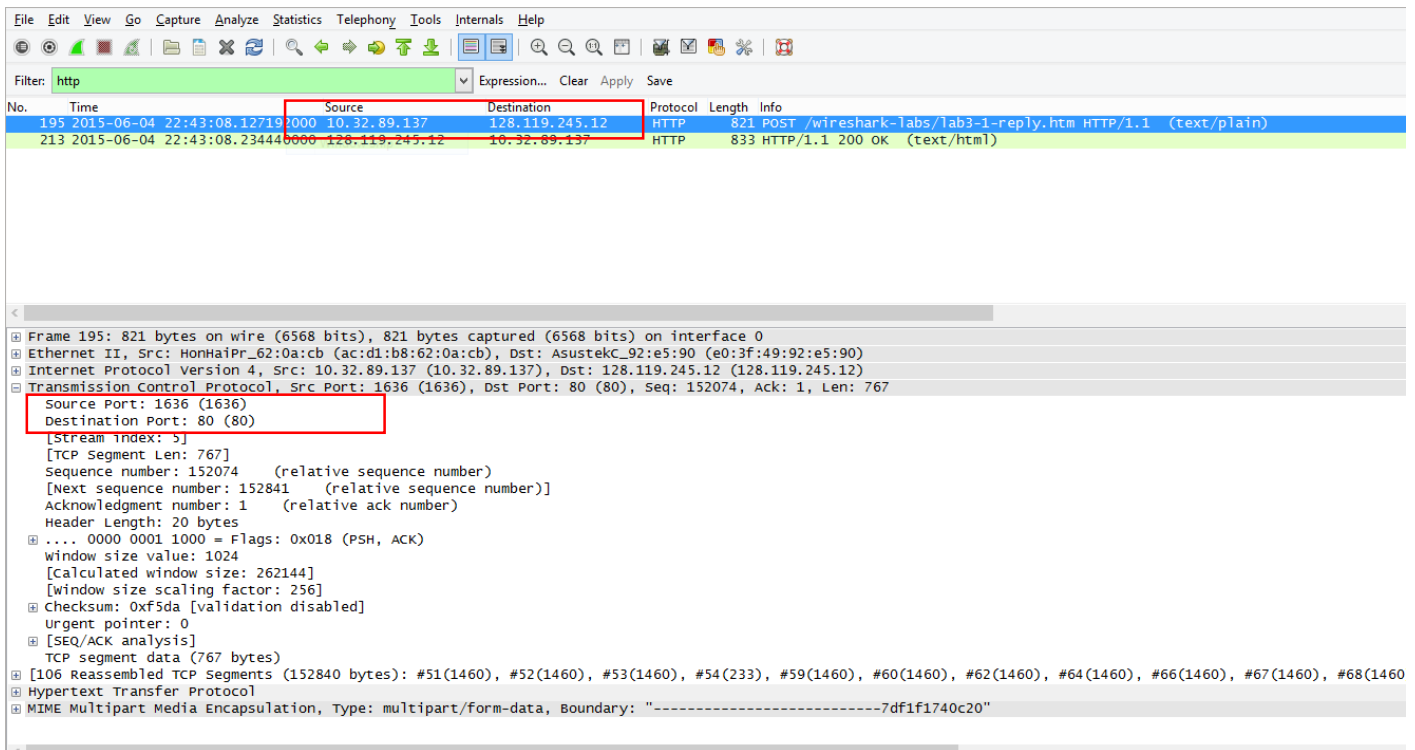
The following are the list of questions and related answers for this lab.

The image shows a Wireshark packet capture trace. The top pane displays a list of captured packets. Packet 41 is selected, showing a TCP segment from source 10.32.89.137 to destination 128.119.245.12 on port 80. The bottom pane shows the details of this packet, including the Ethernet II header, Internet Protocol Version 4 header, and Transmission Control Protocol header. The TCP header shows a SYN flag set, sequence number 0, and window size 65535.

No.	Time	Source	Destination	Protocol	Length	Info
38	2015-06-04 22:43:07.450093000	157.56.17.247	10.32.89.137	TCP	1514	[TCP segment of a reassembled PDU]
39	2015-06-04 22:43:07.450169000	10.32.89.137	157.56.17.247	TCP	54	1635-443 [ACK] Seq=186 Ack=2921 Win=65536 Len=0
40	2015-06-04 22:43:07.494823000	157.56.17.247	10.32.89.137	TLSv1.2	76	Server Hello, Certificate, Server Key Exchange, Server Hello Done
41	2015-06-04 22:43:07.508390000	10.32.89.137	128.119.245.12	TCP	66	1636-80 [SYN] Seq=0 win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
42	2015-06-04 22:43:07.510140000	10.32.89.137	128.119.245.12	TCP	66	1637-80 [SYN] Seq=0 win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
43	2015-06-04 22:43:07.512457000	10.32.89.137	157.56.17.247	TLSv1.2	268	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
45	2015-06-04 22:43:07.565821000	157.56.17.247	10.32.89.137	TLSv1.2	161	Change Cipher Spec, Encrypted Handshake Message
46	2015-06-04 22:43:07.566499000	10.32.89.137	157.56.17.247	TLSv1.2	795	Application Data
47	2015-06-04 22:43:07.566664000	10.32.89.137	157.56.17.247	TLSv1.2	203	Application Data
48	2015-06-04 22:43:07.605003000	157.56.17.247	10.32.89.137	TCP	60	443-1635 [ACK] Seq=3050 Ack=1290 Win=64512 Len=0
49	2015-06-04 22:43:07.609376000	128.119.245.12	10.32.89.137	TCP	66	80-1636 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM=1 WS=128
50	2015-06-04 22:43:07.609475000	10.32.89.137	128.119.245.12	TCP	54	1636-80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
51	2015-06-04 22:43:07.611608000	10.32.89.137	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
52	2015-06-04 22:43:07.611642000	10.32.89.137	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
53	2015-06-04 22:43:07.611659000	10.32.89.137	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
54	2015-06-04 22:43:07.611675000	10.32.89.137	128.119.245.12	TCP	287	[TCP segment of a reassembled PDU]
55	2015-06-04 22:43:07.612428000	10.32.89.137	81.19.104.84	SSL	416	Continuation Data

Frame 41: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
 Ethernet II, Src: HonHaiPr_62:0a:cb (ac:d1:b8:62:0a:cb), Dst: AsustekC_92:e5:90 (e0:3f:49:92:e5:90)
 Internet Protocol Version 4, Src: 10.32.89.137 (10.32.89.137), Dst: 128.119.245.12 (128.119.245.12)
 Transmission Control Protocol, Src Port: 1636 (1636), Dst Port: 80 (80), Seq: 0, Len: 0
 Source Port: 1636 (1636)
 Destination Port: 80 (80)
 [Stream Index: 5]
 [TCP Segment Len: 0]
 Sequence number: 0 (relative sequence number)
 Acknowledgment number: 0
 Header Length: 32 bytes
 ... 0000 0000 0010 = Flags: 0x002 (SYN)
 Window size value: 65535
 [Calculated window size: 65535]
 Checksum: 0xf07a [validation disabled]
 Urgent pointer: 0
 Options: (12 bytes), Maximum segment size, No-operation (NOP), window scale, No-operation (NOP), No-operation (NOP), SACK permitted

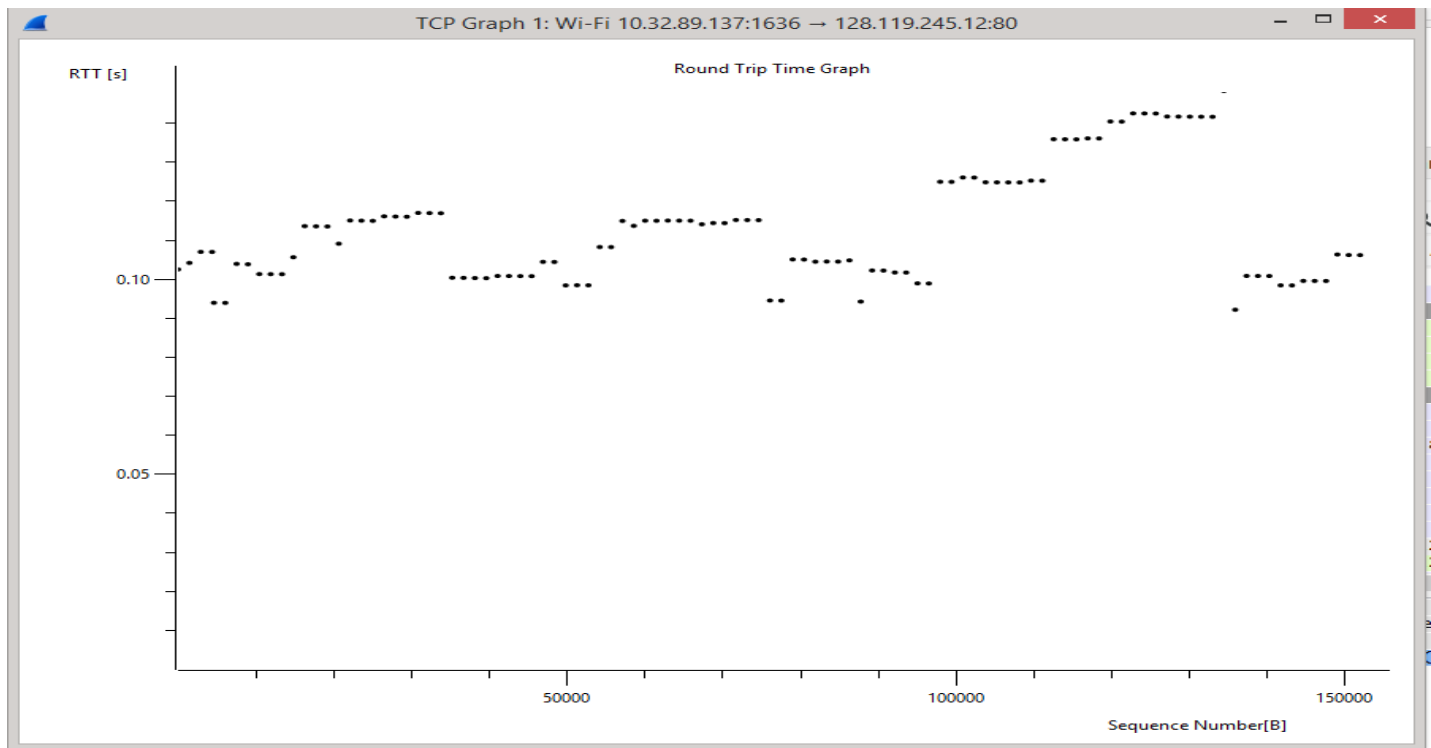
- What is the IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu? To answer this question, it's probably easiest to select an HTTP message and explore the details of the TCP packet used to carry this HTTP message, using the "details 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 source IP address was 10.32.89.137 using source port 1636..
- 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 destination IP address is 128.119.245.12 receiving on 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?
 - My IP address source is 10.32.39.137 sending on port 1636.

TCP BASICS

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 segment used to initiate the TCP connection is 0. The message contains a SYN flag indicating that it is a SYN segment (See screen capture 1)
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 SYNACK segment is 0.
 - The value of the acknowledgement field is 1. This value is determined by the initial sequence number +1.
 - The message carries flags that show it to be a SYN ACK message.
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 152074 (See screen capture 2).



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

Estimated RTT packet 1 : $0.875 * .028 + 0.125 * .028 = .028$

Estimated RTT packet 2 : $0.875 * .042 + 0.125 * .035 = .035$

Estimated RTT packet 3 : $0.875 * .054 + 0.125 * .070 = .070$

Estimated RTT packet 4 : $0.875 * .055 + 0.125 * .114 = .114$

Estimated RTT packet 5 : $0.875 * .077 + 0.125 * .140 = .140$

Estimated RTT packet 6 : $0.875 * .078 + 0.125 * .190 = .190$

Segment	Relative Segment Number	Segment Number	Time Sent	Acknowledgement Received	RTT	Estimated RTT
1	1	Odd601f	.026	.054	.028	.028
2	566	Odd6042	.042	.077	.035	.035
3	2026	Odd609d	.054	.124	.070	.070
4	3486	Odd60f9	.055	.169	.114	.114
5	4946	Odd60f9	.077	.217	.140	.140
6	6406	Odd61af	.078	.268	.190	.190

8. What is the length of each of the first six TCP segments?⁴
- Segment 1 = 565 bytes
 - Segment 2 = 1460 bytes
 - Segment 3 = 1460 bytes
 - Segment 4 = 1460 bytes
 - Segment 5 = 1460 bytes
 - Segment 6 = 1460 bytes
9. What is the minimum amount of available buffer space advertised at the receiver for the entire trace? Does the lack of receiver buffer space ever throttle the sender?
- The minimum amount of available buffer space is advertised as the window size: 5840 bytes. The lack of receiver buffer space does not ever throttle the sender
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. To check this, I looked for any repeating segment numbers
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 in an ack. If the data is doubled then that segment is acknowledging every other
12. What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.
- The file is 177851 bytes divided by the total time 7.596 seconds and average throughput is 23413.77 bytes per second

TCP CONGESTION CONTROL IN ACTION

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? Comment on ways in which the measured data differs from the idealized behavior of TCP that we've studied in the text.
- The slowstart phase begins at about zero and ends at about .15 seconds in according to the graph then congestion takes over.