Name - Ashutosh Soni Id – 2018UCP1505

**TCP**

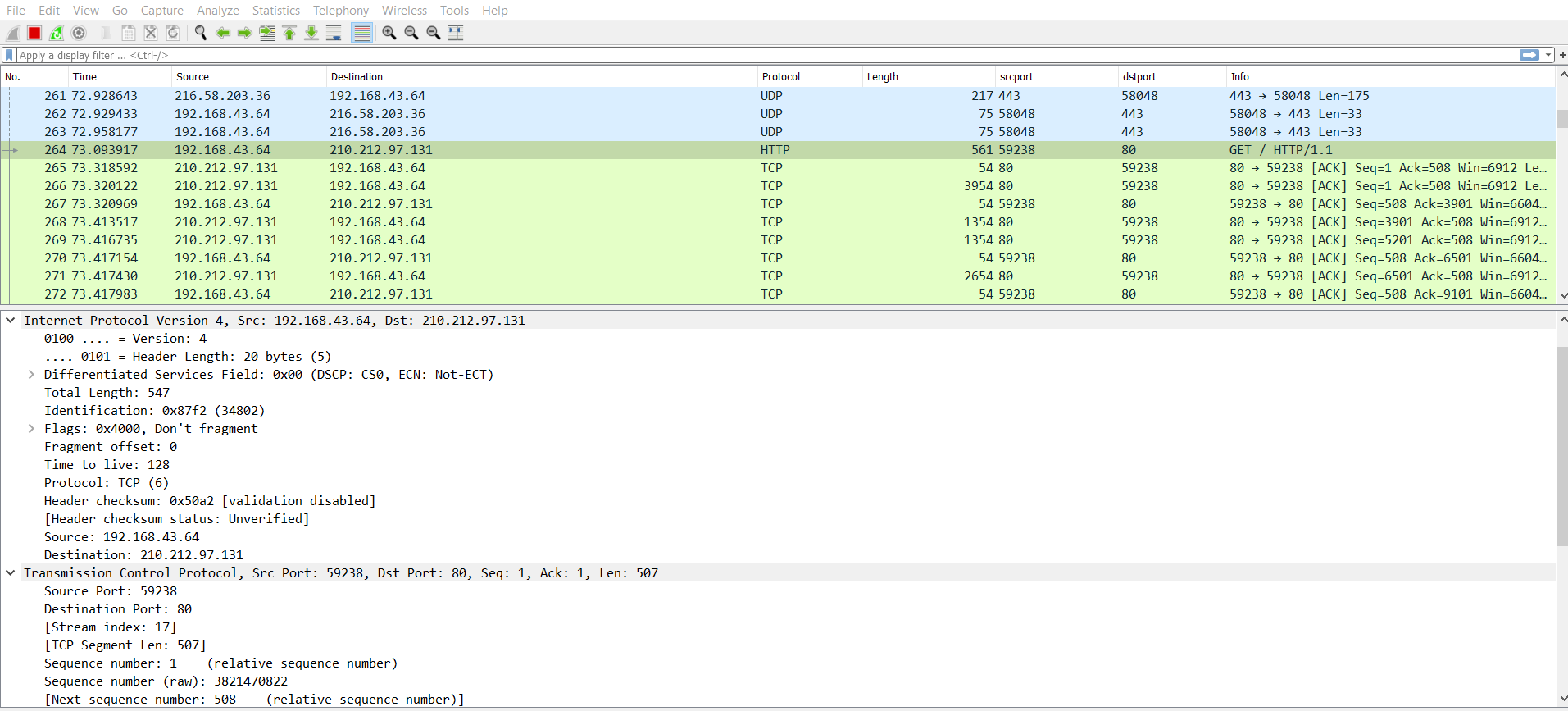
**Computer Networks**

---------------------------------------------------------------------------------------------------------

Q1: What is the IP address and TCP port number used by the client computer (source) that is transferring the file to mnit.ac.in? 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”.

Ans: IP address of source: 192.168.43.64

TCP port number of client computer: 59238



Q2: What is the IP address of mnit.ac.in? On what port number is it sending and receiving TCP segments for this connection?

Ans: From above screenshot we get that

IP address of destination (mnit.ac.in): 210.212.97.131

TCP port number of destination: 80

Q3: What is the IP address and TCP port number used by your client computer (source) to transfer the file to mnit.ac.in?

Ans: From above screenshot we get that

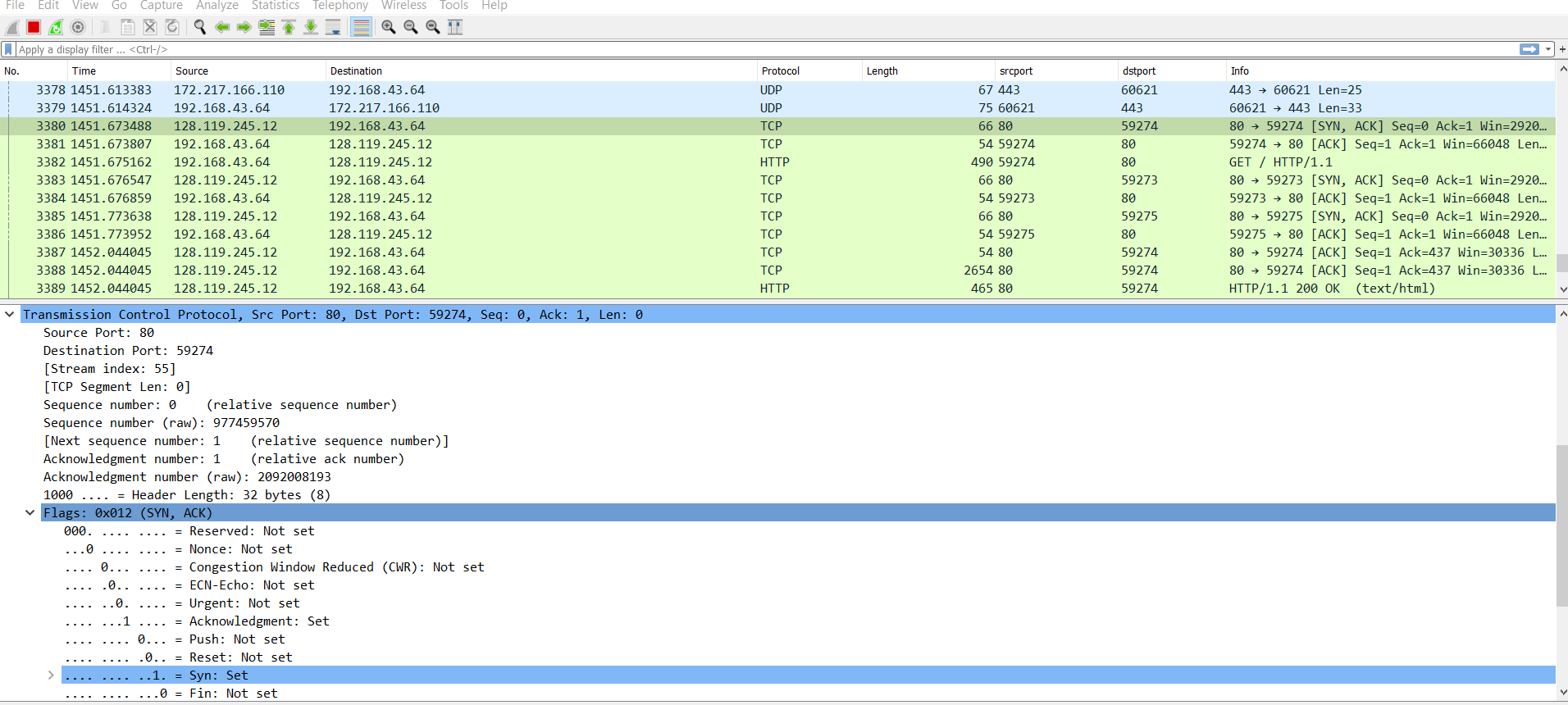
IP address of source: 192.168.43.64

TCP port number of client computer: 59238

Q4: 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?

Ans: Sequence number of the TCP SYN segment is used to initiate the TCP connection between the client and gaia.cs.umass.edu. This value is 0 in this trace.

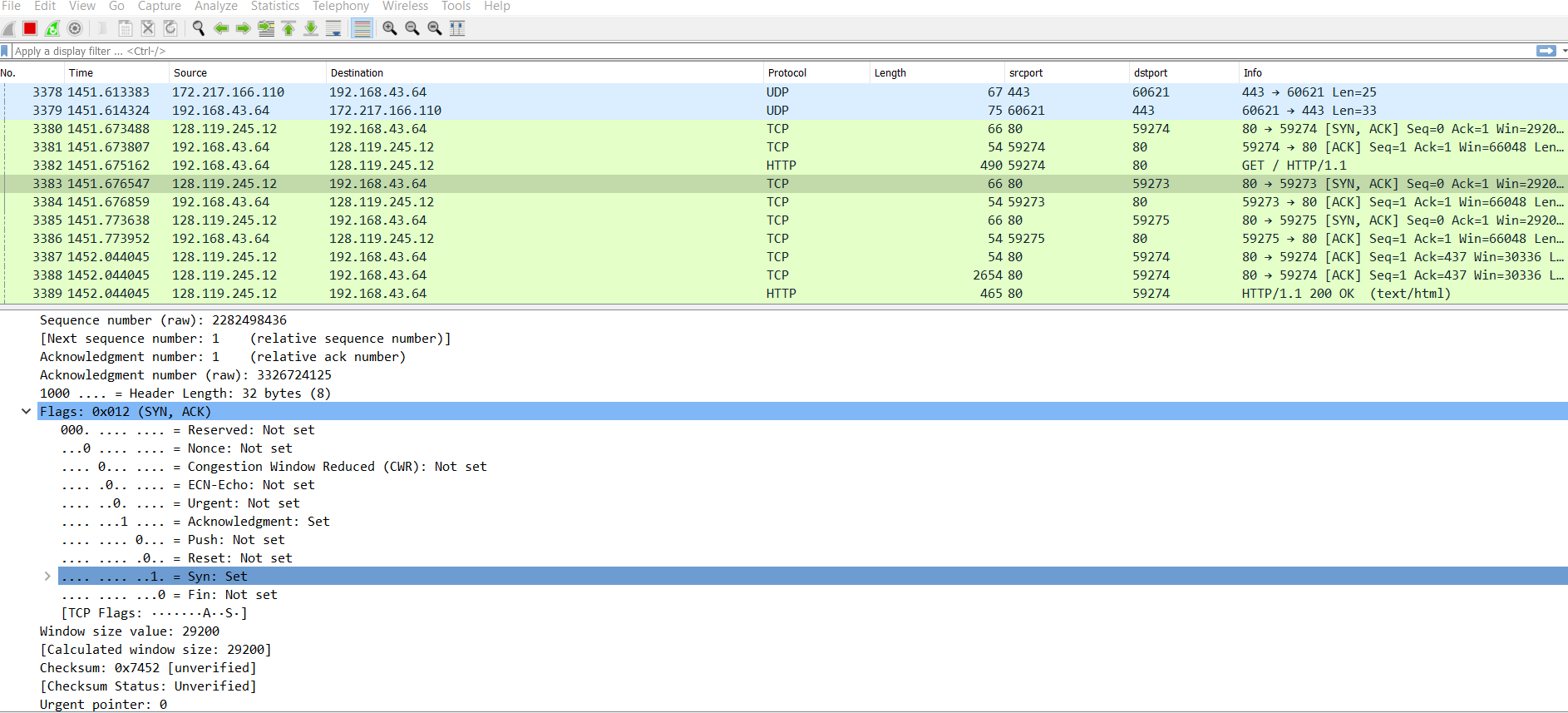
The SYN flag is set to 1 and it indicates that this segment is a SYN segment.



Q5: What is the sequence number of the SYNACK segment sent by mnit.ac.in to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment? How did mnit.ac.in determine that value? What is it in the segment that identifies the segment as a SYNACK segment?

Ans: Sequence number of the SYNACK segment from gais.cs.umass.edu to the client computer in reply to the SYN has he value of 0 in this trace.

The value of the Acknowledgement field in the SYNACK segment is 1. The value of the Acknowledgement field in the SYNACK segment is determined by gaia.cs.umass.edu by adding 1 to the initial sequence number of SYN segment from the client computer (i.e. the sequence number of the SYN segment initiated by the client computer is 0.).

 The SYN flag and Acknowledgement flag in the segment are set to 1 and they indicate that this segment is a SYNACK segment.

Q6: 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.

Ans: No. 4 segment is the TCP segment containing the HTTP POST command. The sequence number of this segment has the value of 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 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 for all subsequent segments.

Ans: The HTTP POST segment is considered as the first segment. Segment 1-6 are No. 4,5,7,8,10 and 11 in this trace respectively. The ACKs of segment 1-6 are No. 6,9,12,14,15 and 16 in this trace.

Segment 1 sequence number: 1

Segment 2 sequence number: 261

Segment 3 sequence number: 301

Segment 4 sequence number: 431

Segment 5 sequence number: 987

Segment 6 sequence number: 2012

* EstimatedRTT = 0.875\*EstimatedRTT + 0.125\*SampleRTT

EstimatedRTT for segment 1: 0.0125

EstimatedRTT for segment 2: 0.0223

EstimatedRTT for segment 3: 0.0331

EstimatedRTT for segment 4: 0.0411

EstimatedRTT for segment 5: 0.0564

EstimatedRTT for segment 6: 0.0753

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

Ans: Length of the first TCP segment (containing the HTTP POST): 565 bytes

Length of each of the other five TCP segments: 1460 bytes (MSS)

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?

Ans: The minimum amount of buffer space (receiver window) advertised at mnit.ac.in for the entire trace is 7459 bytes, which shows in the first acknowledgement form the server. The receiver window grows steadily until a maximum receiver buffer size of 62780 bytes. The sender is never throttled due to lacking of receiver buffer space by inspecting the trace.

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.

Ans: The acknowledged sequence of the ACKs are listed as follows.

ACK 1 acknowledged sequence no. - 566 acknowledged data 566

ACK 2 acknowledged sequence no. - 1021 acknowledged data 1460

ACK 3 acknowledged sequence no. – 2111 acknowledged data 1460

ACK 4 acknowledged sequence no. - 2981 acknowledged data 1460

ACK 5 acknowledged sequence no. - 3567 acknowledged data 1254

ACK 6 acknowledged sequence no. - 4121 acknowledged data 1460

ACK 7 acknowledged sequence no. - 6561 acknowledged data 1460

The difference between the acknowledged sequence number of two consecutive ACKs indicates the data received by the server between these two ACKs. By inspecting the amount of acknowledged data by each ACK, these are cases where the receiver is ACKing every other segment. For example, segment of No. 80 acknowledged data with 2920 = 1460\*2.

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

Select a TCP segment in the Wireshark’s “listing of captured-packets” window. Then select the menu : Statistics->TCP Stream Graph-> Time-Sequence Graph(Stevens).

Ans: The computation of TCP throughput largely depends on the selection of averaging time period. As a common throughput computation, in this question, we select the average time period as the whole connection time. Then, the average throughput for this TCP connection is computed as the ratio between the total amount data and the total transmission time. The total amount data transmitted can be computed by the difference between the sequence number of the first TCP segment (i.e. 1 byte for No. 4 segment) and the acknowledged sequence number of the last ACK (164091 bytes for No. 202 segment). Therefore, the total data are 164091 - 1 = 164090 bytes. The whole transmission time is the difference of the time instant of the first TCP segment (i.e., 0.026477 second for No.4 segment) and the time instant of the last ACK (i.e., 5.455830 second for No. 202 segment). Therefore, the total transmission time is 5.455830 - 0.026477 = 5.4294 seconds. Hence, the throughput for the TCP connection is computed as 164090/5.4294 = 30.222 KByte/sec.

**UDP**

1. Select one UDP packet from your trace. From this packet, determine how many fields there are in the UDP header.

Ans:

UDP header contains 4 fields:

1. source port; 2. destination port; 3. Length; 4. Checksum

2. By consulting the displayed information in Wireshark’s packet content field for this packet, determine the length (in bytes) of each of the UDP header fields.

Ans: The UDP header has a fixed length of 8 bytes. Each of these 4 header fields is 2 bytes long.

3.The value in the Length field is the length of what? Verify your claim with your captured UDP packet.

Ans: The length field specifies the number of bytes in the UDP segment (header plus data). An explicit length value is needed since the size of the data field may differ from one UDP segment to the next. The length of UDP payload for selected packet is 32 bytes. 40 bytes - 8 bytes = 32 bytes.

4. What is the maximum number of bytes that can be included in a UDP payload? (Hint: the answer to this question can be determined by your answer to 2)

Ans: The maximum number of bytes that can be included in a UDP payload is (2^16 – 1) bytes plus the header bytes. This gives 65535 bytes – 8 bytes = 65527 bytes.

5. What is the largest possible source port number? (Hint: see the hint in 4)

Ans: The largest possible source port number is (2^16 – 1) = 65535.

6. What is the protocol number for UDP? Give your answer in both hexadecimal and decimal notation. To answer this question, you’ll need to look into the Protocol field of the IP datagram containing this UDP segment.

Ans: The IP protocol number is 0x11, which is 17 in decimal above.

7. Examine a pair of UDP packets in which the first packet is sent by your host and the second packet is a reply to the first packet. Describe the relationship between the port numbers in the two packets

Ans: The source port of the UDP packet sent by the host is the same as the destination port of the reply packet, and conversely the destination port of the UDP packet sent by the host is the same as the source port of the reply packet.