**2. Analyzing UDP datagrams using Wireshark:**

• Start your web browser and clear the browser's cache memory, but do not access any website yet.

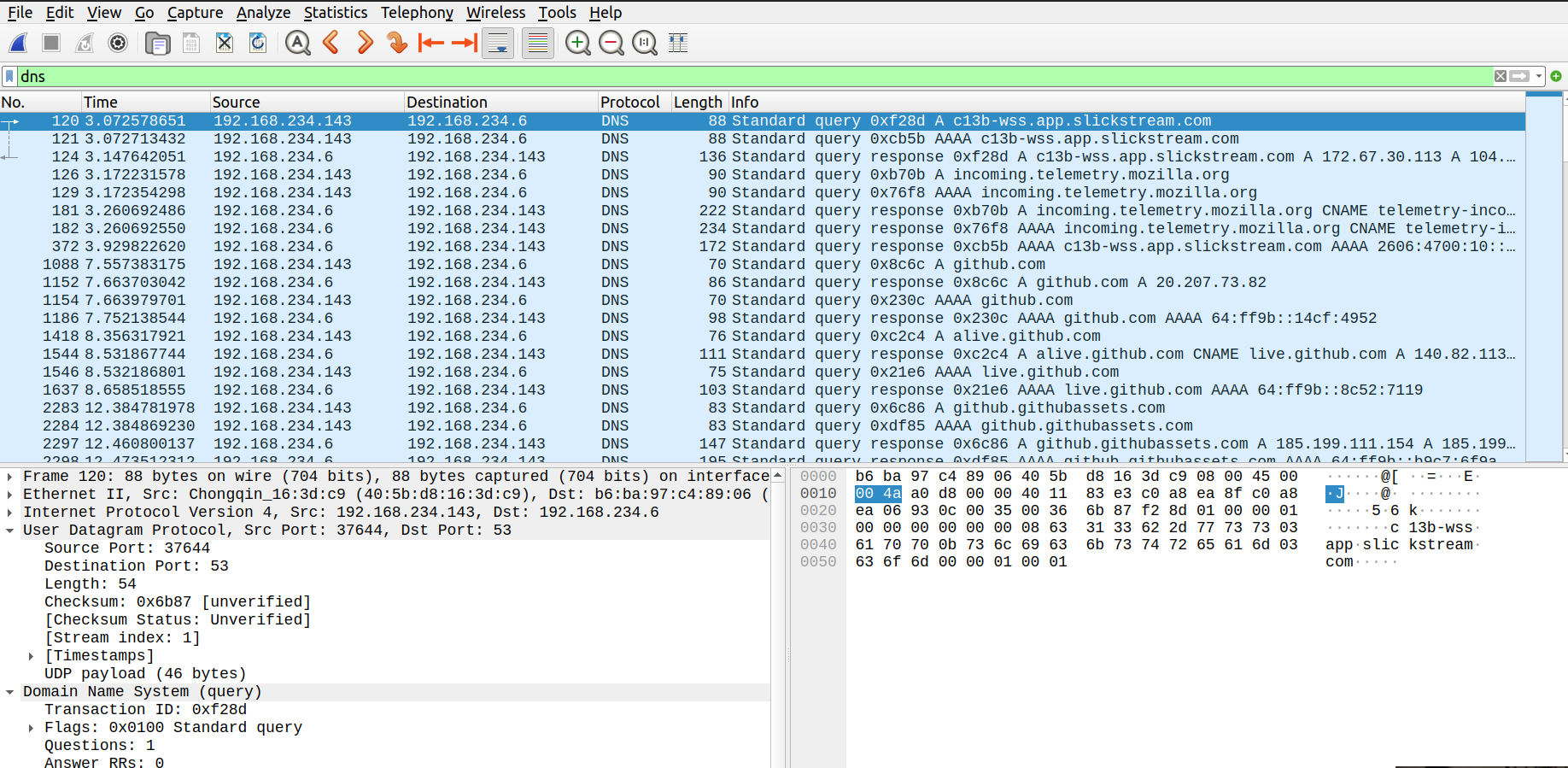
• Open Wireshark and start capturing.

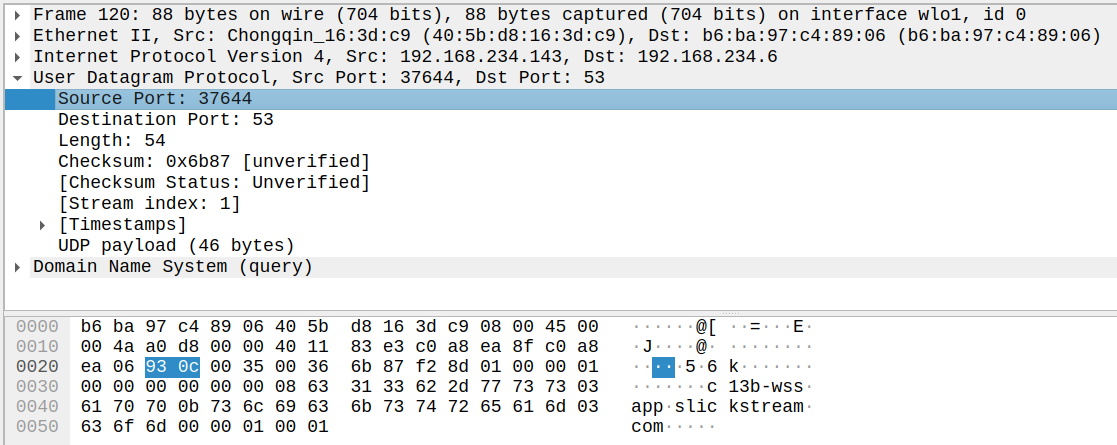
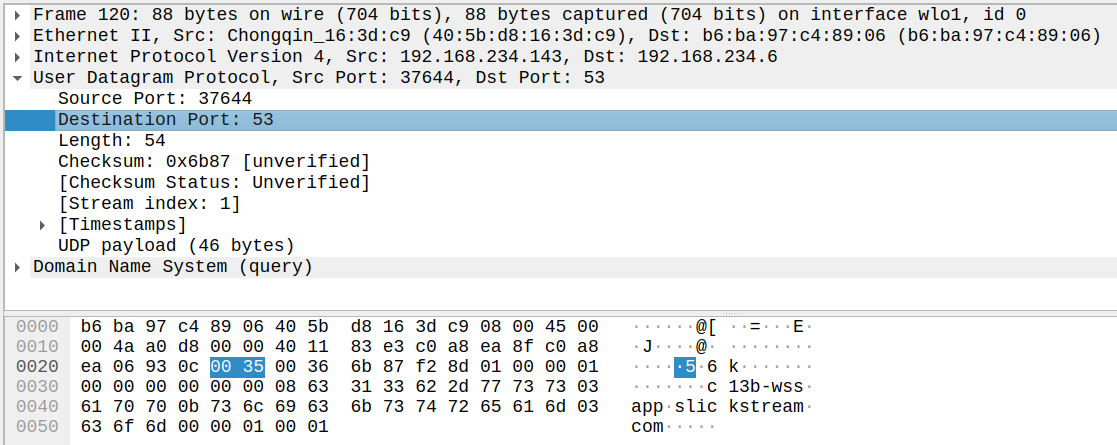
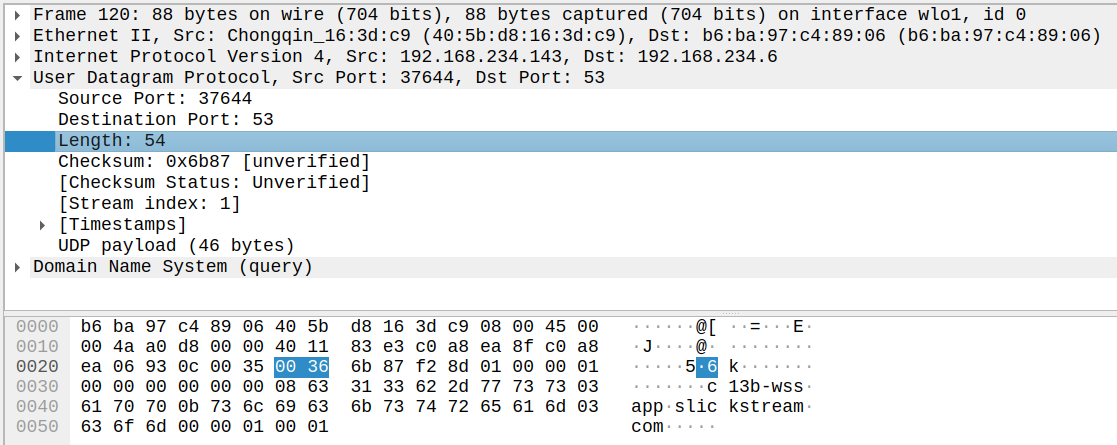
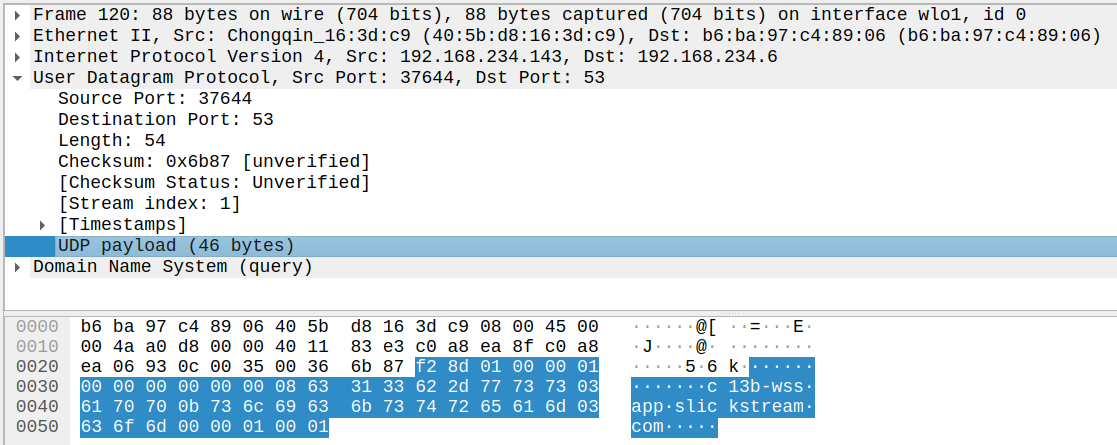
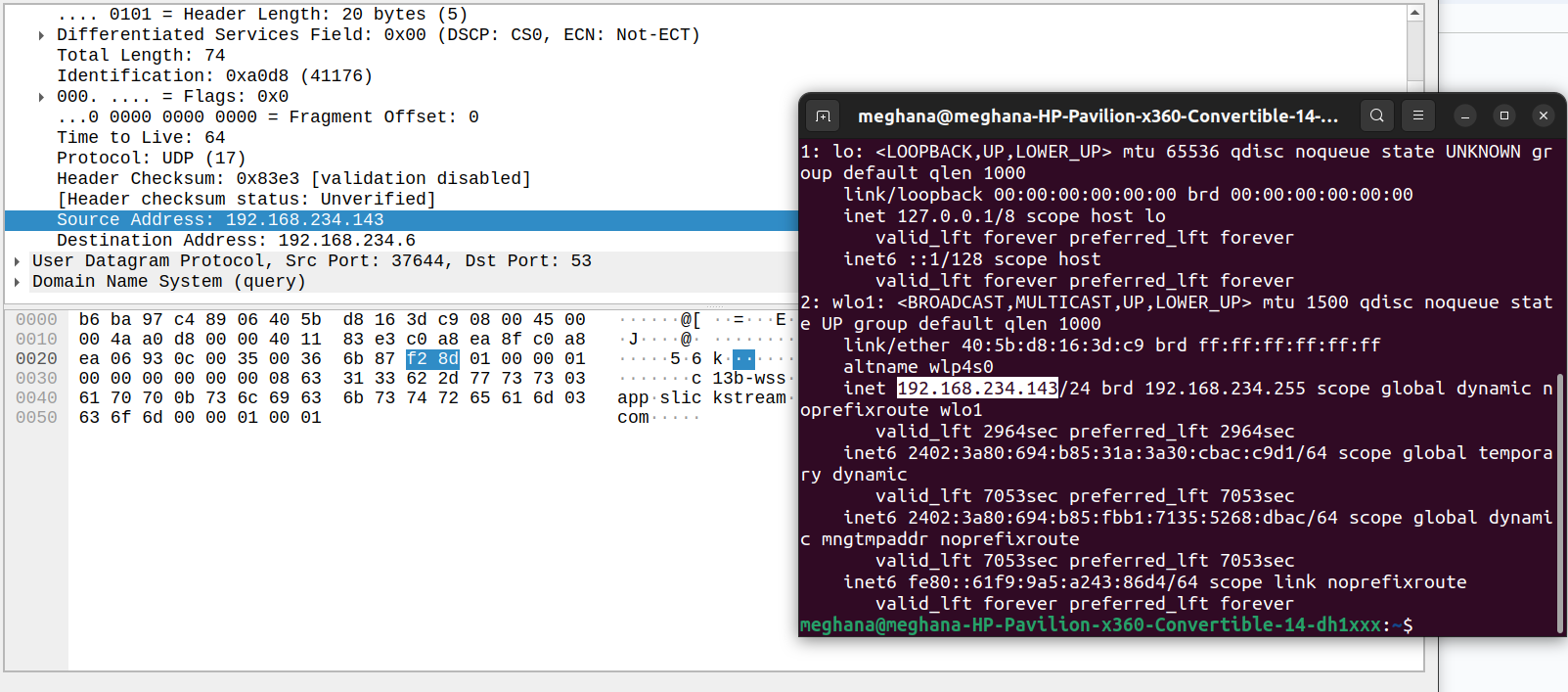
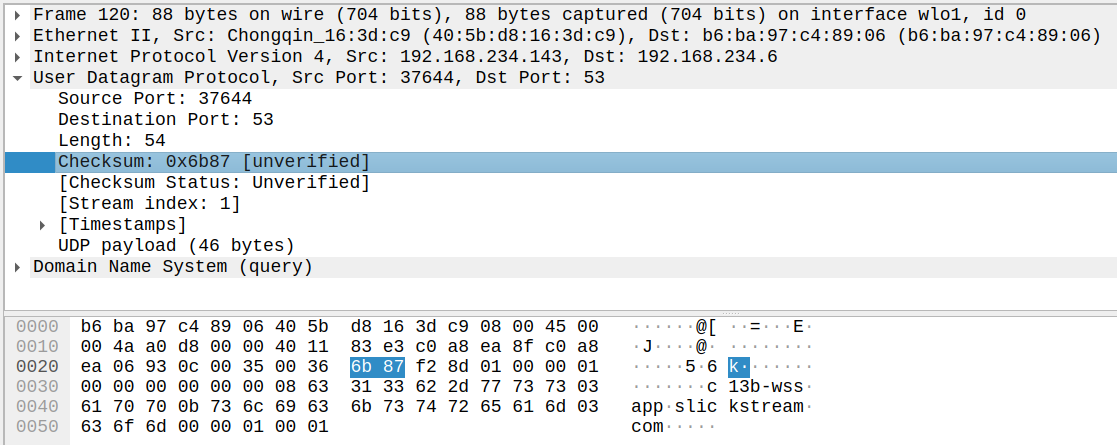
• Go back to your web browser and retrieve any file from a website. Wireshark starts capturing packets.

• After enough packets have been captured, stop Wireshark, and save the captured file.

• Using the captured file, analyse TCP & UDP packets captured. Note: DNS uses UDP for name resolution & HTTP uses TCP

Using the captured information, answer the following questions in your lab report.



* 1. **A.** In the packet list pane, select the first DNS packet. In the packet detail pane, select the **User Datagram Protocol**. The UDP hexdump will be highlighted in the packet byte lane. Using the hexdump, Answer the following:
  2. **a.** the source port number.
  3. 
  4. 37644
  5. In hex:93 0c
  6. **b.** the destination port number.
  7. 
  8. 53
  9. In hex:00 35
  10. **c.** the total length of the user datagram.
  11. 
  12. 54 bytes
  13. In hex:00 36
  14. **d.** the length of the data.
  15. 
  16. 46 bytes
  17. **e.** whether the packet is directed from a client to a server or vice versa.
  18. 
  19. Client to server (192.168.234.143 is my IP address)
  20. **f.** the application-layer protocol.
  21. DNS (port 53)
  22. **g.** whether a checksum is calculated for this packet or not.
  23. 
  24. Yes, the value is 0x6b87(but it is not verified)

1. **B.** What are the source and destination IP addresses in the DNS query message? What are those addresses in the response message? What is the relationship between the two?
   1. Source: 192.168.234.143
   2. Destination: 192.168.234.6
   3. The source and destination machines in the query message exchange in the response message because in the query the client was the source but in the response the server is the source. Hence the IP addresses also seem to exchange.
2. **C.** What are the source and destination port numbers in the query message? What are those addresses in the response message? What is the relationship between the two? Which port number is a well-known port number?

Source port: 37644

Destination port: 53

* 1. 192.168.234.143 is the source IP address (and server IP address) and 192.168.234.6 is the destination address (and client IP address).
  2. The source and destination machines in the query message exchange in the response message because in the query the client was the source but in the response the server is the source. Hence the IP addresses also seem to exchange.

53 is the well-known port number. It is the port for DNS.

1. **D.** What is the length of the first packet? How many bytes of payload are carried by the first packet?

Length: 54

Payload: 46 bytes

**2b. Analyzing TCP packets using Wireshark:**

Start your web browser and clear the browser's cache memory, but do not access any website yet.

■ Open Wireshark and start capturing.

■ Go back to your web browser and retrieve any file from a website. Wireshark starts capturing packets.

■ After enough packets have been captured, stop Wireshark and save the captured file.

■ Using the captured file, select only those packets that use the service of TCP. For this purpose, type **tcp**(lowercase) in the *filter field* and press **Apply**. The packet list pane of the Wireshark window should now display a bunch of packets.

***Part I: Connection-Establishment Phase***

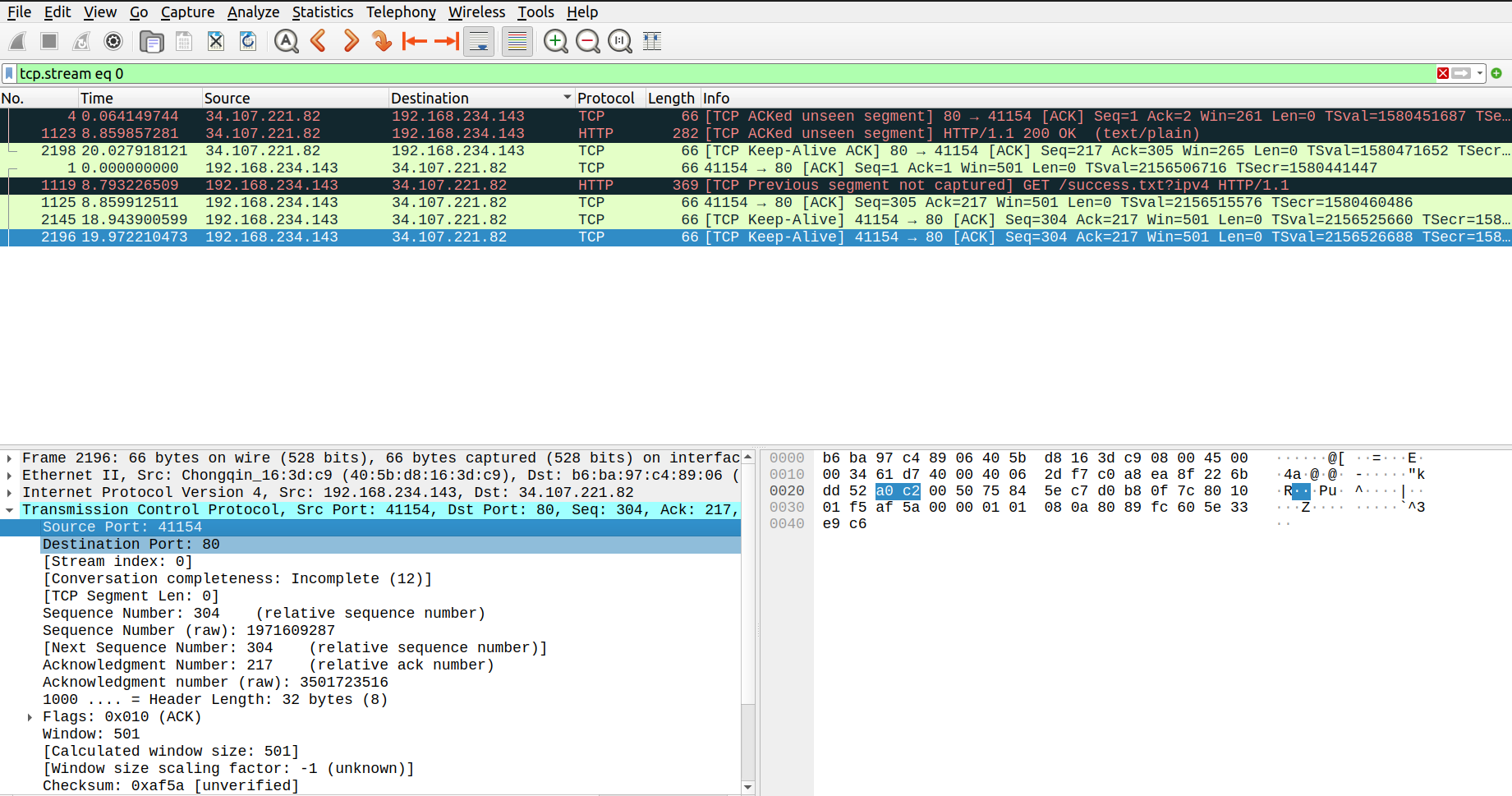
Identify the TCP packets used for connection establishment. Note that the last packet used for connection establish may have the application-layer as the source protocol.

The highlighted packets are used for connection establishment.

***Questions***

Using the captured information, answer the following question in your lab report about packets used for connection establishment.

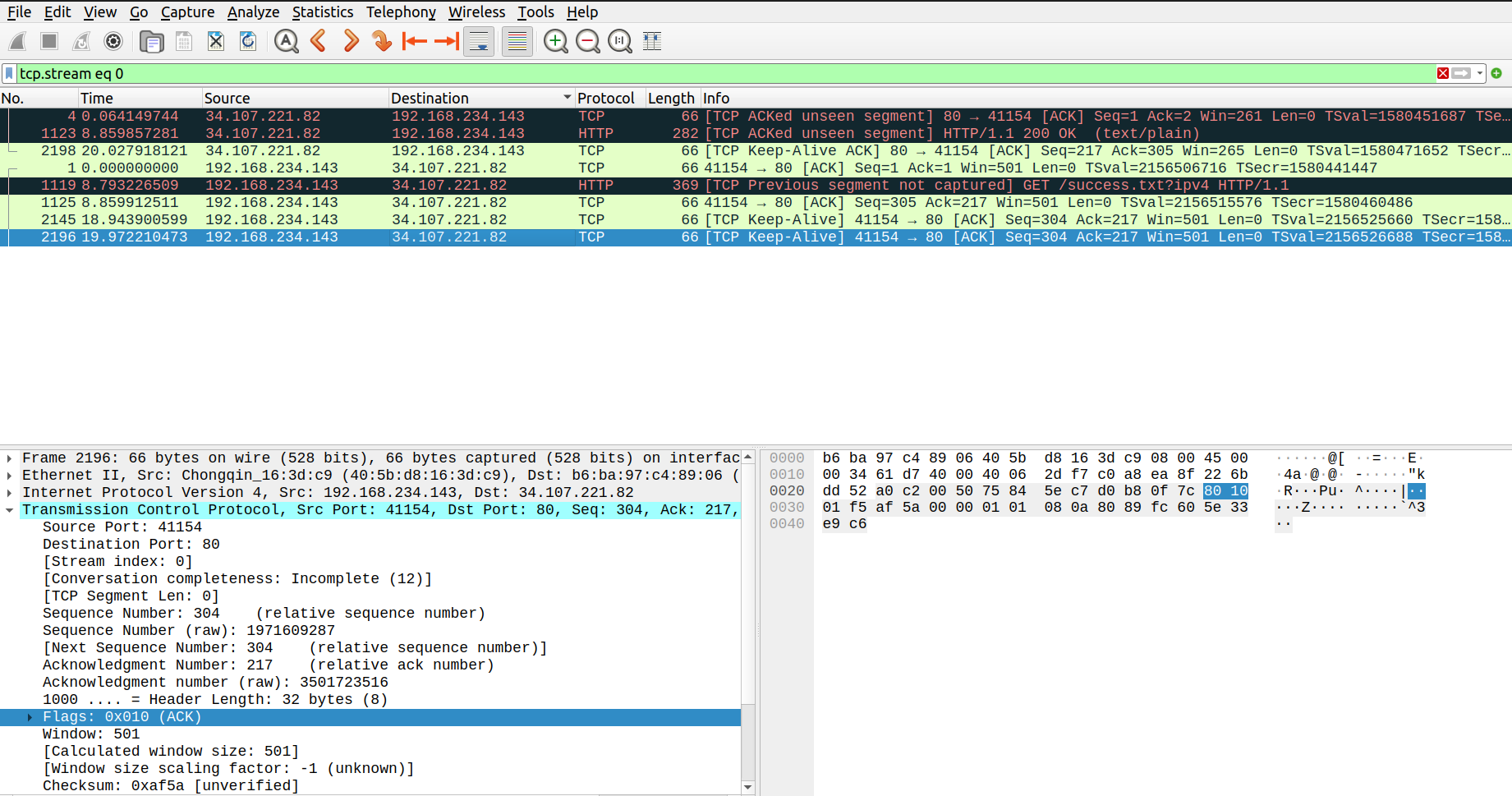
**1.** What are the socket addresses for each packet?



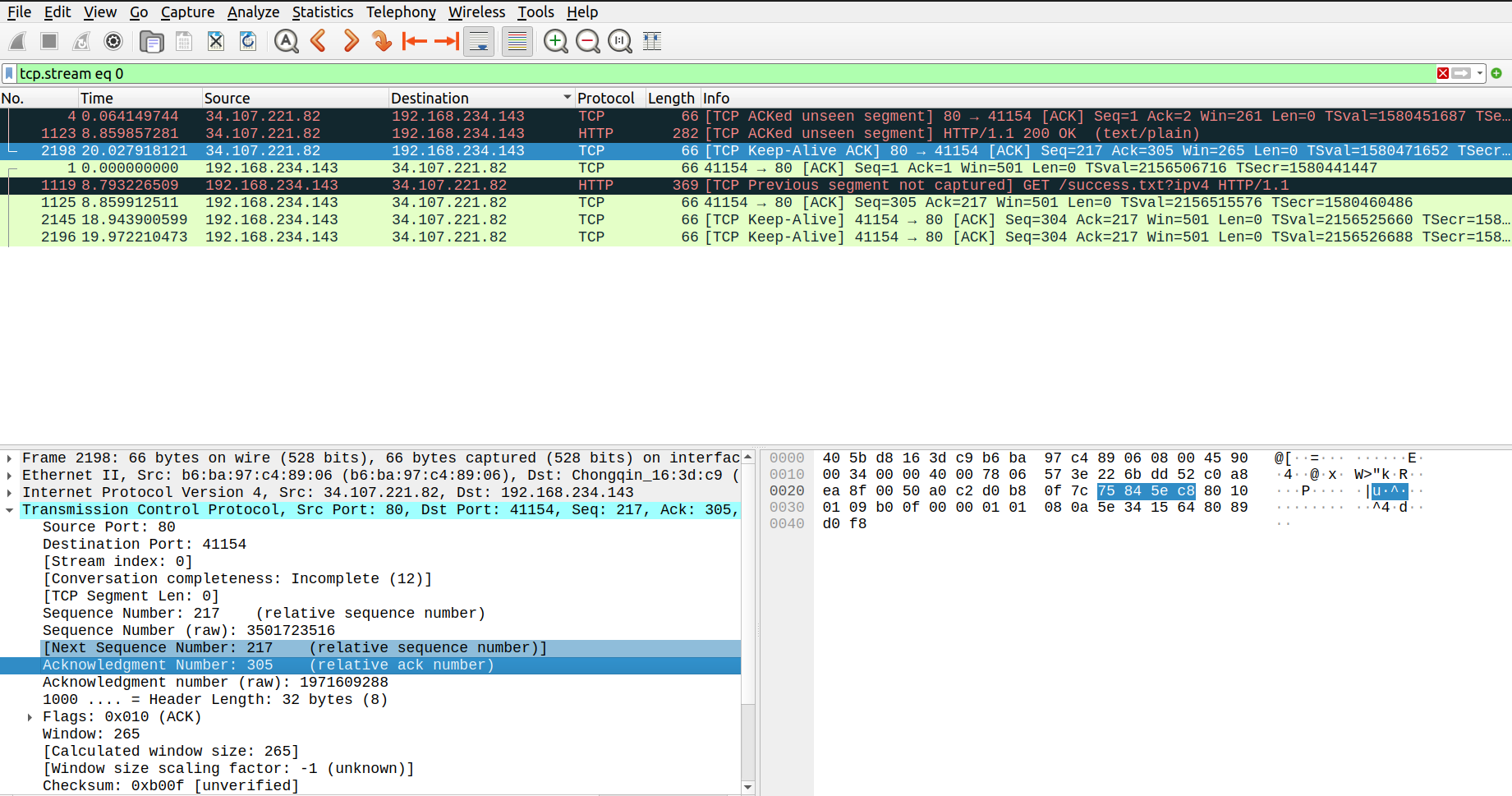
Socket address on server side: 192.168.234.143:80

Socket address on client side: 34.107.221.82:41154

**2.** What flags are set in each packet?



The flag set in each packet is 0x010(ACK)

**3.** What are the sequence number and acknowledgment number of each packet?

Packet no. : 1 => Sequence no. : 1,Acknowledgement no. : 1

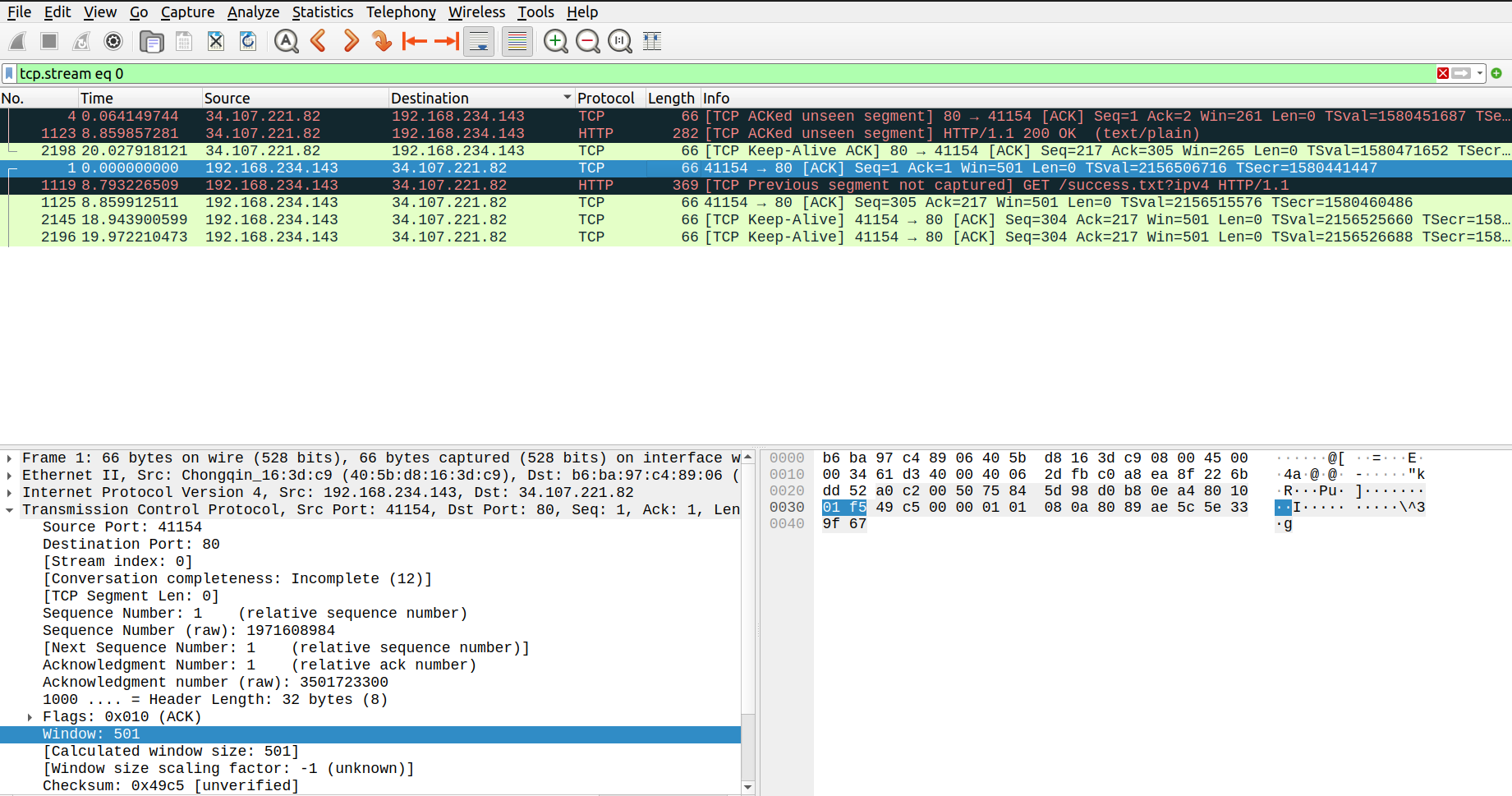
Packet no. : 1125 => Sequence no. : 305,Acknowledgement no. : 217

Packet no. : 2145 => Sequence no. : 304,Acknowledgement no. : 217

Packet no. : 2196 => Sequence no. : 304,Acknowledgement no. : 217

Packet no. : 2198 => Sequence no. : 217,Acknowledgement no. : 305

**4.** What are the window size of each packet?



Packet no. : 1,1125,2145,2196 all have Window size 501.

Packet no. : 2198 => Window size : 265

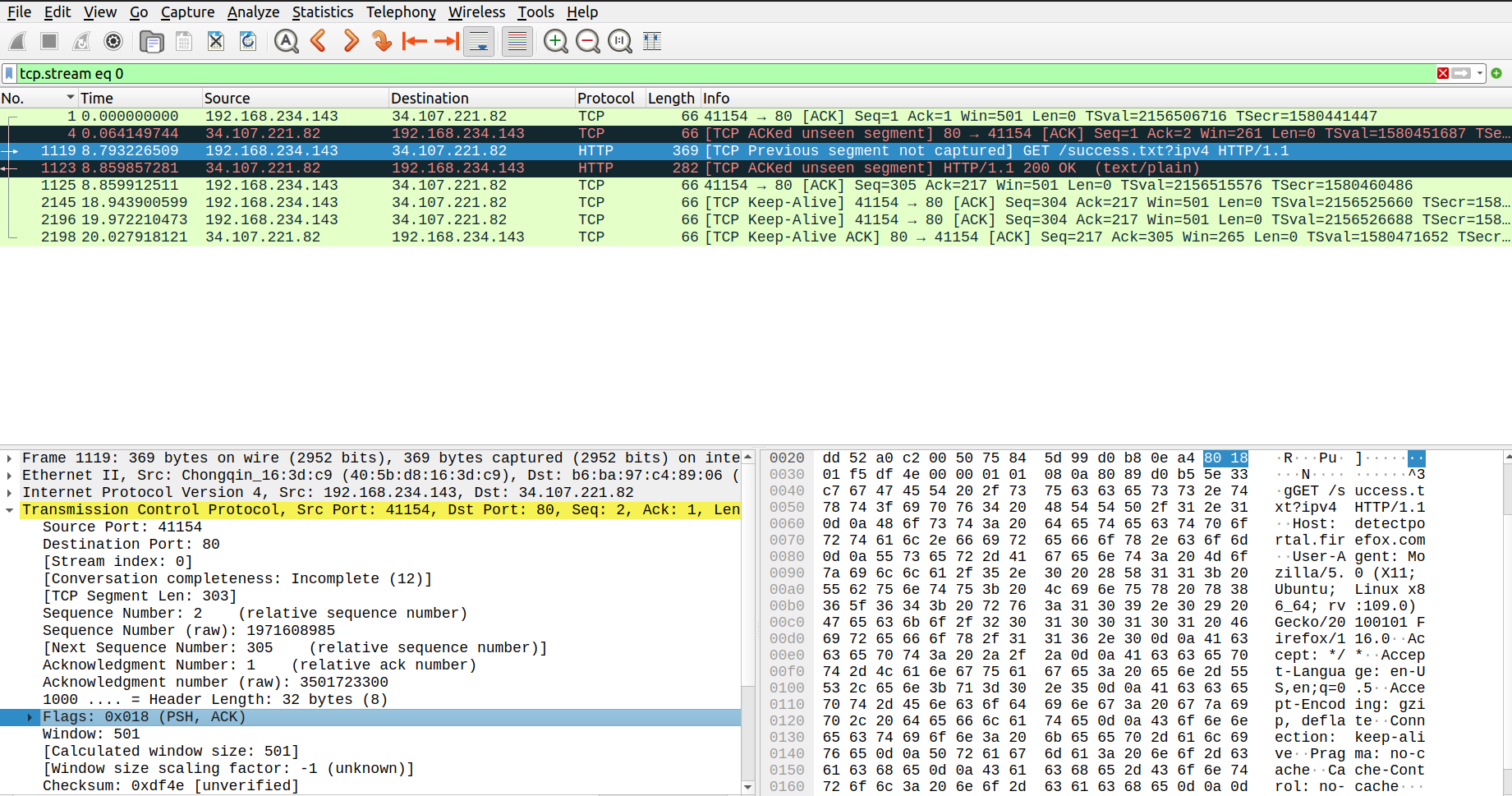
***Part II: Data-Transfer Phase***

The data-transfer phase starts with an HTTP GET request message and ends with an HTTP OK message.

***Questions***

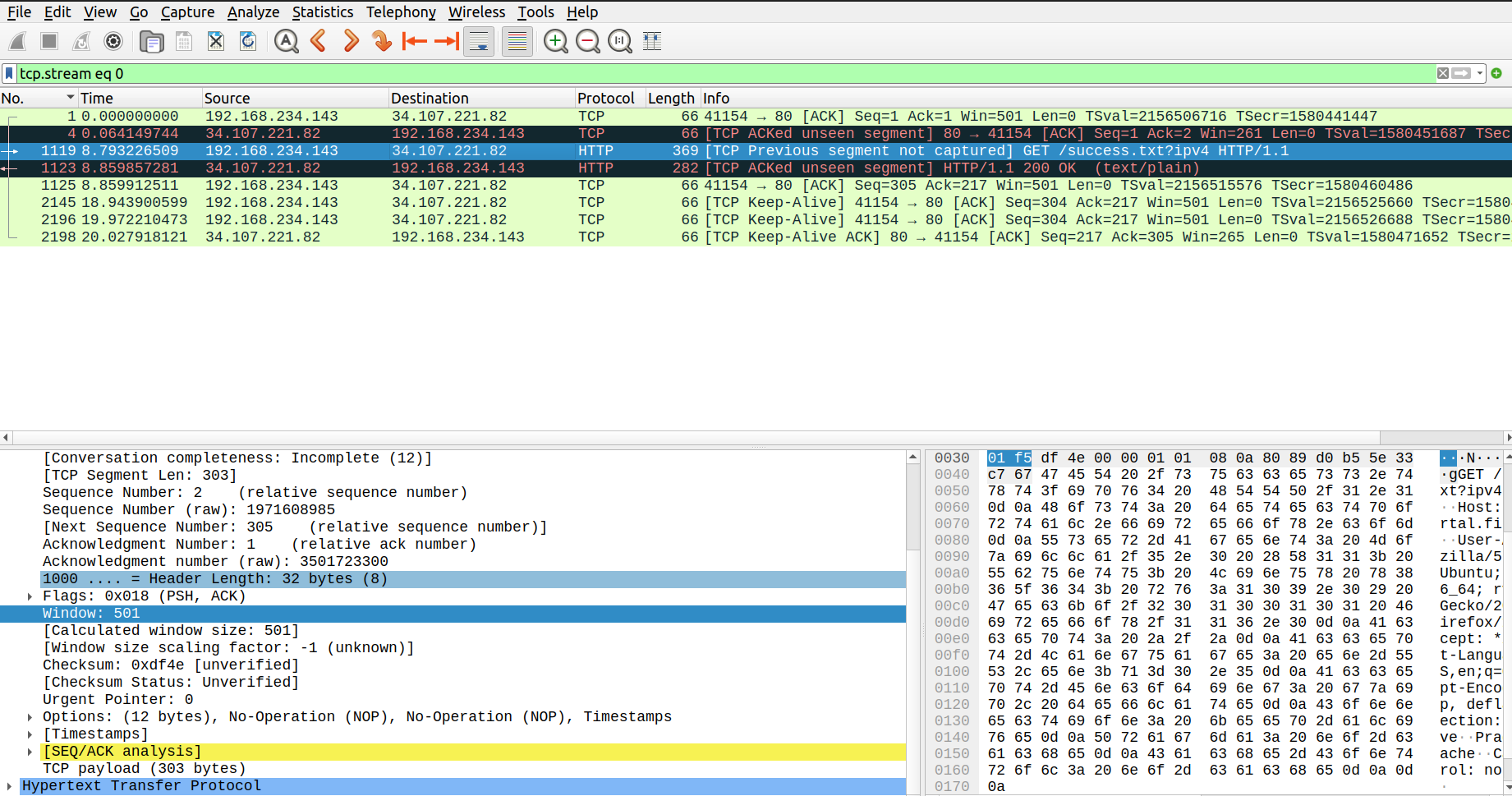
Using the captured information, answer the following question in your lab report about packets used for data transfer.

**1.** What TCP flags are set in the first data-transfer packet (HTTP GET message)?

Flags : 0x018(PSH,ACK)

Push and Acknowledgement

**2.** How many bytes are transmitted in this packet?

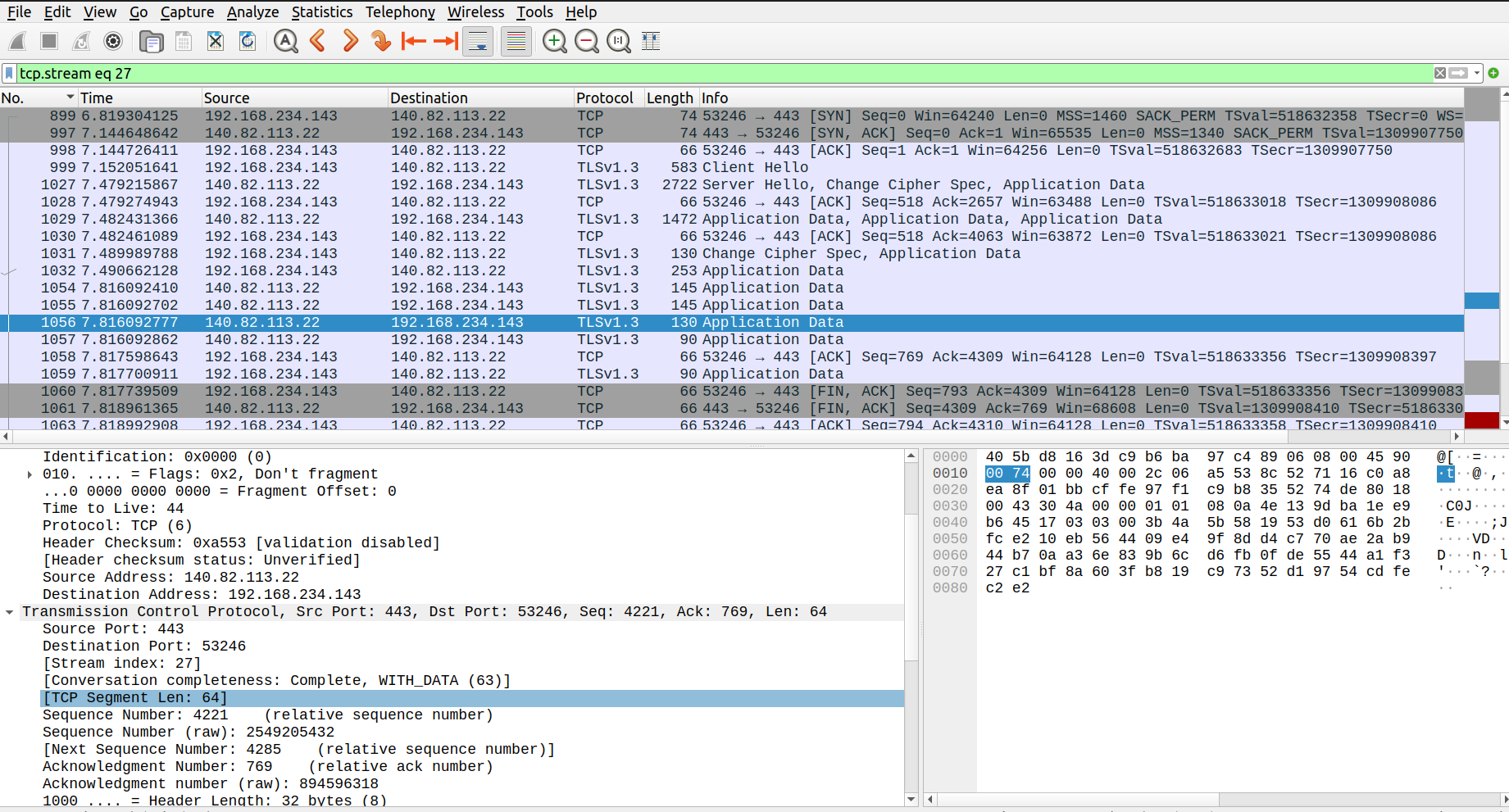


TCP Payload:303 bytes

**3.** How often does the receiver generate an acknowledgment? To which acknowledgment rule (defined in Page 200 in the textbook) does your answer correspond to?

**4.** How many bytes are transmitted in each packet? How are the sequence and acknowledgment numbers related to number of bytes transmitted?

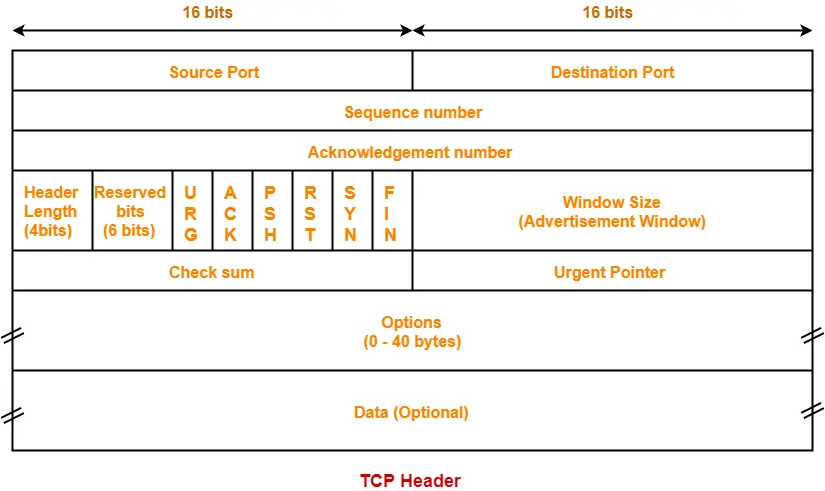
The stream 0 seemed to have only ACK flags,so there was no data transfer that was completed and no connection termination,hence I switched to a different stream(stream 27) to check this.



Bytes transmitted = Total length – header length of IP – header length of TCP

The first 3 packets have TCP segement length 0(0 bytes transferred),packet 999 transmitted 517 bytes,packet 1027 transmitted 2656 bytes,packet 1028 transmitted 0 bytes,packet 1029 transmitted 1406 bytes,1030 transmitted 0 bytes and so on.

Sequence number and Acknowledgement no. each form 32 bits(4 bytes) of the TCP header which is used in the calculation of bytes transmitted.The sequence number increases by the number of bytes transmitted, and the acknowledgment number increases by the number of bytes received.



**5.** What are the original window sizes that are set by the client and the server? Are these numbers expected? How do they change as more segments are received by the client?

Client:501

Server:261

In the TCP protocol, during the initial connection establishment phase, both the client and the server negotiate and set the initial window size for data transmission.

The values change based on buffer space available. Window size increases when buffer space increases and decreases when buffer space decreases. If more of the received segments are in the buffer, the window size decreases, and as the segments are consumed by the receiver, buffer size increases.

**6.** Explain how the window size is used in flow control?

Flow control is a mechanism used in networking to manage the amount of data that can be sent from the sender to the receiver, ensuring that the receiver can handle the incoming data without becoming overwhelmed.

The window size in flow control helps regulate the rate of data transmission between a sender and a receiver. It prevents scenarios where the sender sends data faster than the receiver can process, leading to data loss or congestion. By using the window size mechanism, TCP ensures efficient and reliable data transfer even in varying network conditions.

**7.** What is the purpose of the HTTP OK message in the data transfer phase?

The purpose of the "HTTP 200 OK" response in the data transfer phase is to indicate that the client's request was valid, the server could fulfill it, and the client can now expect the requested data to be included in the response. It's an acknowledgment that the server has completed its part of the transaction successfully, and the client can proceed to use the received data.

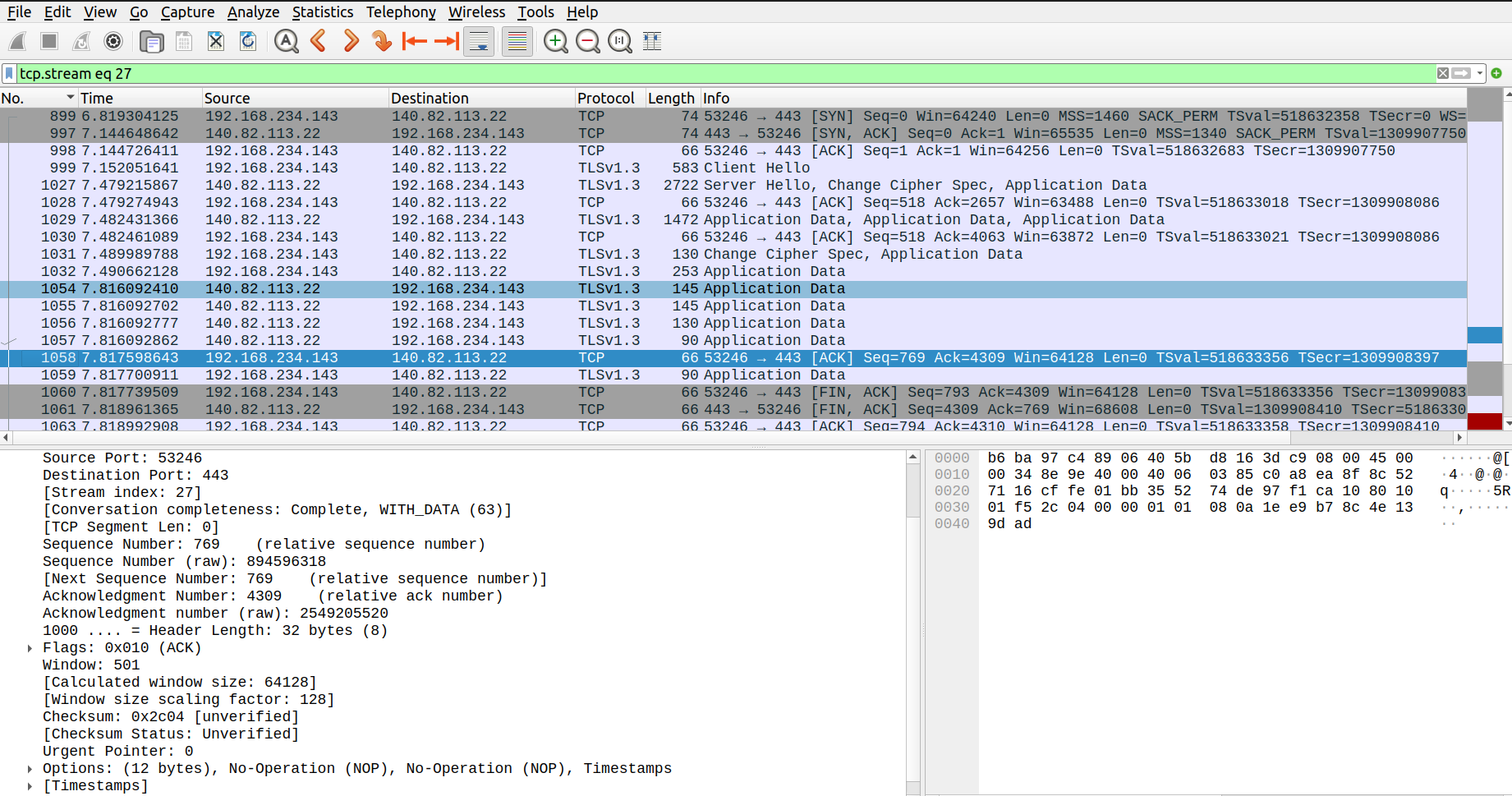
***Part III: Connection Termination Phase***

The data-transfer phase is followed by the connection termination phase. Note that some packets used in the connection-termination phase may have the source or sink protocol at the application layer. Find the packets used for connection termination.

***Questions***

Using the captured information, answer the following question in your lab report about packets used for connection termination.

**1.** How many TCP segments are exchanged for this phase?

 There are 97 segments exchanged for this phase. Each packet represents a TCP segment exchanged during the connection termination phase.

**2.** Which end point started the connection termination phase?

The client (ip address:192.168.234.143, port no.:53246) started the connection termination phase.

**3.** What flags are set in each of segments used for connection termination?

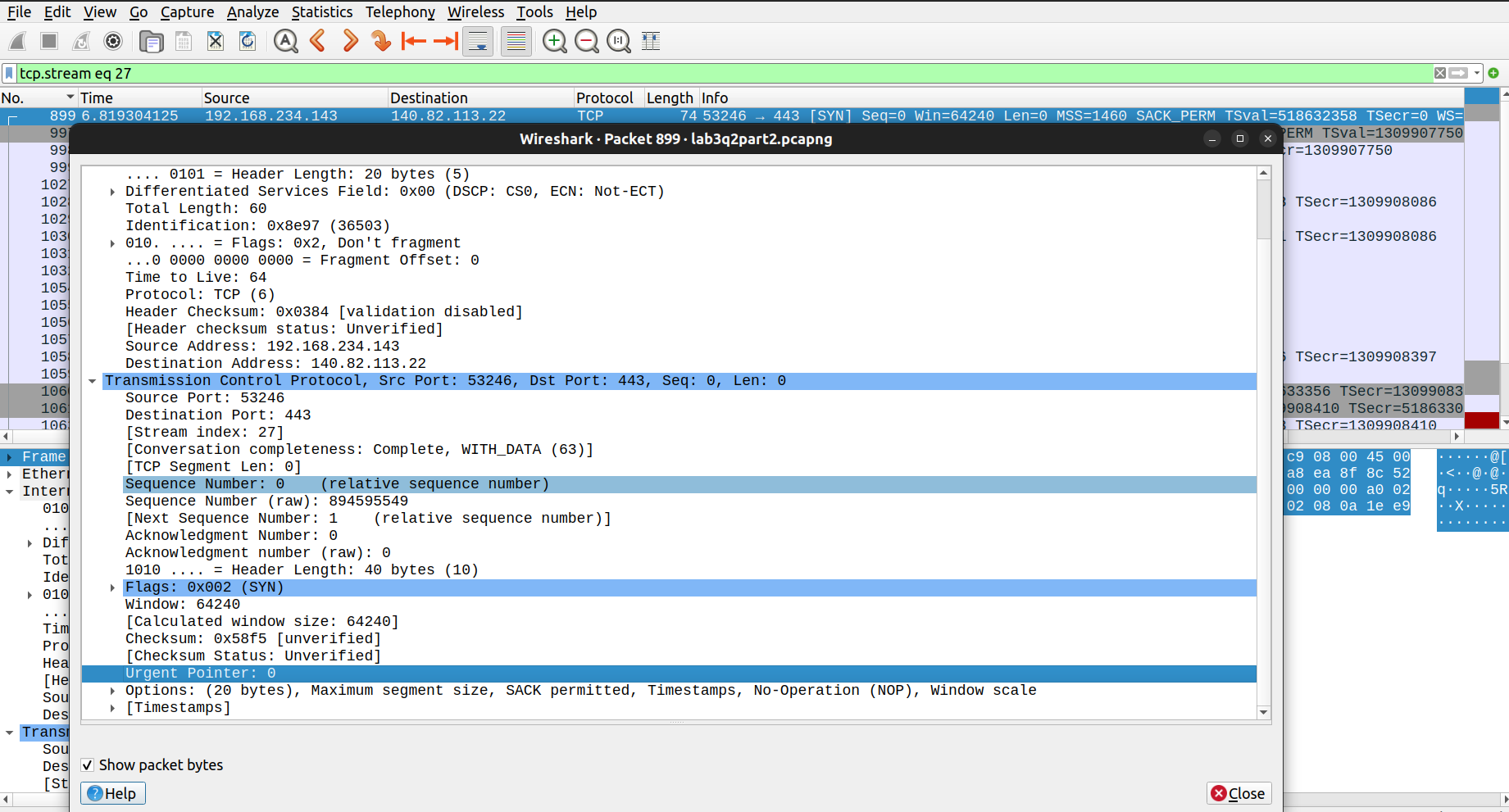
The flags used are 0x002(SYN),0x012(SYN,ACK),0x010(ACK),0x011(FIN,ACK),0x004(RST)

**II. LAB EXERCISES**

1. From the captured information, answer the following question in your lab report.

i. Using the hexdump, determine the following for any TCP packet:

The source port number, the destination port number, the sequence number, the acknowledgment number, the header length, the set flags, the window size and the urgent pointer value.



ii. Using the information in the detail pane lane, verify your answers is question 1.

Source port:53246

Destination Port:443

Seq no. : 0

Ack no. : 0

Header length:40 bytes

Flags: 0x002(SYN)

Window size:64240

Urgent Pointer:0

iii. Does any of the TCP packet header carry options? Explain your answer.

iv. What is the size of a TCP packet with no options. What is the size of a TCP packet with options?

v. Is window size in any of the TCP packet zero? Explain your answer.

2. Analyze Interaction between your TCP Client-Server Programs using wireshark

3. Analyze Interaction between your UDP Client-Server Programs using wireshark