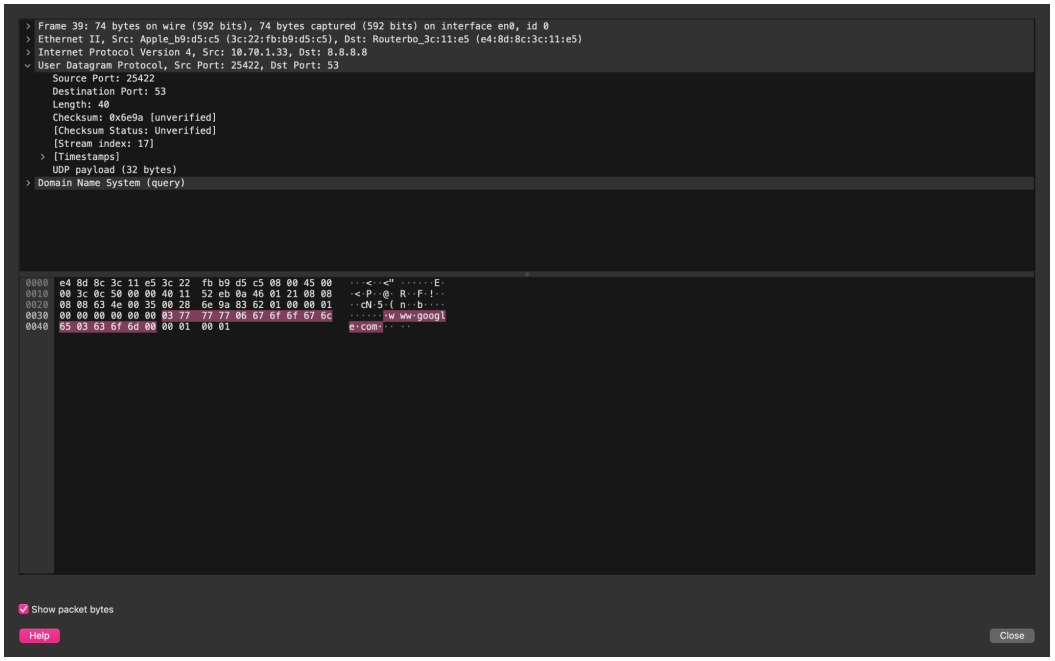
Lab-3 **Aditi Shrivastava - 210905244**

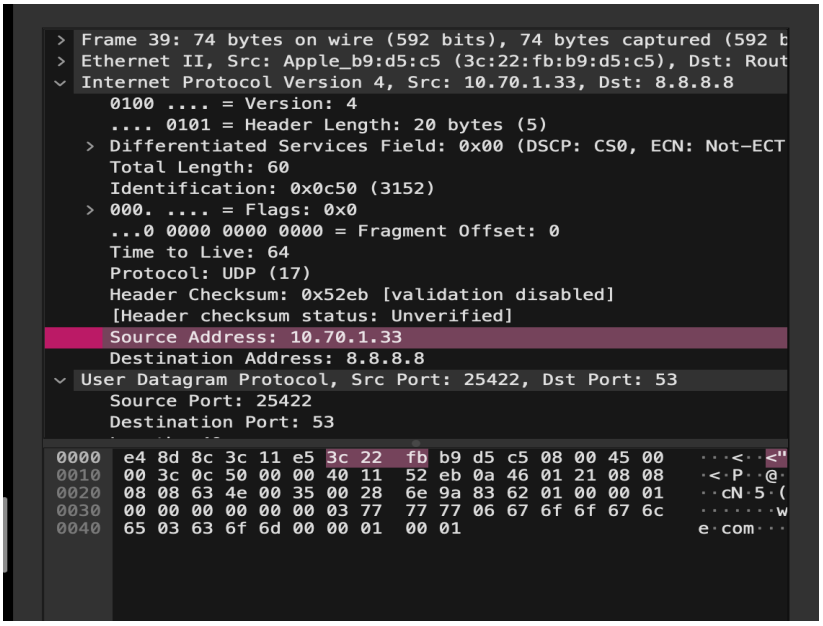
Socket Programming in 'C' using UDP and Network Monitoring and Analysis with Wireshark

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:

a. the source port number. 25422
b. the destination port number. 53
c. the total length of the user datagram. 72
d. the length of the data. 40
e., Whether the packet is directed from a client to a server or vice versa. - client to server
f. the application-layer protocol.- UDP
g. whether a checksum is calculated for this packet or not. - 0x6e9a



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?



Answer:
Source Address: 10.70.1.33
Destination Address: 8.8.8.8
In response, the addresses are exactly inverted. The source becomes the destination, and the destination becomes the source.

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?
Answer:
Source and destination port numbers in query: 25422, 53
Source and destination port numbers in response: 53, 25422
The standard port for DNS:- 53

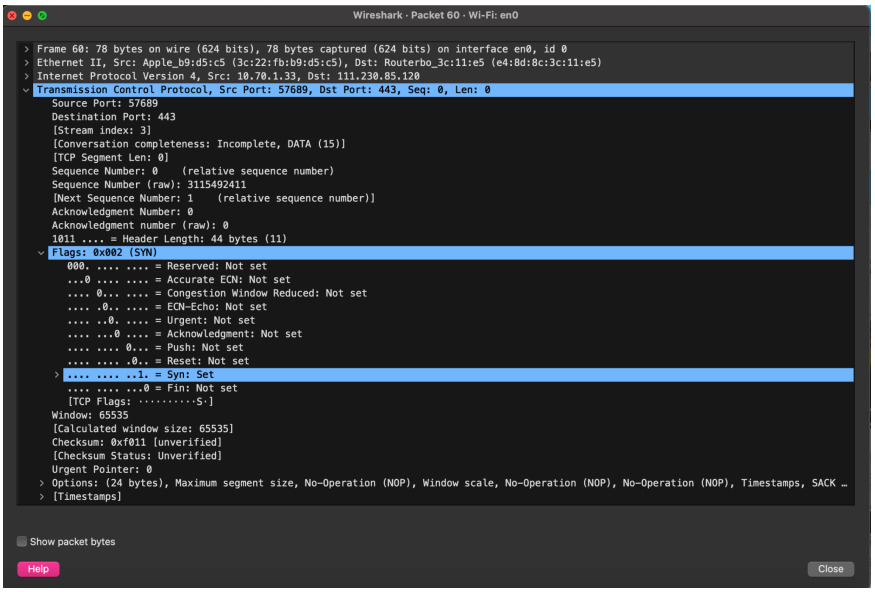
D. What is the length of the first packet? How many bytes of payload are carried by the first packet?
Answer:
The total length of the query is:- 72
UDP Payload is 32 Bytes

Part I: Connection-Establishment Phase

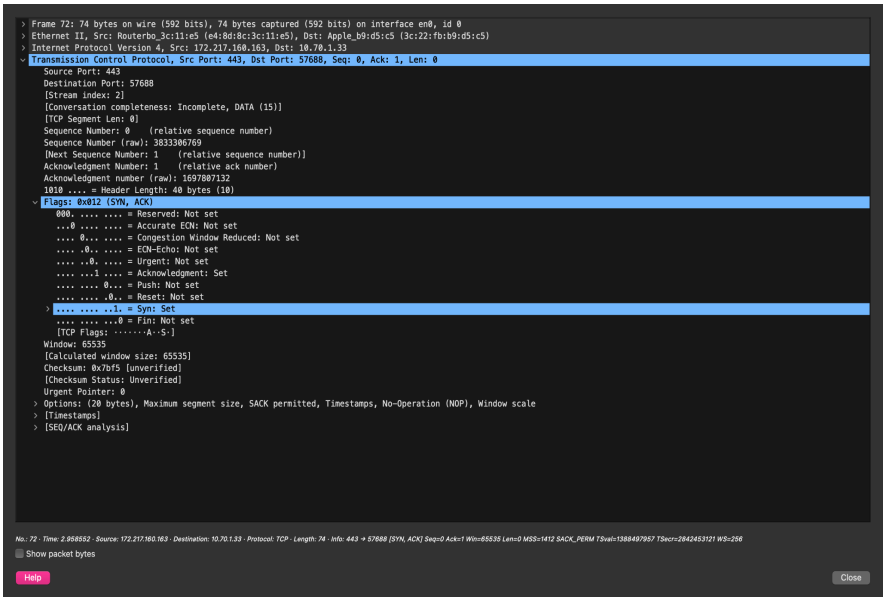
Identify the TCP packets used for connection establishment. Note that the last packet used for connection establishment may have the application layer as the source protocol.

Whenever a TCP conversation needs to occur, the client initiates and tries to make a connection. The server is passively open and always listening for connections. A packet sent to the server by the client begins an active open handshake. This is the 3-way handshake, named as such because 3 steps have to occur to bring up a connection.
A TCP connection establishment consists of a 3-way handshake. (SYN , SYN-ACK , ACK)

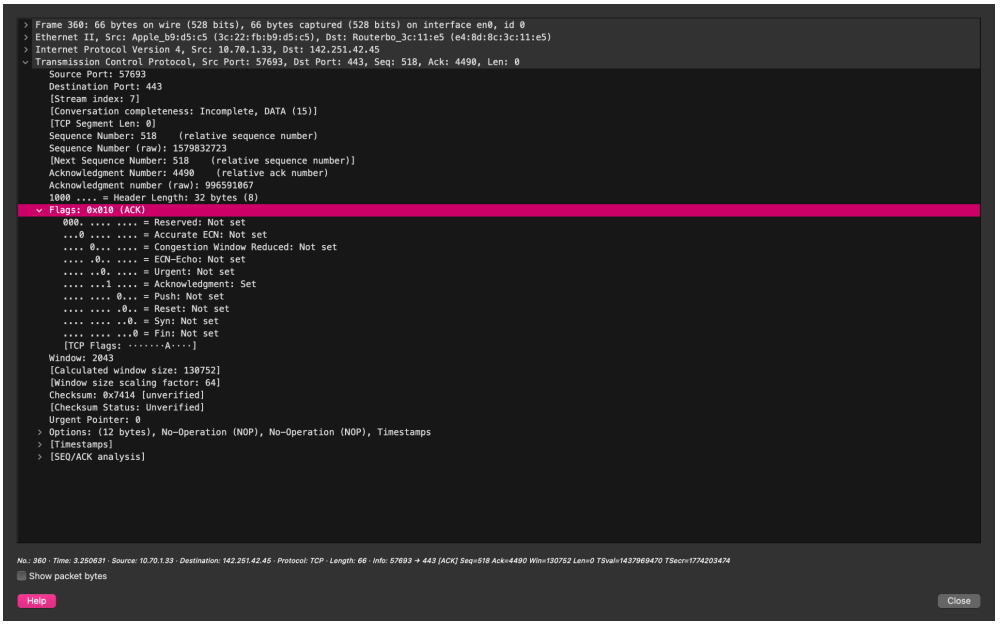
SYN: First, the client sends a packet with a sequence number **and only the SYN flag bit** set in the header. This initial packet allows the client to set what the first sequence number should be for request packets originating from the client. This is the client's synchronization step.



SYN-ACK: Second, the server responds to the SYN packet with an SYN/ACK packet. **Here, the server sets both the SYN flag bit and the ACK flag bit**. **This packet confirms the sequence number sent by the client by acknowledging it**.



ACK: Finally, the client responds to the SYN/ACK packet with an ACK packet that acknowledges the server's sequence number request.



1. What are the socket addresses for each packet?

Answer:

SYN: Source- 10.70.1.33:57693

Destination- 111.230.85.120:443

SYN-ACK: Source- 111.230.85.120:443

Destination- 10.70.1.33:57693

ACK: Source- 10.70.1.33:57693

Destination- 142.251.42.45:443

2. What flags are set in each packet?
Answer:

- SYN: 0x002 (set)
- SYN-ACK: 0x012 (set)
- ACK: 0x010 (set)

3. What are the sequence number and acknowledgement number of each packet?
Answer:

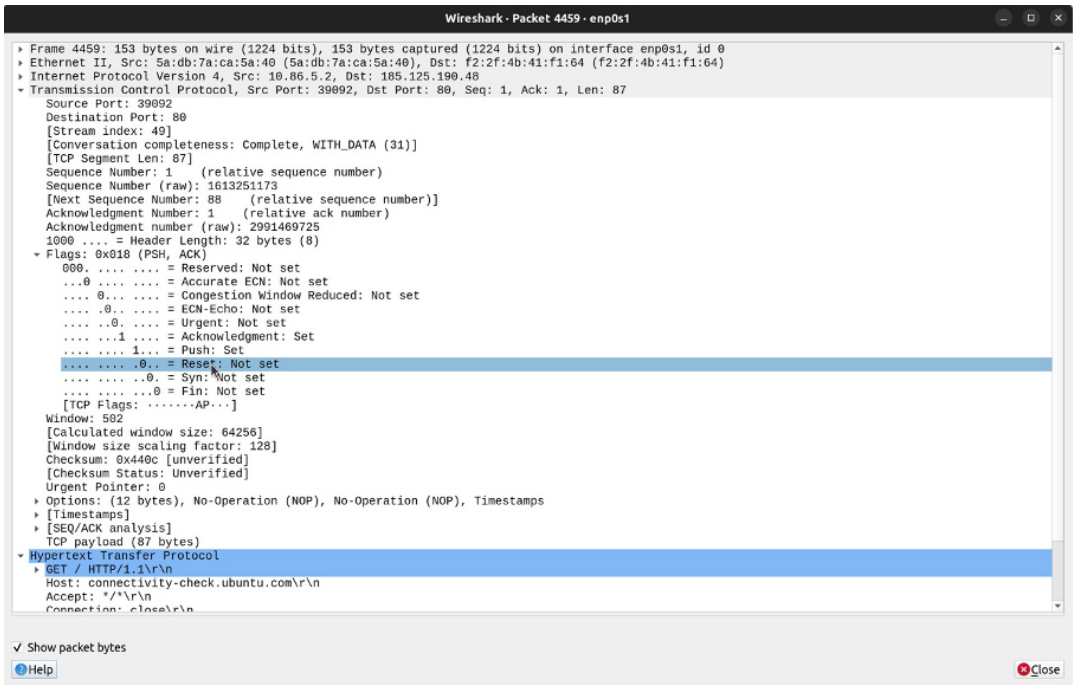
- SYN: Seq Number:3115492411, Ack Number: 0
- SYN-ACK: Seq Number: 3833306769, Ack Number: 1697807132
- ACK: Seq Number: 1579832723, Ack Number: 996591067

4. What are the window size of each packet?

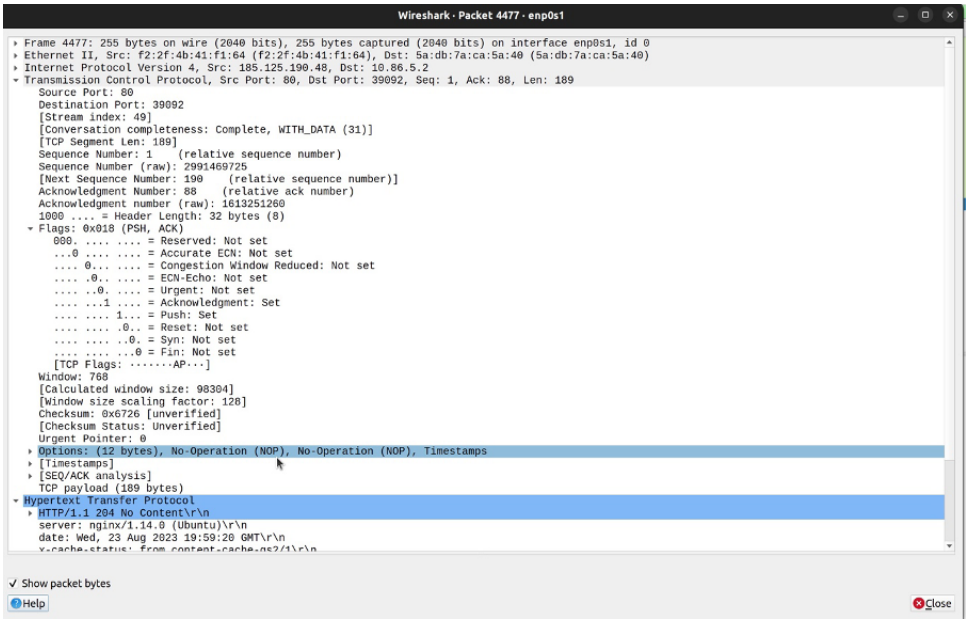
- SYN: 65535
- SYN-ACK: 655535
- ACK: 130752

Part II: Data-Transfer Phase
The data-transfer phase starts with an HTTP GET request message and ends with an HTTP OK message.

GET Request:



OK Response:



1. What TCP flags are set in the first data-transfer packet (HTTP GET message)?
Answer: ACK and PUSH flags are set

2. How many bytes are transmitted in this packet?
Answer: 72

3. How often does the receiver generate an acknowledgement? To which acknowledgement rule (defined on Page 200 in the textbook) does your answer correspond?
Answer: 0.003 seconds

4. How many bytes are transmitted in each packet? How are the sequence and acknowledgement numbers related to the number of bytes transmitted?
Answer: 72 bytes are transmitted in the GET request, and 174 bytes are received in the OK response. The sequence number increases by the number of bytes transmitted, and the acknowledgement 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?
Answer: The window size when sending a request to the server was set at 64256. The window size when receiving a response was 98304. The window size will increase till a certain capacity (limited by hardware) to prevent congestion.

6. Explain how the window size is used in flow control.
Answer: In a sliding window system in TCP, the size of the window is governed by 2 things:

- The size of the send buffer on the sending system

- The size and available space in the receive buffer on the receiving system

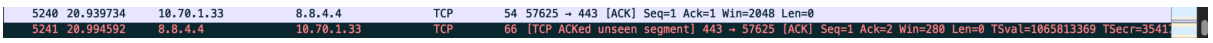
To avoid congestion, the sender cannot send more bytes than the space available in the receive buffer of the receiver. The sender must wait till the bytes in the receiving buffer have been acknowledged. This prevents congestion and helps in flow control.

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

Answer: The HTTP OK message is feedback about the request that was previously sent. An OK response means that the request has succeeded.

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.



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

Answer: 4 segments are exchanged in the connection termination phase. (FIN, ACK, FIN, ACK)

2. Which endpoint started the connection termination phase?

Answer: The client started the connection termination phase.

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

Answer: First, the client sends a segment with FIN and ACK flags set. Then the server responds with a segment with FIN and ACK flags set. Then the client sends a segment with ACK flag set, and finally, the server sends a segment with FIN and ACK flags set.