CS 558: Computer Systems Lab

Assignment –2 : Network Protocol Analysis Using Wireshark

Abhinit SIngh	234101058
Akshat Dubey	234101059
Param Bharatbhai Patel	234101062
Vinayak Bhausaheb Ichake	234101066

1. List out all the protocols used by the application at different layers (only those which you can figure out from traces). Study and briefly describe their packet formats.

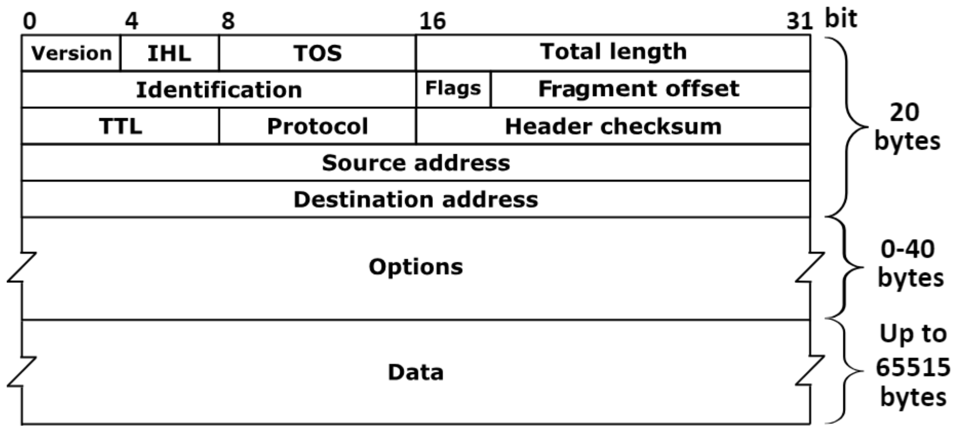
Answer :-

List of protocols:

- **IPv4**, an integral component of the network layer, facilitates the assignment of IP addresses crucial for internet communication.
- **TCP**, a cornerstone of the transport layer, guarantees reliable, connection-oriented communication, ensuring data integrity across networks.
- **UDP**, operating at the transport layer, provides a lightweight, connectionless mode of communication, ideal for scenarios prioritizing speed over reliability.
- **QUIC**, a transport layer protocol layered atop UDP, is purpose-built for achieving low-latency, secure communication, catering to modern internet demands.
- **TLSv1.2 and TLSv1.3**, a transport layer security protocol, though an older iteration, remains prevalent in facilitating secure communication over networks, ensuring data confidentiality and integrity.
- **IPv4**, an integral component of the network layer, facilitates the assignment of IP addresses crucial for internet communication.
- **Ethernet** serves as a foundational data link layer protocol tailored for wired local area networks (LANs).
- **DNS**, an application layer protocol, plays a pivotal role in translating user-friendly domain names into machine-readable IP addresses, enabling seamless internet navigation.

Packet Formats:

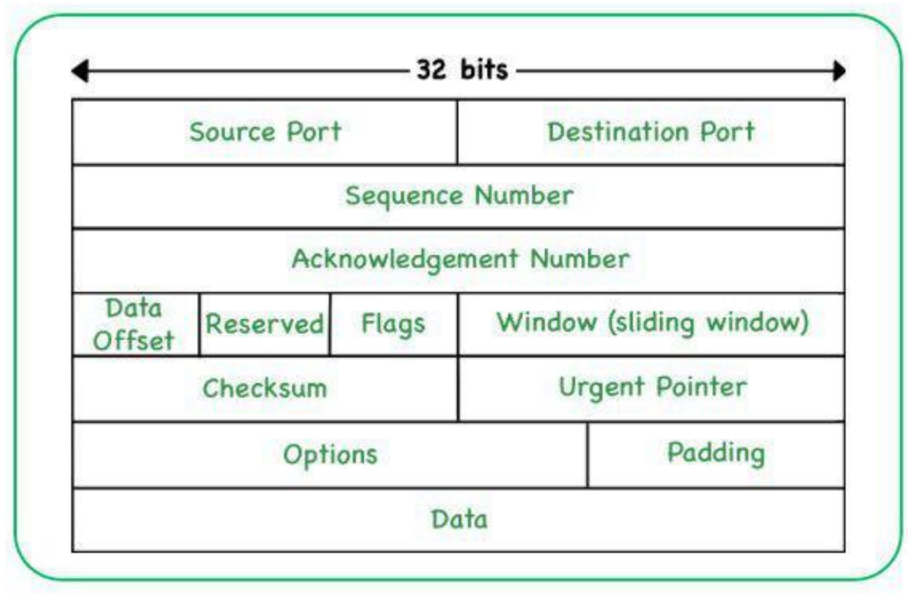
- **Ethernet:**
  - Frame Format: Ethernet frames consist of a destination MAC address, source MAC address, EtherType/Length field, payload (data), and a frame check sequence (FCS) for error detection.
- **IPv4:**



  - The IPv4 header serves as the essential framework for transmitting data across networks, encapsulating crucial information for routing and delivery.
  - ❖ Version: Identifies the IP version being used, typically IPv4.
  - ❖ Header Length: Specifies the size of the IPv4 header in 32-bit words.
  - ❖ Type of Service: Prioritizes packets based on quality of service requirements.
  - ❖ Total Length: Indicates the total length of the IPv4 packet, including header and data.
  - ❖ Identification, Flags, Fragment Offset: Enable packet fragmentation and reassembly for transmission across networks with varying maximum packet sizes.
  - ❖ Time to Live (TTL): Limits the lifespan of the packet, preventing it from circulating indefinitely.
  - ❖ Protocol: Specifies the protocol used in the data portion of the packet, such as TCP, UDP, or ICMP.
  - ❖ Header Checksum: Facilitates error detection in the IPv4 header.
  - ❖ Source and Destination IP Addresses: Identify the sender and intended recipient of the packet, enabling routing across interconnected networks.

- ❖ Together, these fields enable the reliable and efficient transmission of data packets across diverse networks comprising the internet and other IP-based infrastructures.

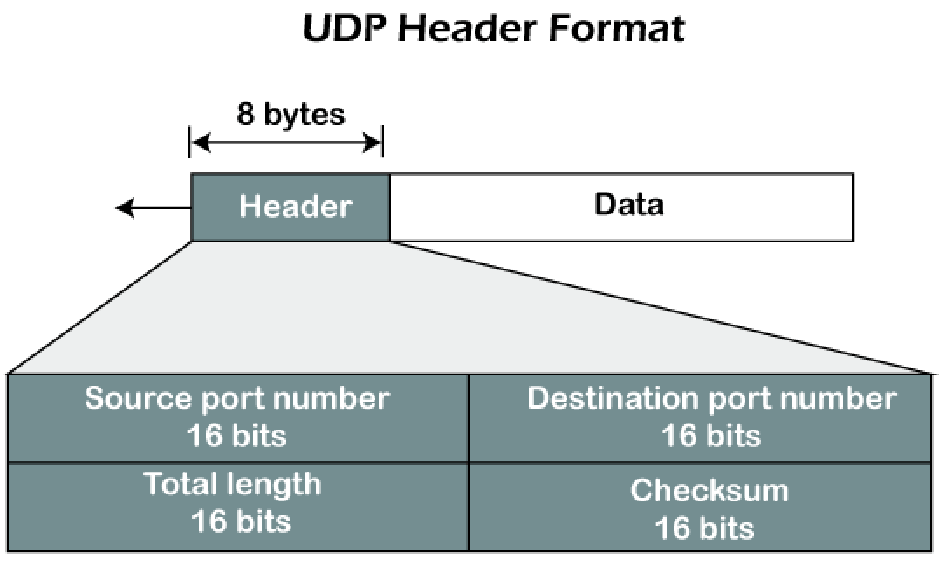
- **TCP**:



  - ○ The TCP (Transmission Control Protocol) header serves as the envelope for TCP segments, encapsulating crucial information for reliable and ordered data transmission. It includes fields vital for establishing and managing connections between communicating hosts.
  - ❖ Source Port and Destination Port: Identify the sending and receiving applications.
  - ❖ Sequence Number and Acknowledgment Number: Ensure ordered and reliable data delivery, tracking the sequence of transmitted segments and acknowledging received ones.
  - ❖ Header Length: Specifies the size of the TCP header in 32-bit words.
  - ❖ Flags (e.g., SYN, ACK): Control flags that manage the TCP connection's state and behavior, including initiation, acknowledgment, and termination.
  - ❖ Window Size: Determines the amount of data a sender can transmit before receiving an acknowledgment, facilitating flow control.
  - ❖ Checksum: Enables error detection in the TCP header and data.

- ❖ Urgent Pointer: Indicates the end of urgent data in the TCP segment, if present.
- ❖ These fields collectively facilitate reliable, connection-oriented communication over IP networks, forming the backbone of many internet-based applications and services.

- **UDP**:

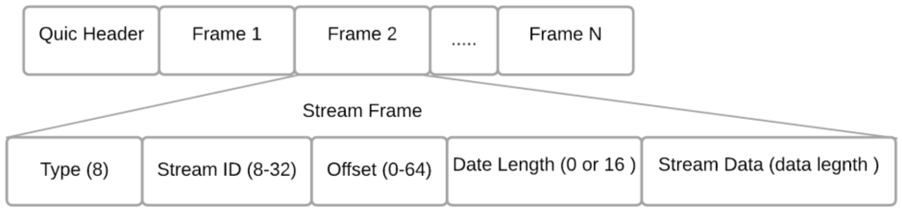


  - ○ The User Datagram Protocol (UDP) header is a concise structure designed for lightweight, connectionless communication. Its simplicity contrasts with the more elaborate TCP header. In the UDP header:

- ❖ Source Port: Identifies the sending application on the host.
- ❖ Destination Port: Specifies the intended recipient application.
- ❖ Length: Indicates the length of the UDP header and the data payload.
- ❖ Checksum: Facilitates error detection in the UDP header and data payload.

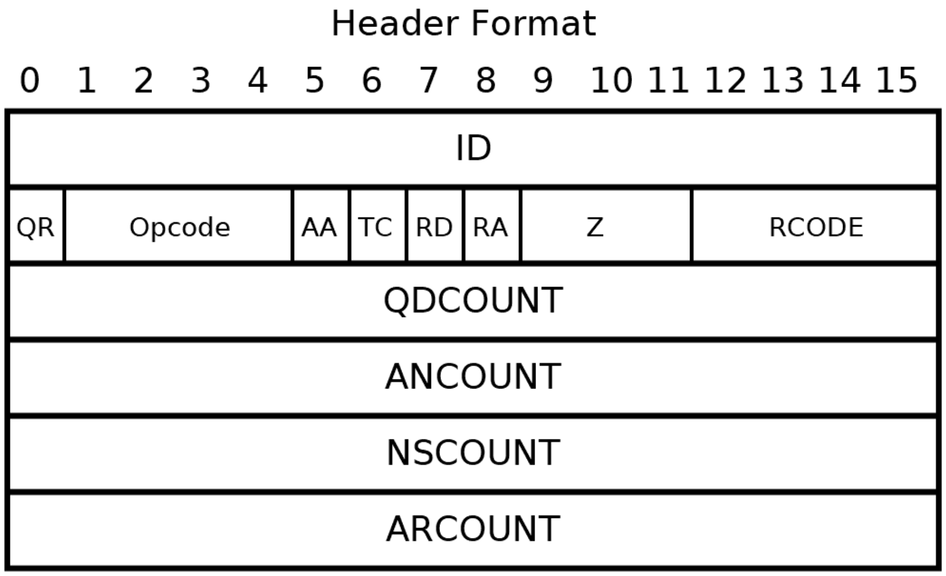
  - ○ Unlike TCP, UDP does not include features like sequence numbers, acknowledgments, or flow control mechanisms. This minimalistic design makes UDP well-suited for scenarios where speed and reduced overhead are prioritized over reliability, such as real-time streaming, DNS queries, and simple request-response transactions. However, the lack of built-in error recovery means applications using UDP must implement their own mechanisms for ensuring data integrity and reliability if needed.

- **QUIC:**



  - The QUIC packet format consists of several segments:
  - ❖ Long Header: This segment contains essential metadata for packet routing and processing, including the QUIC version, connection ID, packet number, and other control information. It's primarily used for the initial handshake and key exchanges.
  - ❖ Short Header: Unlike the long header, the short header is more concise and is used for subsequent packets within an established connection. It contains minimal routing information, usually just the packet number, as the connection parameters are already known.
  - ❖ Payload: The payload segment carries the actual application data or higher-layer protocol information, such as HTTP requests/responses, DNS queries, or other application-specific data. The payload length can vary depending on the application's needs and the QUIC packet size limitations.
  - ❖ Encryption: Each segment, including the header and payload, undergoes encryption to protect the confidentiality and integrity of the transmitted data. Encryption prevents unauthorized access, eavesdropping, and tampering, ensuring secure communication between endpoints.
  - These segments collectively form the QUIC packet structure, enabling efficient and secure transmission of data over the Internet.

- **DNS:**



  - DNS messages follow a structured format comprising a header and various fields:
  - **ID (Identification):** Unique identifier for the DNS message, facilitating proper identification and matching of responses to queries.
  - **QR (Query/Response):** Indicates whether the message is a query (0) or a response (1).
  - **Opcode:** Specifies the operation being performed, such as standard query (0), inverse query (1), server status request (2), etc.
  - AA (Authoritative Answer): Indicates if the responding server is authoritative for the queried domain.
  - TC (Truncated): Signals that the message was truncated during transmission due to exceeding the maximum size.
  - RD (Recursion Desired): Indicates if the client requests recursive resolution from the DNS server.
  - RA (Recursion Available): Informs the client whether the server supports recursive resolution.
  - Z (Reserved): Reserved for future use; currently set to zero.
  - RCODE (Response Code): Provides information about the status of the response, indicating success, errors, or other conditions.
  - Counts: Includes fields for the number of questions, answers, authority records, and additional records present in the message.

    - These components collectively define the structure of DNS messages, enabling efficient communication between DNS clients and servers for domain name resolution and other DNS-related services.

- **TLSv1.3 and TLSv1.2:**

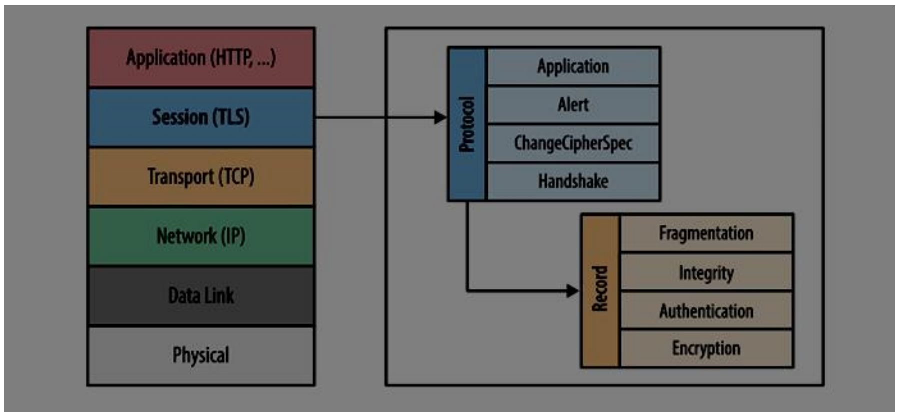


Figure 4-1. Transport Layer Security (TLS)

In TLSv1.3 and TLSv1.2, packet structure comprises two primary layers: the Handshake layer and the Record layer.

❖ In the Handshake Layer, the Header contains critical information like the message type (e.g., ClientHello, ServerHello), length, and a message sequence number. The Body carries message-specific content such as key exchange parameters, certificates, and cryptographic details. Notably, TLSv1.3 introduces advanced key exchange mechanisms like Diffie-Hellman, absent in TLSv1.2.

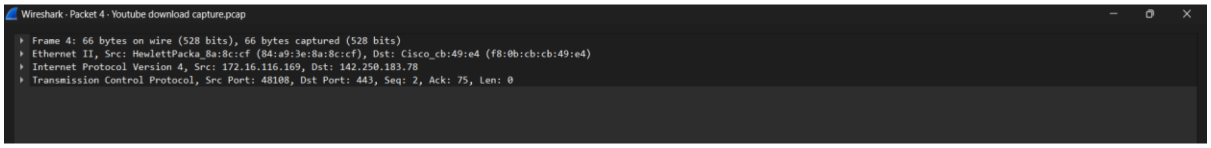
❖ Moving to the Record Layer, its Header includes the content type (e.g., Handshake, Application Data), version, length, and a record sequence number. The Payload contains encrypted data, which might also be compressed, depending on the cipher suite negotiated during the handshake process.

These distinct layers and their components form the framework for secure communication in TLS, enabling the exchange of cryptographic parameters and encrypted data between communicating entities.

2. Highlight and explain the observed values for various fields of the protocols. Example: Source or destination IP address and port number, Ethernet address,protocol number, etc.

Answer :-

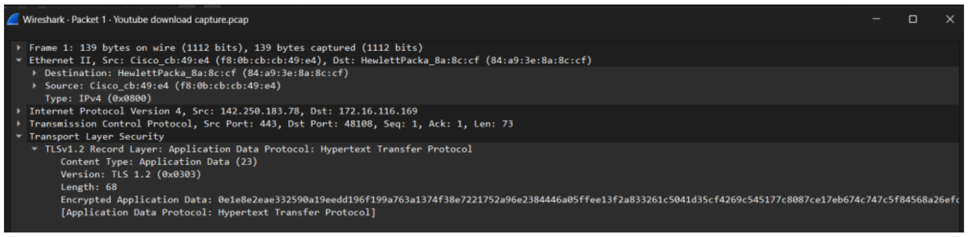
Transmission control Protocol(TCP)



```
Wireshark · Packet 4 · Youtube download capture.pcap
▸ Frame 4: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
▸ Ethernet II, Src: HewlettPacka_8a:8c:cf (84:a9:3e:8a:8c:cf), Dst: Cisco_cb:49:e4 (f8:0b:cb:cb:49:e4)
▸ Internet Protocol Version 4, Src: 172.16.116.169, Dst: 142.250.183.78
▸ Transmission Control Protocol, Src Port: 48108, Dst Port: 443, Seq: 2, Ack: 75, Len: 0
```

Source IP-> 172.16.11.169
Destination IP-> 142.250.183.78
Source Port-> 48108
Destination Port-> 443

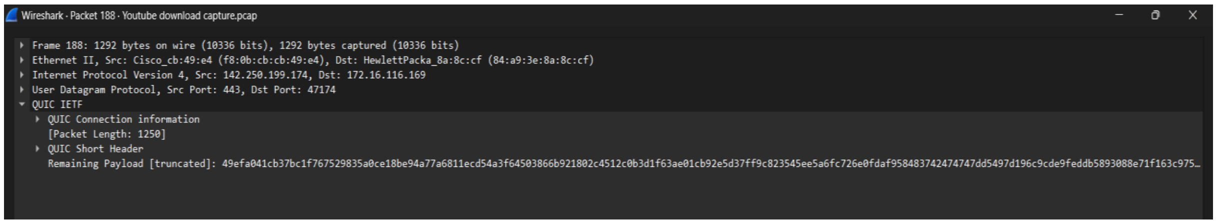
Transport Layer Security, or TLS



```
Wireshark · Packet 1 · Youtube download capture.pcap
▸ Frame 1: 139 bytes on wire (1112 bits), 139 bytes captured (1112 bits)
▾ Ethernet II, Src: Cisco_cb:49:e4 (f8:0b:cb:cb:49:e4), Dst: HewlettPacka_8a:8c:cf (84:a9:3e:8a:8c:cf)
  ▸ Destination: HewlettPacka_8a:8c:cf (84:a9:3e:8a:8c:cf)
  ▸ Source: Cisco_cb:49:e4 (f8:0b:cb:cb:49:e4)
    Type: IPv4 (0x0800)
▸ Internet Protocol Version 4, Src: 142.250.183.78, Dst: 172.16.116.169
▸ Transmission Control Protocol, Src Port: 443, Dst Port: 48108, Seq: 1, Ack: 1, Len: 73
▾ Transport Layer Security
  ▾ TLSv1.2 Record Layer: Application Data Protocol: Hypertext Transfer Protocol
      Content Type: Application Data (23)
      Version: TLS 1.2 (0x0303)
      Length: 68
      Encrypted Application Data: 0e1e8e2eae332590a19eedd196f199a763a1374f38e7221752a96e2384446a05ffee13f2a833261c5041d35cf4269c545177c8087ce17eb674c747c5f84568a26efc
      [Application Data Protocol: Hypertext Transfer Protocol]
```

Source IP->142.250.183.78
Destination IP->172.16.116.169
Source Port -> 443
Destination Port -> 48108
Ethernet Address Src-> f8:0b:cb:cb:49:e4
Ethernet Address Dest.-> 84:a9:3e:8a:8c:cf

Quick UDP Internet Connection(QUIC) :-



```
Wireshark · Packet 188 · Youtube download capture.pcap
▸ Frame 188: 1292 bytes on wire (10336 bits), 1292 bytes captured (10336 bits)
▸ Ethernet II, Src: Cisco_cb:49:e4 (f8:0b:cb:cb:49:e4), Dst: HewlettPacka_8a:8c:cf (84:a9:3e:8a:8c:cf)
▸ Internet Protocol Version 4, Src: 142.250.199.174, Dst: 172.16.116.169
▸ User Datagram Protocol, Src Port: 443, Dst Port: 47174
▾ QUIC IETF
  ▸ QUIC Connection information
    [Packet Length: 1250]
  ▸ QUIC Short Header
    Remaining Payload [truncated]: 49efa041cb37bc1f767529835a0ce18be94a77a6811ecd54a3f64503866b921802c4512c0b3d1f63ae01cb92e5d37ff9c823545ee5a6fc726e0fdaf958483742474747dd5497d196c9cde9feddb5893088e71f163c975…
```

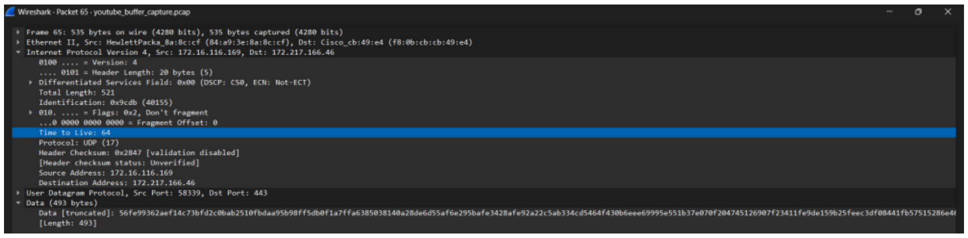
Source IP->142.250.199.174

Destination IP->172.16.116.169

Source Port -> 443

Destination Port -> 47174

User Datagram Protocol (UDP) :-



```
Wireshark · Packet 65 · youtube_buffer_capture.pcap
▸ Frame 65: 535 bytes on wire (4280 bits), 535 bytes captured (4280 bits)
▸ Ethernet II, Src: HewlettPacka_8a:8c:cf (84:a9:3e:8a:8c:cf), Dst: Cisco_cb:49:e4 (f8:0b:cb:cb:49:e4)
▾ Internet Protocol Version 4, Src: 172.16.116.169, Dst: 172.217.166.46
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 521
    Identification: 0x9cdb (40155)
  ▸ 010. .... = Flags: 0x2, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: UDP (17)
    Header Checksum: 0x2847 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 172.16.116.169
    Destination Address: 172.217.166.46
▸ User Datagram Protocol, Src Port: 58339, Dst Port: 443
▾ Data (493 bytes)
    Data [truncated]: 56fe99362aef14c73bfd2c0bab2510fbdaa95b98ff5db0f1a7ffa6385038140a28de6d55af6e295bafe3428afe92a22c5ab334cd5464f430b6eee69995e551b37e070f204745126907f23411fe9de159b25feec3df08441fb57515286e4
    [Length: 493]
```

Source IP->172.16.116.169

Destination IP->172.217.166.46

Source Port -> 58339

Destination Port -> 443

Source and Destination IP Addresses:

IP Address: Identifies the source and destination devices in the network.

Significance: Helps in routing the data packets between devices. Source IP is where the packet originates, and the destination IP is where it is intended.

Source and Destination Port Numbers:

Port Number: Specifies the application or service on the device.
Significance: Differentiates between multiple applications/services running on the same device. Allows proper delivery of data to the intended application.

Ethernet Addresses (MAC Addresses):

MAC Address: A unique hardware address assigned to a network interface controller (NIC).
Significance: Identifies the source and destination devices on a local network. Used for local data link layer communication.

Protocol Number:

Protocol Number: A numerical value assigned to a specific protocol (e.g., TCP, UDP, ICMP).
Significance: Indicates the type of transport layer protocol being used. For example, TCP (6) or UDP (17).

Time to Live (TTL):

TTL Field: Limits the lifespan of a packet, preventing it from circulating indefinitely.
Significance: Helps prevent packets from endlessly circulating in the network. Decreases with each hop, and if it reaches zero, the packet is discarded.

Flags (in TCP):

Flags (e.g., SYN, ACK, FIN): Control bits in TCP headers.
Significance: Manages the connection state. For example, SYN for initiating a connection, ACK for acknowledgment, FIN for terminating a connection.

Payload/Data:

Payload: The actual data being transmitted.
Significance: Carries the information intended for the application layer. The type of data can be inferred from the protocol and port numbers.

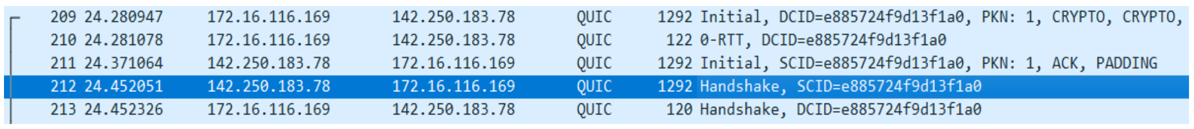
Sequence and Acknowledgment Numbers (in TCP):

Sequence Number: Specifies the position of the data in the stream.
Acknowledgment Number: Confirms the receipt of data.
Significance: Essential for managing data transmission and ensuring reliable delivery.

3. Explain the sequence of messages exchanged by the application for using the available functionalities in the application. For example: upload, download, play, pause, etc. Check whether there are any handshaking sequences in the application. Briefly explain the handshaking message sequence, if any.

Answer :-

1) Downloading :



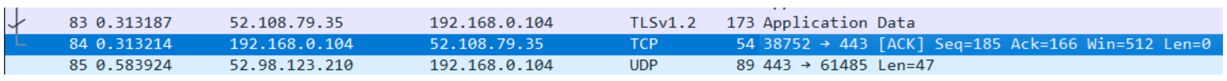
```
209 24.280947    172.16.116.169    142.250.183.78    QUIC    1292 Initial, DCID=e885724f9d13f1a0, PKN: 1, CRYPTO, CRYPTO,
210 24.281078    172.16.116.169    142.250.183.78    QUIC     122 0-RTT, DCID=e885724f9d13f1a0
211 24.371064    142.250.183.78    172.16.116.169    QUIC    1292 Initial, SCID=e885724f9d13f1a0, PKN: 1, ACK, PADDING
212 24.452051    142.250.183.78    172.16.116.169    QUIC    1292 Handshake, SCID=e885724f9d13f1a0
213 24.452326    172.16.116.169    142.250.183.78    QUIC     120 Handshake, DCID=e885724f9d13f1a0
```

While downloading a YouTube video, the communication between the client and the server occurs over HTTPS (port 443). However, the download process involves the transfer of the entire video file from the server to the client, with acknowledgment packets (ACK) ensuring data integrity during transmission.

It uses QUIC handshake in this proces. Both the client and server generate a Connection ID (CID) for the connection. The Connection ID helps in routing QUIC packets and allows for connection migration in case of network changes.

2) Play/Pause :

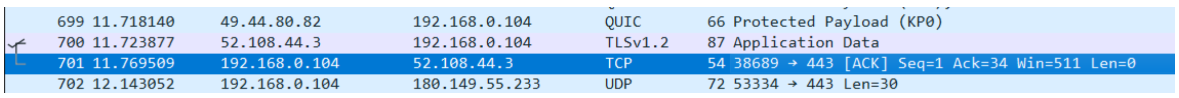
Whenever the video is paused then an ACK is sent from source port 38752 to destination port 443.



```
83 0.313187    52.108.79.35     192.168.0.104    TLSv1.2    173 Application Data
84 0.313214    192.168.0.104    52.108.79.35     TCP         54 38752 → 443 [ACK] Seq=185 Ack=166 Win=512 Len=0
85 0.583924    52.98.123.210    192.168.0.104    UDP         89 443 → 61485 Len=47
```

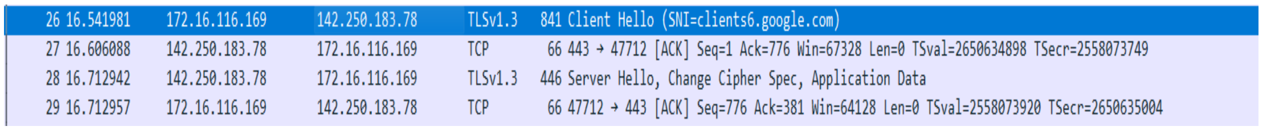
For the other scenario, when the video is resumed (paying the video after pausing), then also, we are getting ACK. This time, it is sent from source port 38689 to destination port 443.

Due to the dynamic allocation of ports by the client's operating system during TCP connection establishment, we are getting different client port. This is why interactions like playing after pausing may use different source ports.



```
699 11.718140   49.44.80.82     192.168.0.104   QUIC     66 Protected Payload (KP0)
700 11.723877   52.108.44.3     192.168.0.104   TLSv1.2  87 Application Data
701 11.769509   192.168.0.104   52.108.44.3     TCP      54 38689 → 443 [ACK] Seq=1 Ack=34 Win=511 Len=0
702 12.143052   192.168.0.104   180.149.55.233  UDP      72 53334 → 443 Len=30
```

3) Streaming:



```
26 16.541981  172.16.116.169  142.250.183.78  TLSv1.3  841 Client Hello (SNI=clients6.google.com)
27 16.606088  142.250.183.78  172.16.116.169  TCP       66 443 → 47712 [ACK] Seq=1 Ack=776 Win=67328 Len=0 TSval=2650634898 TSecr=2558073749
28 16.712942  142.250.183.78  172.16.116.169  TLSv1.3  446 Server Hello, Change Cipher Spec, Application Data
29 16.712957  172.16.116.169  142.250.183.78  TCP       66 47712 → 443 [ACK] Seq=776 Ack=381 Win=64128 Len=0 TSval=2558073920 TSecr=2650635004
```

Client Hello (SNI=clients6.google.com):

This packet represents the client's initial message to the server, indicating its intention to establish a TLS connection. It includes the Server Name Indication (SNI), which tells the server which domain the client is attempting to connect to.

TCP ACK (Acknowledgment):

This is an acknowledgment from the server to the client, confirming the receipt of the client's request (Client Hello). The ACK packet acknowledges the receipt of the data sent by the client and also includes TCP timestamp values.

Server Hello, Change Cipher Spec, Application Data:

The server responds to the client's request with its own message. It includes the Server Hello, which contains information necessary to establish the secure connection, such as the selected cipher suite and other parameters. The "Change Cipher Spec" message signals to the client that subsequent communication will be encrypted using the negotiated parameters. "Application Data" implies that the server may also be sending application-specific data.

TCP ACK:

This packet represents the client's acknowledgment of the server's response. It confirms the receipt of the Server Hello and acknowledges the server's message. Similar to Packet 27, it also includes TCP timestamp values.

TCP ACK:
This is another acknowledgment from the client to the server. It acknowledges the server's response (Packet 28) and confirms the establishment of the TCP connection.

4. Explain how the particular protocol(s) used by the application is relevant for functioning of the application.

Answer :-

The download of YouTube videos and the buffering of video content on YouTube involve several protocols that play distinct roles in ensuring the functionality of the application. Below are the key protocols and their relevance to the functioning of YouTube in terms of downloading and buffering videos:

- **HTTP/HTTPS (Hypertext Transfer Protocol/Secure):**

  Relevance: YouTube uses HTTP or HTTPS for video streaming. When playing a video, the browser or YouTube app sends a request to YouTube servers via HTTP/HTTPS to retrieve video content. HTTPS is crucial for securing communication, ensuring privacy, and maintaining data integrity.

- **TCP (Transmission Control Protocol):**

  Relevance: TCP is used for reliable, connection-oriented communication. During video streaming, TCP ensures data delivery in the correct order and without loss. It's essential for effective video buffering, preventing interruptions in playback caused by lost or out-of-order packets.

- **QUIC (Quick UDP Internet Connections):**

  Relevance: YouTube employs QUIC, a transport layer protocol developed by Google, to enhance the speed and reliability of data transfer. Operating over UDP, QUIC includes features like multiplexing and encryption, contributing to efficient video streaming.

- **DNS (Domain Name System):**

  Relevance: Before fetching video content, the application resolves the domain names of YouTube's servers to IP addresses using DNS. DNS plays a crucial role in translating human-readable domain names (e.g., www.youtube.com) into IP addresses for establishing connections.

- **CDN (Content Delivery Network):**

  Relevance: Content Delivery Networks, often supported by protocols like HTTP, ensure that video content is delivered from servers geographically closer to the user. This minimizes latency and improves the overall streaming experience by reducing data travel time.

- **WebSocket (optional for real-time features):**

  Relevance: While not core to video buffering, YouTube may use WebSocket for real-time features like live chat during video streaming. WebSocket provides full-duplex communication channels, enabling real-time interaction alongside video playback.

- **TLS (Transport Layer Security):**

  Relevance: Ensures secure communication over a computer network. Commonly used to encrypt data between applications and servers, providing a secure connection.

5. Calculate the following statistics from your traces while performing experiments at different times of the day: Throughput, RTT, Packet size, Number of packets lost, Number of UDP & TCP packets, Number of responses received with respect to one request sent. Report the observed values in your answer, preferably using tables.

	Test Case 1	**Test Case 2**	**Test Case 3**	**Test Case 4**
Throughput	5.26 MBPS	4.87 MBPS	5.21 MBPS	4.00 MBPS
RTT(in ms)	40.78	63.89	42.97	80.22
Packet size	1147.24	1139.68	973.86	
Number of packets lost	0	0	0	0
Number of UDP packets	1451	1185	937	1075
Number of TCP packets	107	123	169	122
Number of responses received w.r.t request sent	1.05	1.11	1.21	1.15

6. Check whether the whole content is being sent from the same location/source. List out the IP addresses of content providers if multiple sources exist, and explain the reason behind this.

Answer :-

The whole content is being sent from different locations and sources. Following is the list of IP addresses of content providers .

Different IP addresses during Youtube Video download
142.250.199.174
172.16.116.169
142.250.183.78

Different IP addresses during Youtube Video buffering
142.251.42.14
172.16.116.169
142.250.183.78
172.217.166.46

Due to following main reasons multiple sources exist while downloading or during buffering of a youtube video:

Content Delivery Network (CDN):

YouTube utilizes a CDN to distribute its content across multiple servers located. CDNs store copies of popular videos on servers closer to the end-users,As a result, when you stream a video, your device may connect to a server that is geographically closer to you.
Load Balancing: Load balancing involves distributing incoming network traffic across multiple servers to ensure that no single server is overwhelmed. YouTube uses load balancing to efficiently manage the high volume of video requests. As a user, you may be directed to different servers based on server load, network conditions.

Parallel Downloading and Streaming: YouTube may break down a video into smaller chunks, and these chunks can be downloaded or streamed in parallel from different servers.

Redundancy and Reliability: Multiple sources provide redundancy and improve the reliability of content delivery. If one server experiences issues or becomes unavailable.

Efficient Bandwidth Utilization: Distributing the load across multiple servers allows for more efficient use of available network bandwidth. This approach helps prevent network congestion and ensures a smoother streaming experience for users.

Content Adaptation: YouTube may dynamically adapt the quality of the video stream based on your network conditions. Different sources may be used to deliver content at varying bitrates, ensuring optimal playback quality depending on your available bandwidth.

* * * * * * * *