CS 558: Computer Systems Lab

Assignment –2 : Network Protocol Analysis Using Wireshark

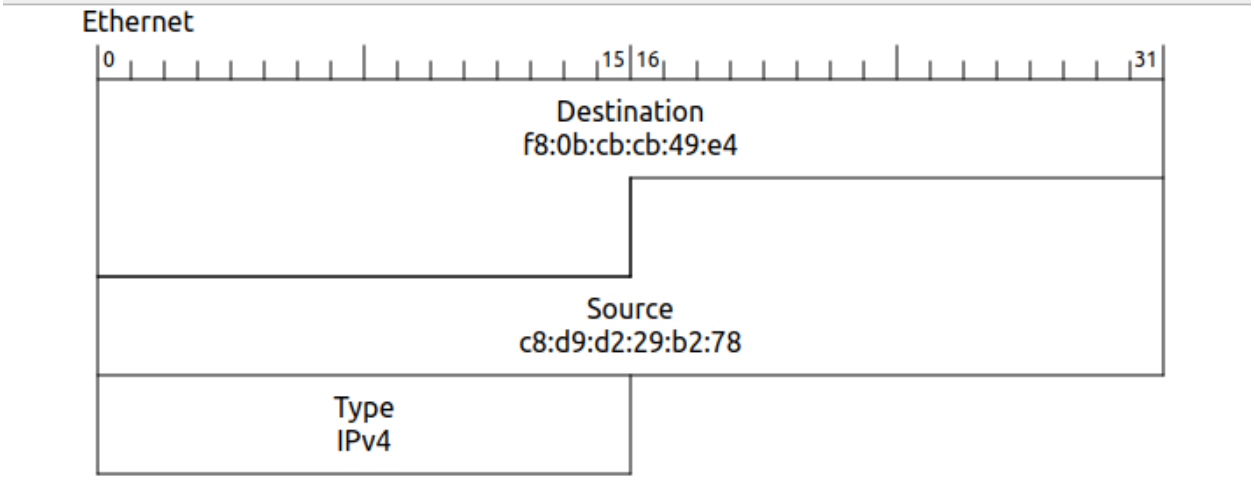
1. List out all the protocols used by the application at different layers (only those which you can figure out from traces). Study and briefly describe their packet formats.

List of protocols:

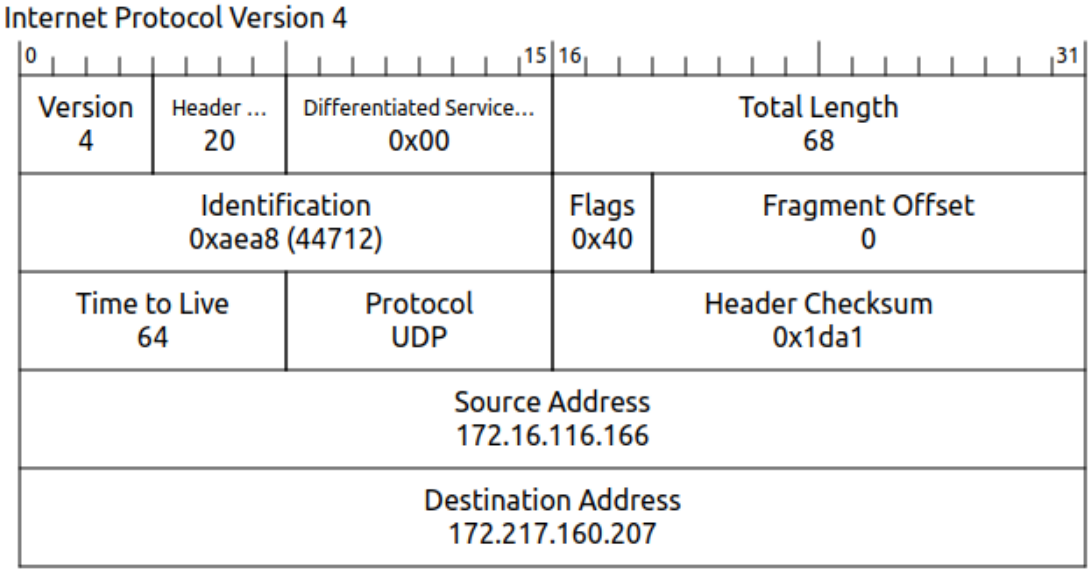
- Ethernet: Data link layer protocol for wired local area networks (LANs).
- IPv4: Network layer protocol providing IP addressing for internet communication.
- TCP: Transport layer protocol offering reliable, connection-oriented communication.
- UDP: Transport layer protocol provides connectionless, lightweight communication.
- QUIC: Transport layer protocol over UDP, designed for low-latency and secure communication.
- DNS: Application layer protocol translating domain names to IP addresses.
- TLSv1.3: Transport layer security protocol, the latest version providing secure communication.

Packet Formats:

- Ethernet:



- The Key fields include:
    - Destination MAC Address (6 bytes):
        - Specifies the Media Access Control (MAC) address of the intended recipient device.
    - Source MAC Address (6 bytes):
        - Indicates the MAC address of the sender's network interface.
    - EtherType or Length Field:
        - For Ethernet II frames, this field specifies the protocol type of the encapsulated payload (e.g., IPv4, IPv6, ARP).
        - For IEEE 802.3 frames, this field indicates the length of the data payload.
- IPv4:

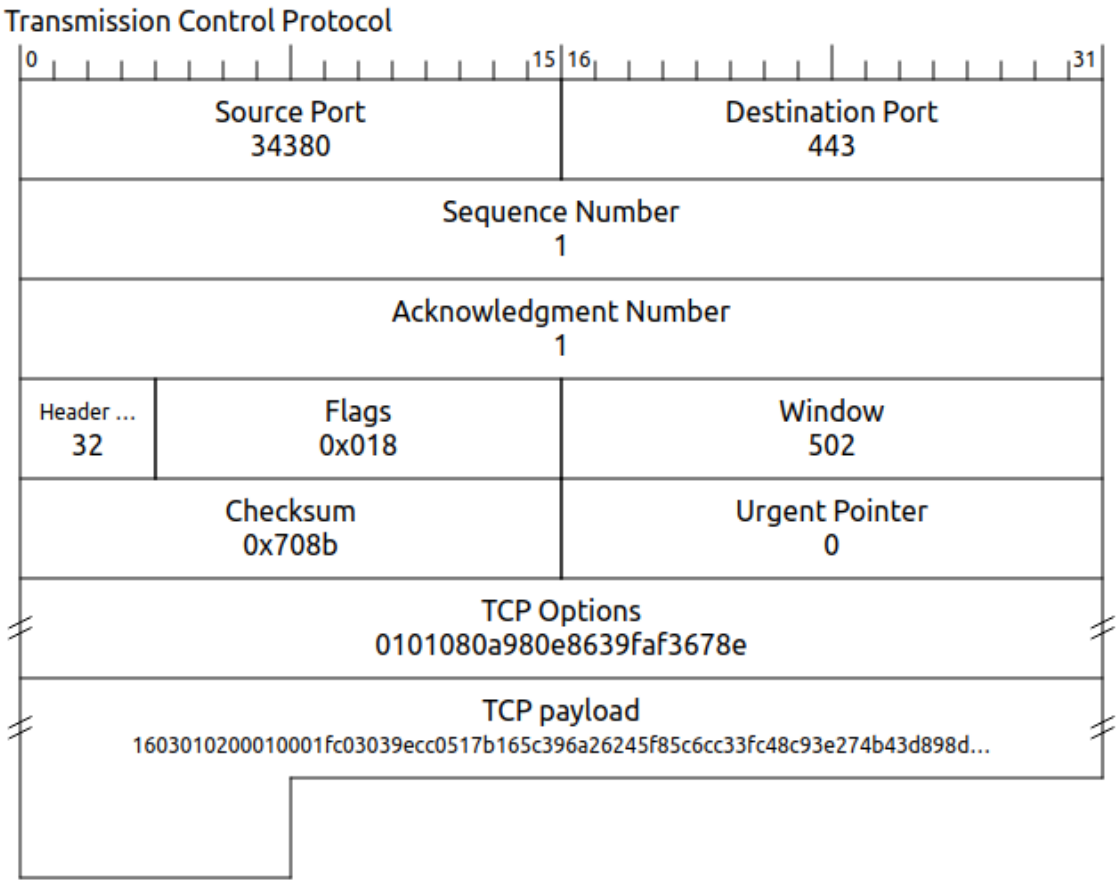


IP (Internet Protocol) facilitates data packet routing from source to destination in networks. The header includes key details:

- Version: Denotes the IP version, often "4" for IPv4.
- Header Length: Specifies the header size.
- Total Length: Indicates overall datagram size in bytes (header + payload).
- DF Bit: Directs routers not to fragment the datagram.
- MF Bit: Signals more fragments if set.
- TTL: Limits the maximum hops.
- Protocol Field: Identifies the destination's higher-layer protocol.

- Source/Destination Address: Holds sender and receiver addresses.
- Options: Used for various purposes like recording routes, source routing, or padding.

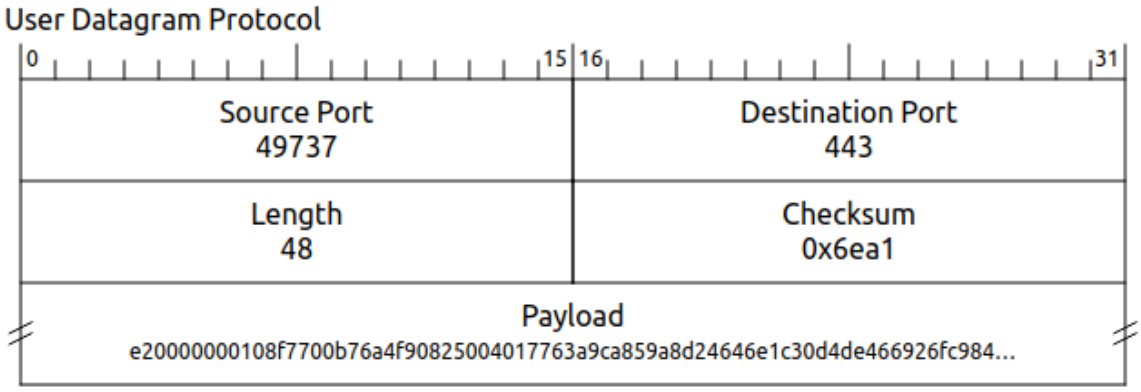
- TCP:



The header includes key details:

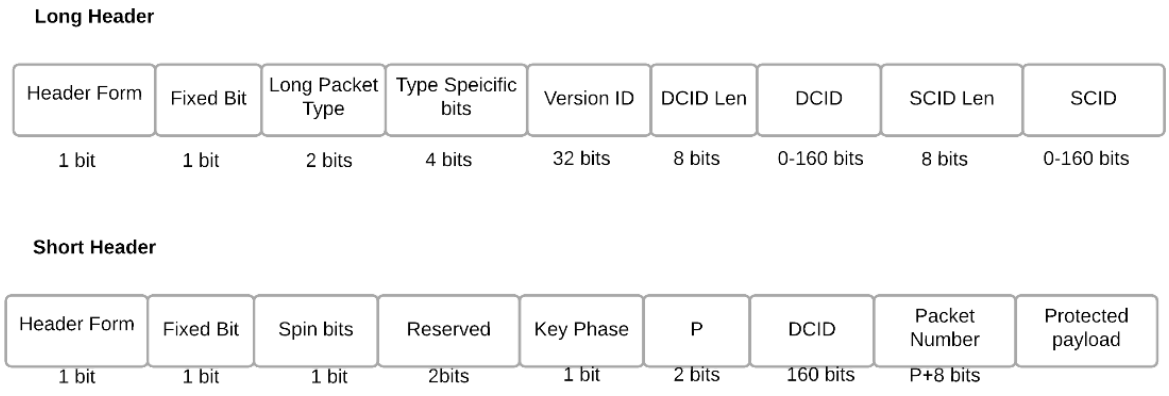
- Source Port and Destination Port: Identify sender and receiver applications.
- Sequence Number: Tracks the first data byte's sequence number.
- Acknowledgement Number: Specifies the expected next data byte.
- Header Length: Indicates TCP header length.
- Flags (e.g., ACK, PSH, SYN): Control packet behavior.
- Checksum: Ensures payload data integrity.
- Window Size: Shows sender's window for unacknowledged data.
- Urgent Pointer: Identifies urgent data in the current segment.
- Options: Used for various purposes like timestamps, window size extension, and parameter negotiation.

- UDP:



Key components in a UDP header:

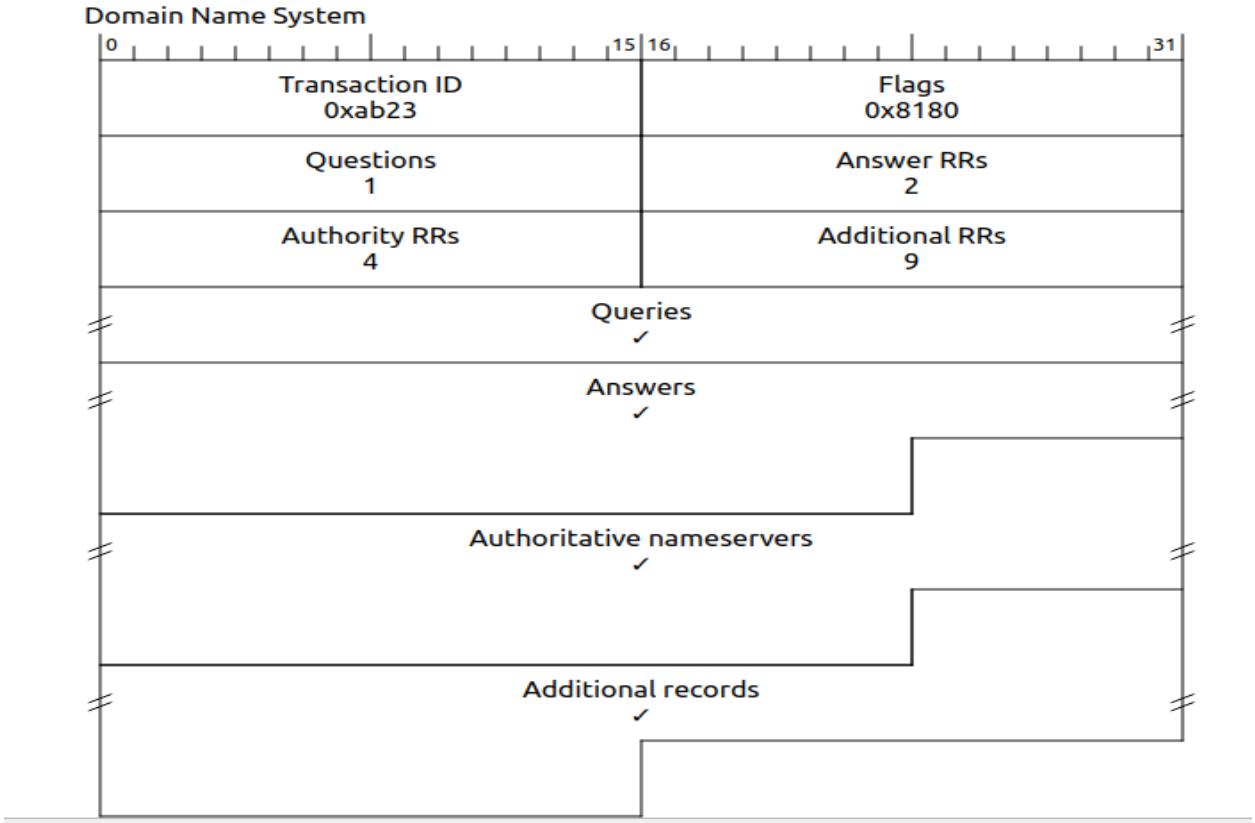
  - Source Port (16 bits): Identifies the sender's port number, indicating the application or service on the sender's device.
  - Destination Port (16 bits): Specifies the recipient's port number, indicating the application or service on the recipient's device.
  - Length (16 bits): Indicates the total length of the UDP header and the data in bytes. The minimum length is 8 bytes.
  - Checksum (16 bits): Used for error-checking purposes. The checksum is calculated over the UDP header, pseudo-header (including source and destination IP addresses), and the data payload. It helps ensure the integrity of the transmitted data.
- QUIC:



The header includes key details:

- Header Form (1 bit): It says the header is long (0) or short (1).
- Fixed Bit (1 bit): It indicates the presence of a long header by setting it to 1.
- Spin Bit (1 bit): This bit is reserved for possible future use and is not yet defined in the QUIC specification.
- Reserved Bits (2 bits): Reserved for future use and should be set to zero.
- Packet Number Length (2 bits): This field specifies the length of the packet number that follows.
- Version (32 bits): In the long header, this field represents the QUIC version being used. It is used for version negotiation during connection establishment.
- Destination Connection ID (0 to 160 bits): This field identifies the recipient's connection. It may not be present in short headers.
- Packet Number (Variable Length): The packet number is included to aid in packet sequencing and reordering
- Payload (Variable Length): The payload carries application data or control information, depending on the packet's purpose and type.

- DNS:

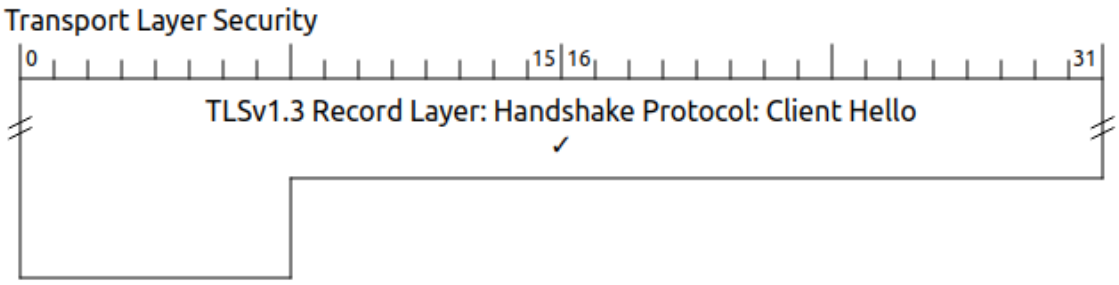


  - The DNS protocol works when your computer sends out a DNS query to a name server to resolve a domain.
  - The packet has a fixed 12-byte header followed by four variable-length fields.
  - The field id is set by the client and returned by the server. It lets the client match responses to requests .
  - The 16-bit *flags* field is divided into numerous pieces:

  - Key flags include:
    - QR (Query/Response): 1 bit indicating whether the message is a query (0) or a response (1).
    - Opcode: 4 bits specifying the type of query (standard query, inverse query, server status request, etc.).
    - AA (Authoritative Answer): 1 bit indicating if the responding DNS server is authoritative for the queried domain.

        - TC (Truncation): 1 bit signaling if the message was truncated due to its size.
        - RD (Recursion Desired): 1 bit indicating if the client requests recursive resolution from the server.
        - RA (Recursion Available): 1 bit indicating if the server supports recursion.
    - Question Count:
        - A 16-bit field specifying the number of questions in the message. Typically, a DNS message contains one question per query.
    - Answer, Authority, and Additional Resource Record Counts:
        - Each of these is a 16-bit field specifying the number of resource records in the corresponding section of the DNS message. These sections include answers to the query, authoritative information, and additional data, respectively.
    - Queries and Resource Records:
        - These sections contain the actual queries and responses, including information such as domain names, record types, time-to-live (TTL), and data.

- TLSv1.3 and TLSv1.2:

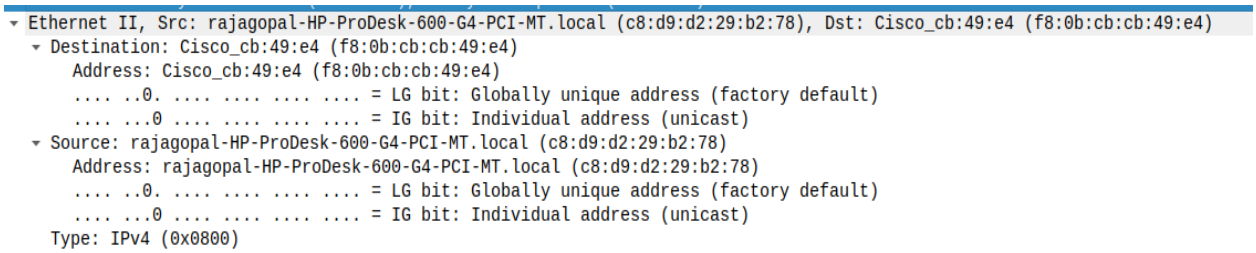


    - The packet structure in TLSv1.3 and TLSv1.2 includes two main layers: the Handshake layer and the Record layer.
    - Handshake Layer:
        - Header: Contains the message type (e.g., ClientHello, ServerHello), length, and a message sequence number.
        - Body: Specific content based on the message type, such as key exchange parameters, certificates, and cryptographic information.

    - Key Exchange: TLSv1.3 supports key exchange using mechanisms like the Diffie-Hellman key exchange and this is not presented in TLSv1.2.
- Record Layer:
    - Header: Contains the content type (e.g., Handshake, Application Data), version, length, and a record sequence number.
    - Payload: Encrypted data, potentially compressed, based on the negotiated cipher suite during the handshake.

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.

Ethernet:



Observations:

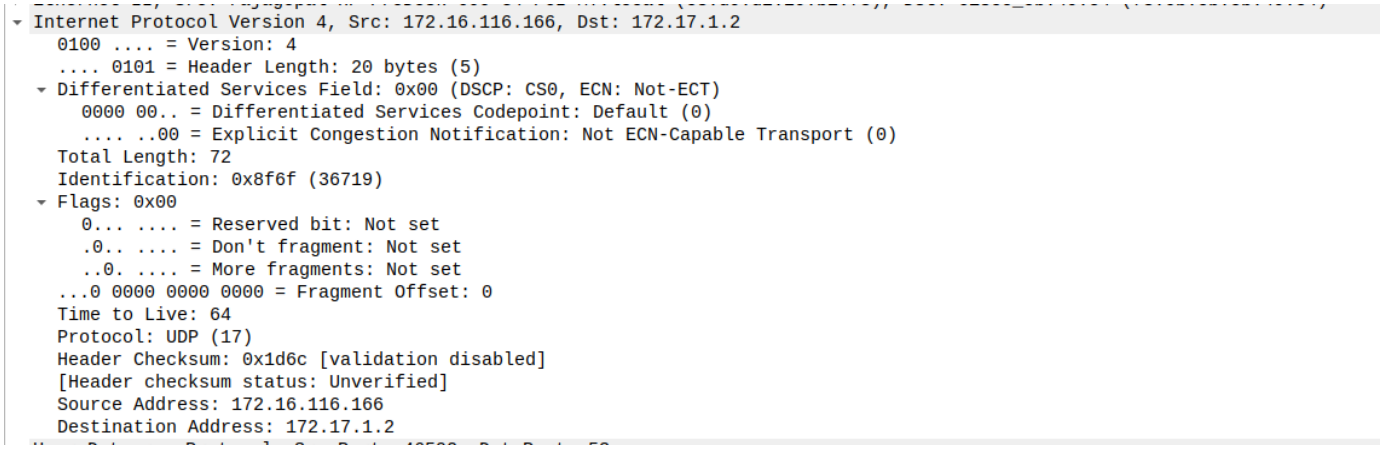
Destination Address -> f8:8b:cb:cb:49:e4

Source Address ->c8:d9:d2:29:b2:78

In MAC addresses, both the source and destination fields feature two significant bits: the LG (Locally/Global Assigned) bit and the IG (Individual/Group) bit. The LG bit distinguishes between addresses that are vendor-assigned (0) and administratively assigned (1).Meanwhile, the IG bit specifies whether the MAC address is intended for unicast (0) or multicast (1) communication. In the scenario described, both the source and destination MAC addresses have LG and IG bits set to 0, signifying that they are globally unique addresses and designated for unicast communication.

Type indicates the upper layer protocol to be used which is IPv4 in this case.

IPv4:



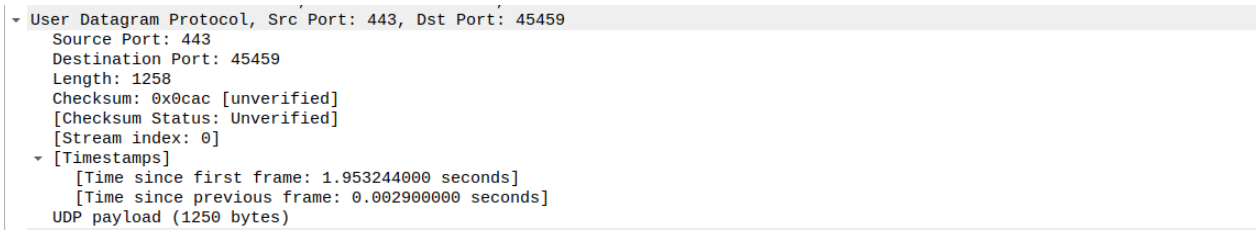
Observations:

source address(pc) is 172.16.116.166

Destination address is 172.17.1.2.

Header length is 20 bytes and total length of packet is 72 bytes and DF bit is 0 which means the packet should not be fragmented and MF bit is 0 which means the packet has no fragments with the same identification number. Time to live is 64 which means the packet can make at most 64 hops. UDP is used as the next level protocol**.**

UDP:



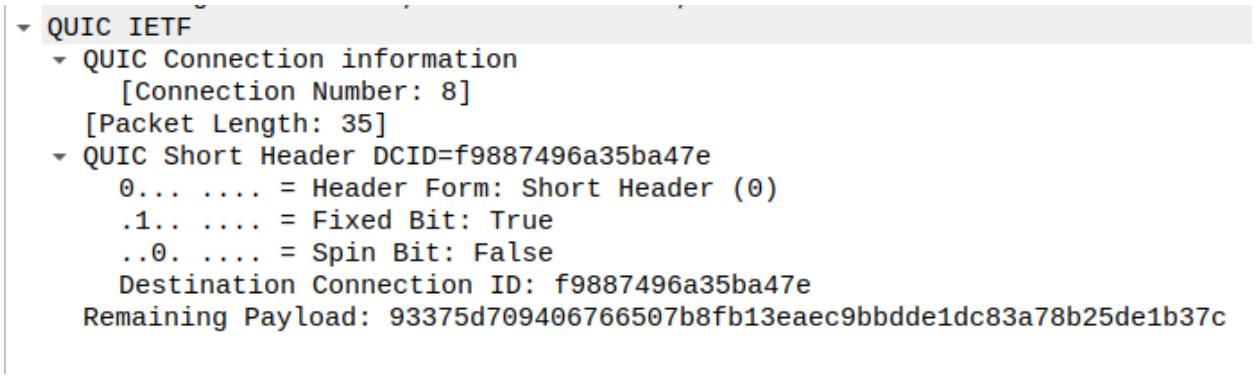
Observations:

Source Port : 443

Destination Port: 45459

Payload is 1250 bytes and Total length is 1258.

QUIC:



Observations:

Header form is 0 which indicates it is a short header. DCID is f9887496a35ba47e and Fixed bit is set to 1 indicating the presence of a long header and the spin bit set to 0.

TLS:

Observations:

Opaque Type is Application and the version used is TLS 1.2. Total length is 915 and session id length is 57.

It is used instead of http protocol.

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.

I) Launching Youtube:

Upon opening a browser and searching for studio.youtube.com, our computer dispatches DNS requests to the DNS server, receiving a response in return.

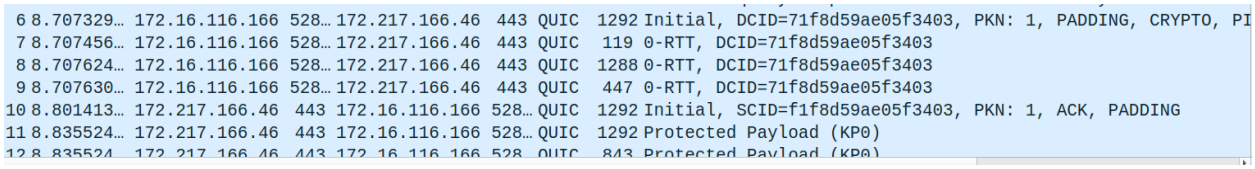
```
1 0.0… 172.16.116.166 172.17.1.2     DNS    89 Standard query 0xa7ca A studio.youtube.com OPT
2 0.0… 172.16.116.166 172.17.1.2     DNS    89 Standard query 0xca44 HTTPS studio.youtube.com OPT
3 0.0… 172.17.1.2     172.16.116.166 DNS   627 Standard query response 0xa7ca A studio.youtube.com CN
4 0.0… 172.17.1.2     172.16.116.166 DNS   386 Standard query response 0xca44 HTTPS studio.youtube.co
```

The below image shows the queries and answers received for those queries.

```
▾ Queries
  ▾ studio.youtube.com: type A, class IN
      Name: studio.youtube.com
      [Name Length: 18]
      [Label Count: 3]
      Type: A (Host Address) (1)
      Class: IN (0x0001)
▾ Answers
  ▸ studio.youtube.com: type CNAME, class IN, cname youtube-ui.l.google.com
  ▸ youtube-ui.l.google.com: type A, class IN, addr 172.217.166.46
  ▸ youtube-ui.l.google.com: type A, class IN, addr 142.250.183.46
  ▸ youtube-ui.l.google.com: type A, class IN, addr 142.250.192.142
  ▸ youtube-ui.l.google.com: type A, class IN, addr 142.251.42.78
  ▸ youtube-ui.l.google.com: type A, class IN, addr 172.217.27.206
  ▸ youtube-ui.l.google.com: type A, class IN, addr 142.250.183.78
  ▸ youtube-ui.l.google.com: type A, class IN, addr 172.217.166.174
  ▸ youtube-ui.l.google.com: type A, class IN, addr 142.250.192.110
  ▸ youtube-ui.l.google.com: type A, class IN, addr 142.251.42.14
```

II) For User Interface Display:

QUIC handshake takes place here between our pc (172.16.116.166) and youtube-ui server(172.217.166.46).



6	8.707329…	172.16.116.166	528…	172.217.166.46	443	QUIC	1292	Initial, DCID=71f8d59ae05f3403, PKN: 1, PADDING, CRYPTO, PI
7	8.707456…	172.16.116.166	528…	172.217.166.46	443	QUIC	119	0-RTT, DCID=71f8d59ae05f3403
8	8.707624…	172.16.116.166	528…	172.217.166.46	443	QUIC	1288	0-RTT, DCID=71f8d59ae05f3403
9	8.707630…	172.16.116.166	528…	172.217.166.46	443	QUIC	447	0-RTT, DCID=71f8d59ae05f3403
10	8.801413…	172.217.166.46	443	172.16.116.166	528…	QUIC	1292	Initial, SCID=f1f8d59ae05f3403, PKN: 1, ACK, PADDING
11	8.835524…	172.217.166.46	443	172.16.116.166	528…	QUIC	1292	Protected Payload (KP0)
12	8.835524…	172.217.166.46	443	172.16.116.166	528…	QUIC	843	Protected Payload (KP0)

QUIC Handshake:

The QUIC handshake is a cryptographic process that establishes a secure connection between a client and a server. It is designed to be faster and more efficient than the traditional TCP/TLS handshake, and it uses a single round trip time (RTT) in most cases.

Client to Server:

1. Initial packet: The client sends an Initial packet containing a ClientHello message. The ClientHello message includes information about the client's supported versions of QUIC, the cipher suites it supports, and a random number

Server to Client:

2. Initial packet: If the server accepts the client's hello, it sends an Initial packet containing a ServerHello message. The ServerHello message includes information about the server's supported versions of QUIC, the cipher suite it has chosen, its certificate chain, and a random number.
3. Handshake packet: The server sends a Handshake packet containing the rest of the TLS handshake messages, such as the CertificateVerify and Finished messages.

Client to Server:

4. Handshake packet: The client sends a Handshake packet containing its Finished message.

III) For Account Information:

In order to load corresponding user account information a DNS query for accounts.youtube.com has been sent to the DNS server. Based on the response using QUIC the corresponding user accounts details are displayed.

IV) For Upload:

Whenever we click on the upload button a DNS query for upload.youtube.com has been sent to the DNS server. In response we received an upload server IP (172.217.160.207).

```
47 7.… 172.16.116.166   55… 172.17.1.1        53   DNS     89   Standard query 0xab23 A upload.youtube.com OPT
48 7.… 172.16.116.166   59… 172.17.1.1        53   DNS     89   Standard query 0x52d4 HTTPS upload.youtube.com OPT
49 8.… 172.17.1.1        53 172.16.116.166   55051 DNS     392  Standard query response 0xab23 A upload.youtube.com CNAME yt-v
50 8.… 172.17.1.1        53 172.16.116.166   59402 DNS     178  Standard query response 0x52d4 HTTPS upload.youtube.com CNAME
```

```
▾ Queries
  ▾ upload.youtube.com: type A, class IN
      Name: upload.youtube.com
      [Name Length: 18]
      [Label Count: 3]
      Type: A (Host Address) (1)
      Class: IN (0x0001)
▾ Answers
  ▸ upload.youtube.com: type CNAME, class IN, cname yt-video-upload.l.google.com
  ▸ yt-video-upload.l.google.com: type A, class IN, addr 172.217.160.207
```

After selecting the file for upload , in parallel upload and form filling related to video has been done. In order to share the form details a TCP/TLS connection is being established between our PC (172.16.116.166 :34380) and the upload server (172.217.160.207:) by performing TCP and TLS Handshake . In parallel for upload, a QUIC connection is being established between our PC(172.16.116.166 : 49737) and the upload server (172.217.160.207: 443) by performing QUIC handshake.

During the form filling, two handshaking procedures take place, first 3-way TCP handshaking and then TLS Handshaking.

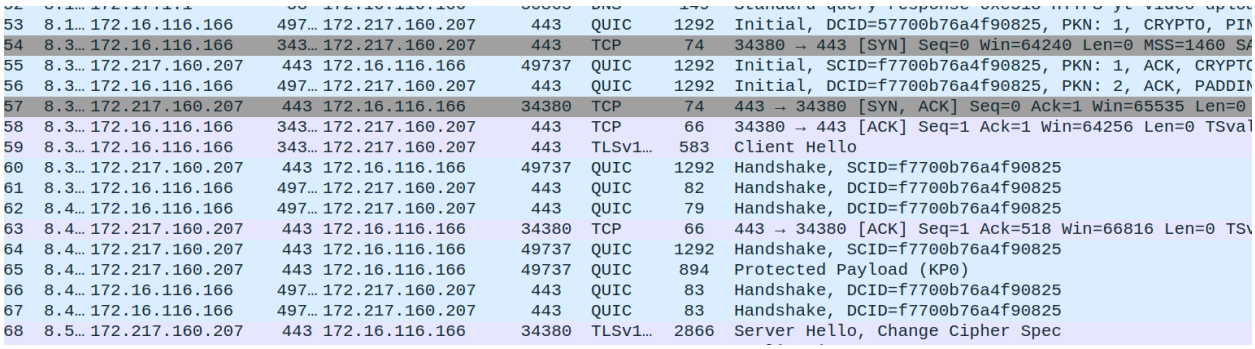
TCP 3-way Handshake: This process is used to make a connection between the client (PC) and the server (upload.youtube.com) in a TCP/IP network. It is a 3-step process between port 34380 of the client and port 443 of the server.

Step 1: SYN: The client sends a segment to the server with SYN (Synchronize Sequence Number) which is the initial sequence number it plans to use and informs the server that the client is likely to start communication.

Step 2: SYN, ACK : The server sends a response with SYN-ACK bit set. ACK (Acknowledgement) indicates that the server has acknowledged the client's sequence number and SYN signifies server's sequence number with which it is likely to start with.

Step 3: ACK : The client then sends a message with ACK bit set, and acknowledges the server's response. The client and server establish a reliable connection for actual data transfer.

TLS Handshake: TLS handshake is used to make the connection secure. First the TLS protocol sends a 'Client Hello' message to initiate a session with the server. The server responds with a 'Server Hello' message containing the server certificate which is used primarily for authentication, cipher suite requirements, and randomly generated data for creating session keys. The client responds with a client key and a secure connection is established between the client and the server.



53	8.1…	172.16.116.166	497…	172.217.160.207	443	QUIC	1292	Initial, DCID=57700b76a4f90825, PKN: 1, CRYPTO, PIN
54	8.3…	172.16.116.166	343…	172.217.160.207	443	TCP	74	34380 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SA
55	8.3…	172.217.160.207	443	172.16.116.166	49737	QUIC	1292	Initial, SCID=f7700b76a4f90825, PKN: 1, ACK, CRYPTO
56	8.3…	172.16.116.166	497…	172.217.160.207	443	QUIC	1292	Initial, DCID=f7700b76a4f90825, PKN: 2, ACK, PADDIN
57	8.3…	172.217.160.207	443	172.16.116.166	34380	TCP	74	443 → 34380 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0
58	8.3…	172.16.116.166	343…	172.217.160.207	443	TCP	66	34380 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval
59	8.3…	172.16.116.166	343…	172.217.160.207	443	TLSv1…	583	Client Hello
60	8.3…	172.217.160.207	443	172.16.116.166	49737	QUIC	1292	Handshake, SCID=f7700b76a4f90825
61	8.3…	172.16.116.166	497…	172.217.160.207	443	QUIC	82	Handshake, DCID=f7700b76a4f90825
62	8.4…	172.16.116.166	497…	172.217.160.207	443	QUIC	79	Handshake, DCID=f7700b76a4f90825
63	8.4…	172.217.160.207	443	172.16.116.166	34380	TCP	66	443 → 34380 [ACK] Seq=1 Ack=518 Win=66816 Len=0 TSv
64	8.4…	172.217.160.207	443	172.16.116.166	49737	QUIC	1292	Handshake, SCID=f7700b76a4f90825
65	8.4…	172.217.160.207	443	172.16.116.166	49737	QUIC	894	Protected Payload (KP0)
66	8.4…	172.16.116.166	497…	172.217.160.207	443	QUIC	83	Handshake, DCID=f7700b76a4f90825
67	8.4…	172.16.116.166	497…	172.217.160.207	443	QUIC	83	Handshake, DCID=f7700b76a4f90825
68	8.5…	172.217.160.207	443	172.16.116.166	34380	TLSv1…	2866	Server Hello, Change Cipher Spec

After the QUIC handshake, All the video data is being sent securely over the established QUIC connection.

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

Ethernet :

Ethernet is the most widely used data link layer protocol. It is preferred over other protocols because of its reliable data transfer, high speed and security. It involves proper error handling and flow control mechanisms for error handling along with CRC for error detection and preamble for synchronization.

IPv4 (Internet Protocol version 4):

This is the fundamental protocol for internet communication, providing addressing and routing for data packets. It ensures that our application's data can be sent and received across different networks.

QUIC (Quick UDP Internet Connections):

Used for secure transport over the internet, QUIC is designed to improve upon traditional TCP by reducing latency and improving performance. It's commonly used for applications that require fast and secure data transfer.

QUIC works by multiplexing data streams, allowing parallel transfer of different parts of a video, leading to faster loading times. Youtube is using QUIC for faster and secure communication.

DNS (Domain Name System):

Essential for translating human-readable domain names into IP addresses, allowing our application to connect to servers by using domain names. DNS is used multiple times in our application for finding the IP of youtube-ui, youtube upload server etc..

TCP (Transmission Control Protocol):

Majorly TCP protocol is used for data communications between the client and the youtube upload server because:

Transmission control protocol is a reliable protocol i.e. it ensures data which is sent reaches the destination successfully. When a sender doesn’t get an acknowledgement after a certain period of time, it will assume that the packet got lost on its way. So, it will send it again. TCP ensures the ordered delivery of packets. Although packets may come out of order, TCP rearranges them before sending them to application. TCP also ensures proper error handling and flow control mechanisms to minimize the error loss rate as we cannot afford any data loss for communication of form related information.

TLS (Transport Layer Security):

Ensures secure communication over a computer network. It's commonly used to encrypt data between our application and the server, 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 (Lab morning)	**Test Case 2 (Lab Evening)**	**Test Case 3 (Hostel Morning)**	**Test Case 4 (Hostel night)**
Throughput	16.375 MBPS	15.125 MBPS	13.5 MBPS	2.75 MBPS
RTT (ms)	41	50	40	65
Packet size	1147.24	1166.47	1173.10	1169.10
Number of packets lost	0	0	0	0
Number of UDP packets	75201	73304	72597	72601
Number of TCP packets	68	103	59	372
Number of responses received w.r.t request sent	0.10	0.11	0.10	0.10

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.

The whole content, specifically the video data, originates consistently from a singular source, which is our PC (172.116.16.166). Remarkably, the data consistently reaches the identical destination, namely the YouTube upload server (172.217.160.207), on every occasion.