CS 558: Computer Systems Lab

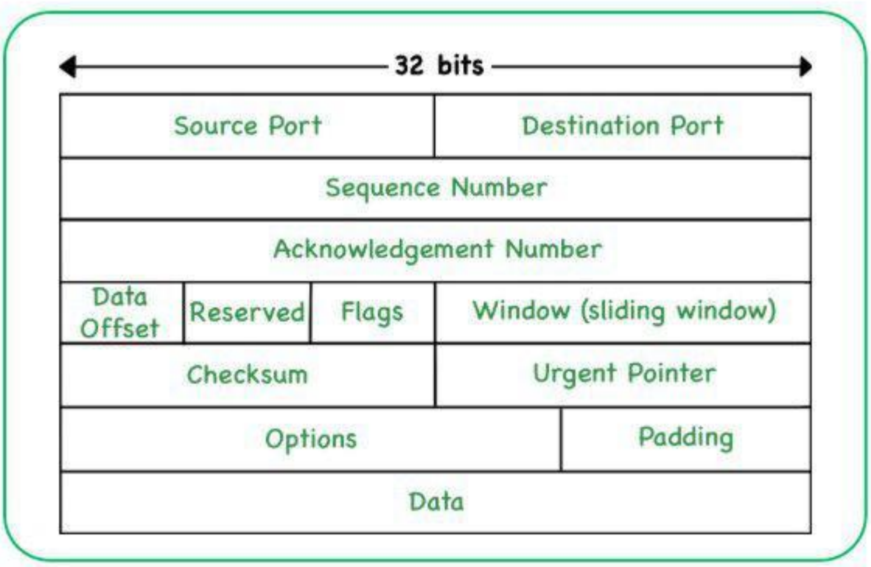
Assignment – 2:
Network Protocol Analysis
Using Wireshark

Shantanu Shrivastav	234101063
Shubham Mourya	234101064
Ayush Agarwal	234101060
Pradoom Varma	234101036

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.

The protocols used are Transmission Control Protocol (TCP) ,User Datagram Protocol (UDP) ,QUIC (Quick UDP Internet Connections) and TLS (Transport Layer Security).

1.Transmission Control Protocol (TCP):



The Transmission Control Protocol (TCP) is a connection-oriented protocol that provides reliable, ordered, and error-checked delivery of data. Here's an explanation of the TCP packet format:-

- Source Port (16 bits): Identifies the sender's port number.
- Destination Port (16 bits): Identifies the receiver's port number.
- Sequence Number (32 bits): Used to reorder segments and to detect missing or duplicate segments.

- Acknowledgment Number (32 bits): If the ACK control bit is set, this field contains the value of the next sequence number the sender of the segment is expecting to receive.

- Data Offset (4 bits): Specifies the size of the TCP header in 32-bit words.

- Reserved (3 bits): Reserved for future use. Must be set to zero.

- Control Flags (9 bits):

- CWR (1 bit): Congestion Window Reduced.
- ECE (1 bit): ECN-Echo. Indicates that the sender is willing to support ECN.
- Urgent (1 bit): Indicates that the Urgent pointer field is significant.
- ACK (1 bit): Indicates that the Acknowledgment field is significant.
- PSH (1 bit): Push Function. Asks to push the buffered data to the receiving application.
- RST (1 bit): Reset the connection.
- SYN (1 bit): Synchronize sequence numbers. Used to initiate a connection.
- FIN (1 bit): No more data from sender.
- Window (16 bits): The size of the sender's receive window. Used for flow control.

- Checksum (16 bits): Used for error-checking of the header and data.

- Urgent Pointer (16 bits): If the URG control flag is set, this field points to the sequence number of the last urgent data byte.

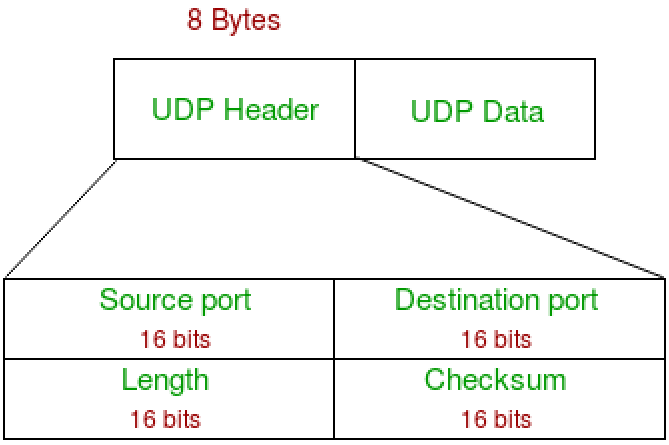
- Options (variable length): Optional field for additional information such as Maximum Segment Size (MSS), window scaling, timestamp, etc.

- Padding (variable length): Used to ensure that the header is a multiple of 32 bits.

- Data (variable length): The actual payload or data being transmitted.

- This header format allows TCP to provide reliable, connection-oriented communication over the Internet. It handles flow control, error recovery, and ensures that data is delivered in order and without duplicates.

2.User Datagram Protocol (UDP):



User Datagram Protocol (UDP) is a connectionless and lightweight transport layer protocol. It provides a simple, low-overhead way to send and receive datagrams, making it suitable for applications where reliability and ordering of packets are not critical. Here's an explanation of the UDP packet format:-

- Source Port (16 bits): Identifies the sender's port number.
- Destination Port (16 bits): Identifies the receiver's port number.
- Length (16 bits): Specifies the length of the UDP header and data in bytes. The minimum length is 8 bytes.
- Checksum (16 bits): Used for error-checking. It is optional, and a value of zero indicates that no checksum is present.
- Data (variable length): The actual payload or data being transmitted.
- UDP is connectionless, meaning it doesn't establish a connection before sending data and doesn't guarantee delivery or order of packets. It is often used for real-time applications like streaming media, online gaming, and DNS, where low latency is more critical than reliability. However, for applications requiring reliable and ordered delivery, TCP is usually the preferred choice.

3.QUIC (Quick UDP Internet Connections):

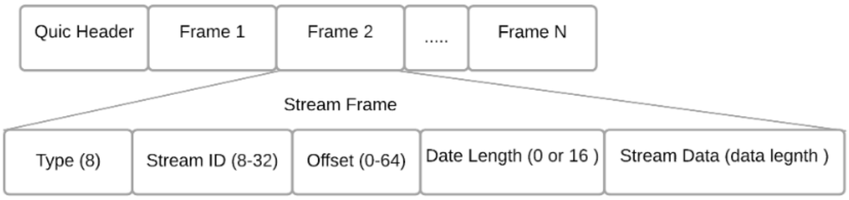
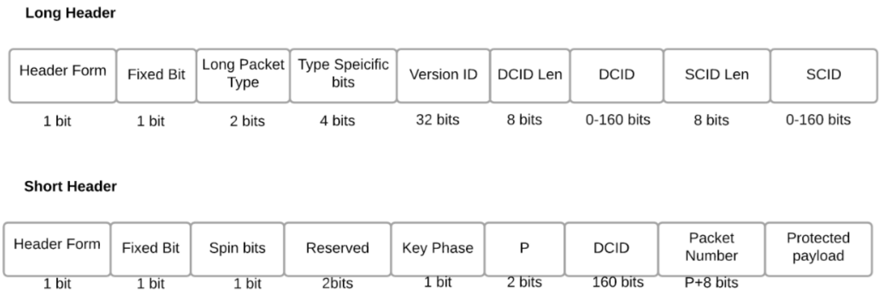


Figure 2:QUIC packet format



Long Header

Header Form	Fixed Bit	Long Packet Type	Type Speicific bits	Version ID	DCID Len	DCID	SCID Len	SCID
1 bit	1 bit	2 bits	4 bits	32 bits	8 bits	0-160 bits	8 bits	0-160 bits

Short Header

Header Form	Fixed Bit	Spin bits	Reserved	Key Phase	P	DCID	Packet Number	Protected payload
1 bit	1 bit	1 bit	2bits	1 bit	2 bits	160 bits	P+8 bits	

Figure 3:QUIC header based on IETF QUIC V1

QUIC (Quick UDP Internet Connections) is a transport layer protocol developed by Google that runs over UDP. It is designed to provide a more efficient and faster alternative to TCP. QUIC is often used for web applications, and it incorporates features like stream multiplexing and improved security. Here's an explanation of the basic QUIC packet format:-

- Flags (8 bits): Various control flags indicating properties of the packet. Flags include:
- Public Reset (R)
- Version Negotiation (VN)
- Long Header (L)
- Key Phase Bit (K)
- Version (32 bits): A version number indicating the QUIC protocol version being used. It is present only in the long header packets.
- Destination Connection ID (64 bits): A unique identifier for the connection. It helps the receiver associate incoming packets with the correct connection.
- Packet Number (8/16/24/32 bits): An identifier for the packet within the context of the connection. It is used for packet ordering and retransmission.
- Payload (variable): The actual data being transmitted. The payload can include QUIC frames such as STREAM, ACK, PING, etc.
- QUIC packets can have either a short header or a long header:
- Short Header: Used for most data packets after the initial handshake. It includes only the flags, connection ID, and packet number.
- Long Header: Used for the initial connection setup and key updates. It includes additional fields like version and destination connection ID.
- QUIC supports features like stream multiplexing, connection migration, and built-in security with encryption. The flexible and extensible nature of QUIC makes it suitable for various applications where low-latency communication is crucial.

4.TLS (Transport Layer Security):

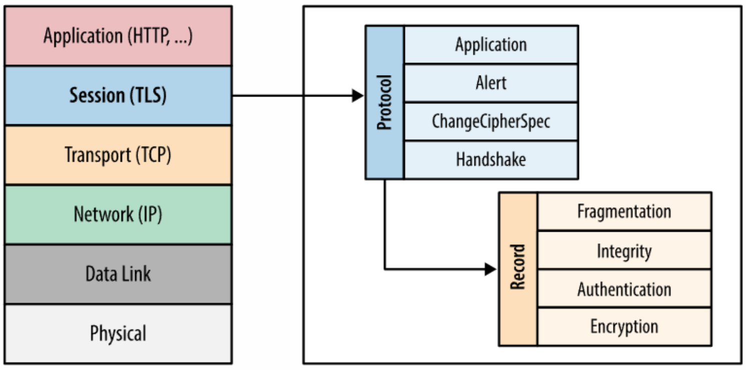


Figure 4-1. Transport Layer Security (TLS)

- TLS (Transport Layer Security) is a protocol that ensures privacy between communicating applications and users on the Internet. TLS operates at the transport layer and provides a secure communication channel over a potentially untrusted network. TLSv1.2 is one of the versions of the TLS protocol. Below is a simplified explanation of the TLSv1.2 packet format:-

- TLS Record Layer:Each TLS message is encapsulated in a record layer. The record layer consists of the following fields:

- Content Type (8 bits): Specifies the type of content contained in the record. Common types include Handshake, Alert, Application Data, and Change Cipher Spec.

- Version (16 bits): Specifies the version of the TLS protocol being used. For TLSv1.2, this field will contain 0x0303.

- Length (16 bits): Indicates the length of the TLS payload, excluding the record layer header.

- TLS Payload (variable): The actual data for the TLS record, depending on the content type.

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.

1. Source IP (Internet Protocol):

The Source IP address refers to the unique numerical label assigned to a device (such as a computer or server) on a network. It identifies the origin or sender of a data packet in an IP network. IP addresses are crucial for routing data across networks, enabling communication between devices.

Observed value – 172.217.166.174

2. Destination IP (Internet Protocol):

Similar to Source IP, the Destination IP address is a unique numerical label assigned to the device that is intended to receive a data packet. It identifies the target or recipient of the packet, facilitating proper routing to the intended destination.

Observed value – 172.16.116.168

3. Port Number:

Ports are used to differentiate between different services or processes running on the same device. A port number is a 16-bit unsigned integer, thus ranging from 0 to 65535. Ports are essential for managing multiple network services on a single device. They work in conjunction with the IP address to direct data to the correct application or service running on a particular device. There are well-known port numbers for common services, such as HTTP (port 80) and HTTPS (port 443).

Observed value – QUIC – 443 , TLS – 465

4. **Ethernet Address:**

Also known as the Media Access Control (MAC) address, the Ethernet address is a unique identifier assigned to each network interface card (NIC) in a device. Unlike IP addresses, which can change based on network configuration, the MAC address is usually hard-coded into the device's hardware. MAC addresses are used at the data link layer to identify devices within the same local network.

Observed value – Source – f8:0b:cb:cb:49:e4
Destination – c8:d9:d2:29:d5:1f

5. **Protocol Number:**

In the context of the Internet Protocol suite, this refers to the protocol field in the IP header. It indicates the specific protocol used in the data portion of the IP packet. For example, a protocol number of 6 indicates that the data portion contains TCP (Transmission Control Protocol) data, while a protocol number of 17 indicates UDP (User Datagram Protocol) data.

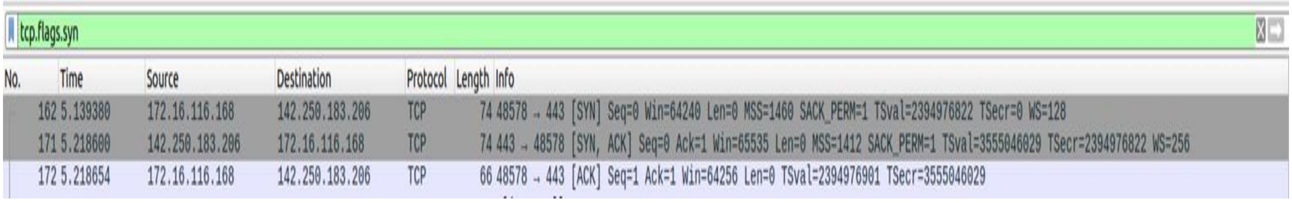
Observed value – UDP - 17

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.

Sequence of messages exchanged are:

1. User Initiates Upload
2. Client-Server Handshake
3. Request Headers
4. Authentication and Authorization
5. Temporary Storage Allocation
6. Data Chunking
7. Data Transmission
8. Server-side Processing
9. Progress Tracking
10. Upload Completion
11. Storage and Post-Processing

Yes, handshaking is performed during our experiment



tcp.flags.syn

No.	Time	Source	Destination	Protocol	Length	Info
162	5.139380	172.16.116.168	142.250.183.206	TCP	74	48578 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2394976822 TSecr=0 WS=128
171	5.218600	142.250.183.206	172.16.116.168	TCP	74	443 → 48578 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1412 SACK_PERM=1 TSval=3555046029 TSecr=2394976822 WS=256
172	5.218654	172.16.116.168	142.250.183.206	TCP	66	48578 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2394976901 TSecr=3555046029

3-way handshaking involves following steps:

Step 1 - SYN (Synchronize)

Step 2 - SYN-ACK (Synchronize-Acknowledge)

Step 3 - ACK (Acknowledge)

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

YouTube uses a combination of several protocols to facilitate the uploading of videos. These protocols play specific roles in ensuring a secure, reliable, and efficient upload process. Let’s break down how UDP, TCP, QUIC, and TLS contribute to the functioning of uploading videos on YouTube:

1. **TCP (Transmission Control Protocol):**

Role: TCP is a connection-oriented protocol, providing reliable and ordered delivery of data. It establishes a connection between the client (uploader) and the server (YouTube) before transferring data.
Relevance for YouTube Uploads: TCP ensures that data, in this case, the video file, is transmitted accurately and without loss. It guarantees that the video reaches YouTubes servers intact, without corruption or missing parts.

2. **UDP (User Datagram Protocol):**

Role: UDP is a connectionless protocol, providing a faster but less reliable way to send data. It does not establish a connection before sending data and does not guarantee delivery or order of packets.
Relevance for YouTube Uploads: While UDP is generally not used for large file transfers due to its lack of reliability, it can be employed for certain aspects of YouTube uploads, such as transmitting live streaming data where a slight delay is acceptable.

3. **QUIC (Quick UDP Internet Connections):**

Role: QUIC is a transport layer protocol developed by Google that runs over UDP. It aims to provide a faster and more secure alternative to traditional protocols like TCP.
Relevance for YouTube Uploads: QUIC can be used to enhance the upload speed and responsiveness during the video upload process. It is designed to reduce latency

and improve performance, which can be particularly beneficial for large file transfers like video uploads.

4. **TLS (Transport Layer Security):**

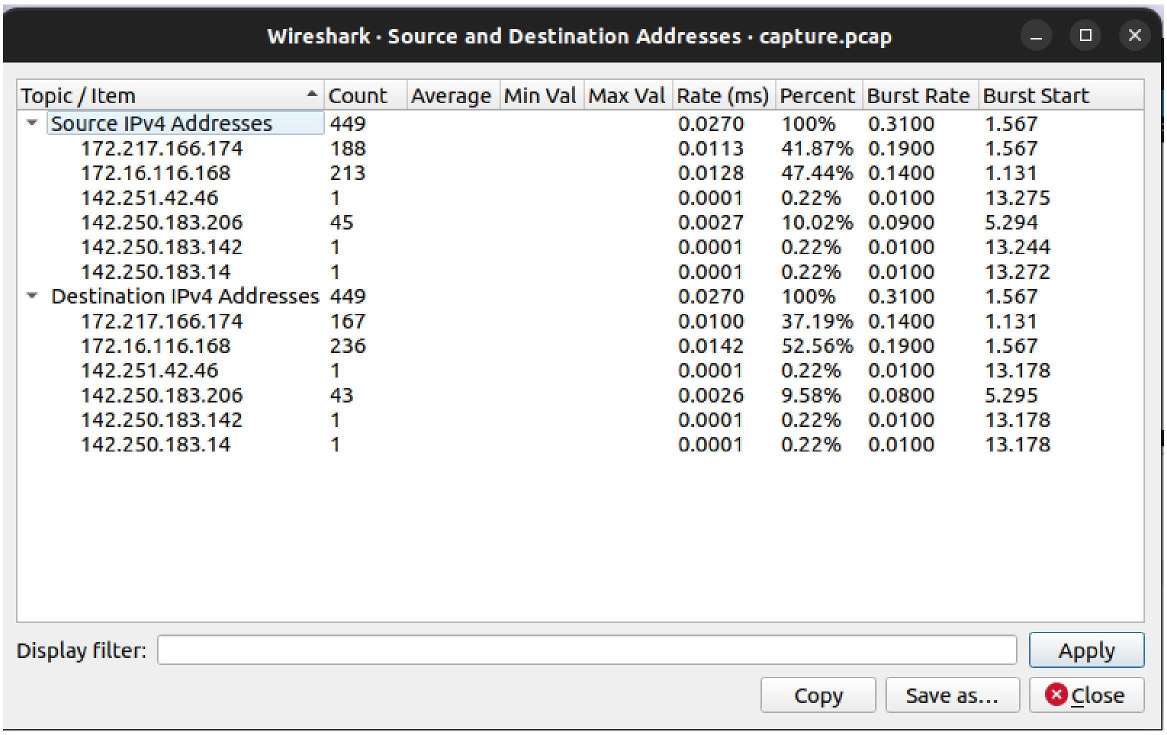
Role: TLS is a cryptographic protocol that ensures secure communication over a computer network. It encrypts data to protect it from unauthorized access or tampering during transmission.
Relevance for YouTube Uploads: TLS is crucial for securing the data during the upload process. It encrypts the communication between the uploader's device and YouTube's servers, safeguarding sensitive information such as login credentials and the content of the video being uploaded.

TCP ensures the reliable delivery of video data, UDP and QUIC may be used for specific performance improvements, and TLS secures the communication channel during the uploading process on YouTube. The combination of these protocols helps ensure a smooth, secure, and efficient video upload experience for users.

5. Calculate the following statistics from your traces while performing experiments at different time 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.

	Lab (14:00)	**Lab (11:00)**	**Hostel (17:00)**
Throughput	30 Mb/s	22Mb/s	24Mb/s
RTT	22 ms	41 ms	26 ms
Packet Size	473 B	506 B	465 B
Number of packets lost	8	0	1
Number of UDP packet	426	277	443
Number of TCP packet	23	2	38
Response received	0.1	0.1	0.1

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.


Wireshark · Source and Destination Addresses · capture.pcap

Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
Source IPv4 Addresses	449				0.0270	100%	0.3100	1.567
172.217.166.174	188				0.0113	41.87%	0.1900	1.567
172.16.116.168	213				0.0128	47.44%	0.1400	1.131
142.251.42.46	1				0.0001	0.22%	0.0100	13.275
142.250.183.206	45				0.0027	10.02%	0.0900	5.294
142.250.183.142	1				0.0001	0.22%	0.0100	13.244
142.250.183.14	1				0.0001	0.22%	0.0100	13.272
Destination IPv4 Addresses	449				0.0270	100%	0.3100	1.567
172.217.166.174	167				0.0100	37.19%	0.1400	1.131
172.16.116.168	236				0.0142	52.56%	0.1900	1.567
142.251.42.46	1				0.0001	0.22%	0.0100	13.178
142.250.183.206	43				0.0026	9.58%	0.0800	5.295
142.250.183.142	1				0.0001	0.22%	0.0100	13.178
142.250.183.14	1				0.0001	0.22%	0.0100	13.178

Display filter:

YouTube may use streaming protocols like DASH (Dynamic Adaptive Streaming over HTTP) or HLS (HTTP Live Streaming). In such cases, the video content may be split into smaller segments, and each segment may be fetched from different servers.

Understanding the presence of multiple sources is often related to YouTube's use of CDNs (Content Delivery Networks) and adaptive streaming technologies. CDNs distribute content to reduce latency and improve user experience by serving content from servers closer to the user. Adaptive streaming allows YouTube to adjust video quality and delivery based on network conditions.