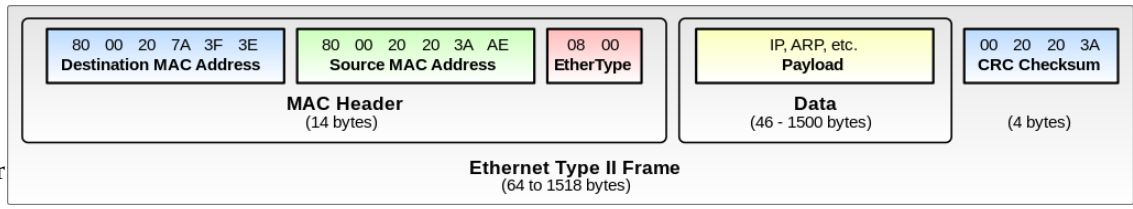
Assignment 2
Name: Tanvi Ohri, Roll No.: 170123051

Internet Connection used: USB tethering through Airtel, Vodafone Mobile networks. In case of common observations, screenshots are attached for Airtel network.

Q1.

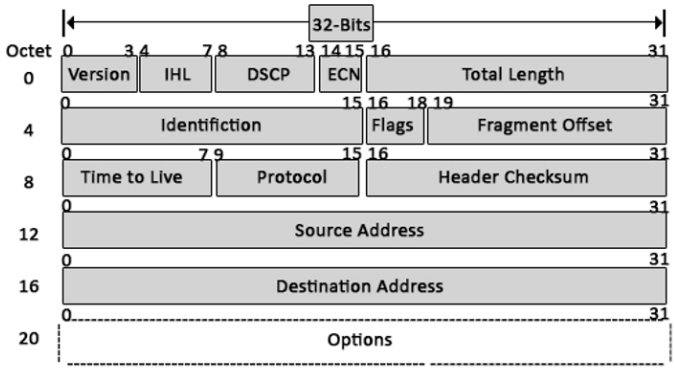
1. Link Layer: Ethernet II



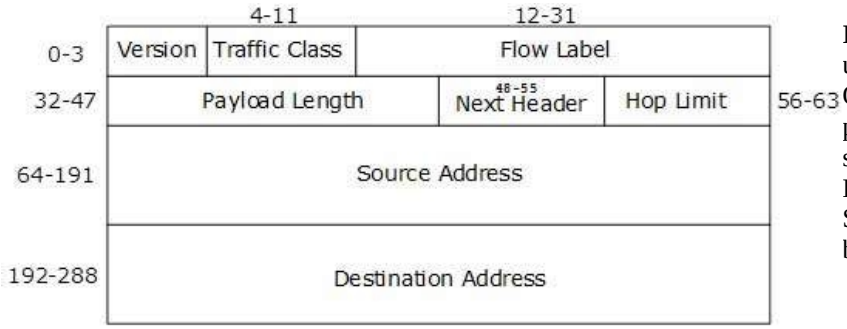
The Destination Address specifies to which adapter the data frame is being sent.The Source Address specifies from which adapter the message originated. The Ethertype specifies the memory buffer in which to place this frame. Following the Ethertype are 46 to 1500 bytes of data, generally consisting of upper layer headers such as TCP/IP or IPX and then the actual user data. CRC checksum is meant for error detection.

2. Network Layer: Internet Protocol v4 for Airtel and Internet Protocol v6 for Vodafone

An IP packet consists of a header section and a data section. Data section holds the payload given to it by the Transport Layer. Headers are explained below:



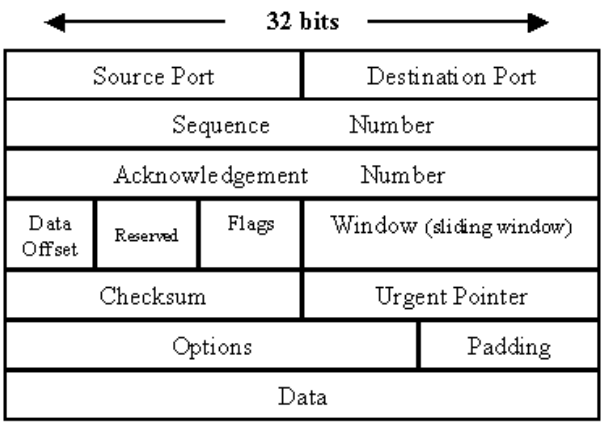
Ipv4: Version: equal to 4, Internet Header Length (IHL) has 4 bits which is the number of 32-bit words in header, Differentiated Services Code Point (DSCP) used in QoS, Explicit Congestion Notification (ECN) allows end-to-end notification of network congestion without dropping packets, Total Length is 16-bit field which defines the entire packet size in bytes, identification field is primarily used for uniquely identifying the group of fragments of a single IP datagram, flags bit is used to control or identify fragments (0 th bit: Reserved and is always 0; 1 st bit: Don't Fragment (DF); 2 nd bit: More Fragments (MF)), Fragment Offset specifies the offset of a particular fragment, Time To Live (TTL) helps prevent datagrams from persisting on network forever, Protocol defines the protocol used in the data portion of the IP datagram, Header Checksum is used for error-checking of the header, Source address & Destination address is the Ipv4 address of the sender and receiver of the packet respectively



Ipv6: Version: IP version 6, Traffic Class: used in congestion control, Flow Label: QoS management, Payload length: payload length in bytes, Next header: specifies the next encapsulated protocol, Hop Limit: replaces the ttl field of Ipv4, Source and Destination addresses: 128 bits each

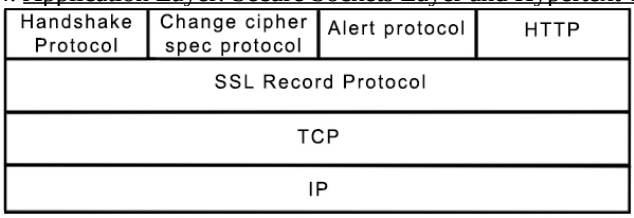
3. Transport Layer: Transmission Control Protocol

TCP segment consists of data bytes to be sent and a header, which can range from 20-60 bytes. The header fields are: Source and destination TCP ports which are the communication endpoints for sending and receiving devices, sequence and acknowledgement numbers to mark the ordering in a group of messages, data offset stores the total size of a TCP header in multiples of four bytes, Reserved data in TCP headers always has a value of zero, a set of six standard and three extended control flags (each an individual bit representing on or off) to manage data flow in specific situations, window size to regulate how much data sender sends to a receiver before requiring an



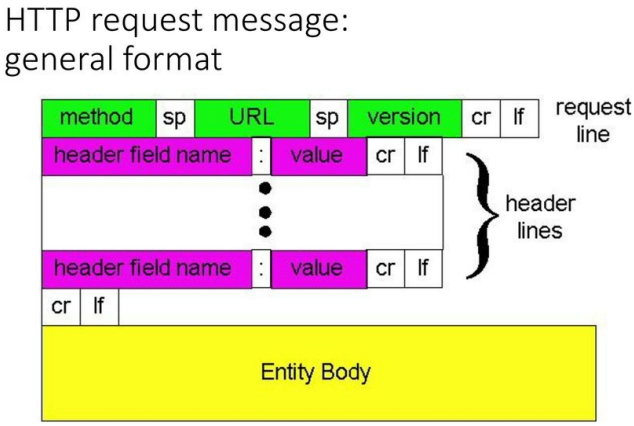
acknowledgment in return, checksum for error detection, urgent pointer can be used as a data offset to mark a subset of a message which require priority processing. Optional TCP data can be used to include support for special acknowledgment and window scaling algorithms.

4. Application Layer: Secure Sockets Layer and Hypertext Transfer Protocol



SSL: It can be used with any protocol that uses TCP as the transport layer. The basic unit of data in SSL is a record. Each record consists of a five-byte record header, followed by data. The header contains – Record Type: four types- Handshake, Change Cipher Spec, Alert, Application Data, Record Version is 16-byte value formatted in network order, Record Length is 16-byte value.

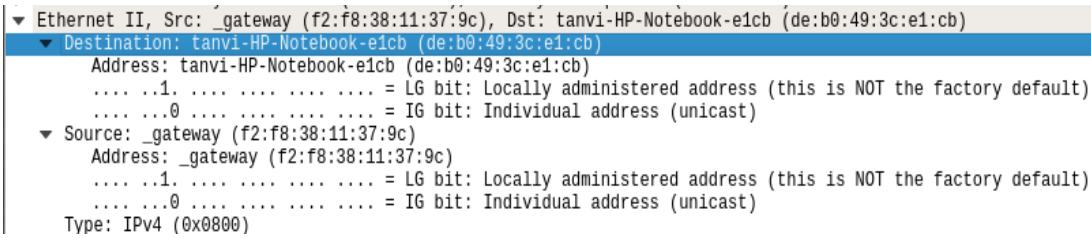
HTTP: Hypertext Transfer Protocol (HTTP) header fields provide required information about the request or response, or about the object sent in the message body. There are four types of HTTP message headers: General-header have applicability for both request and response applicability only for request messages, Server Response-header have applicability only for response messages, Entity-header defines meta information about the entity-body. The header fields are - Connection general-header allows the sender to specify options that are desired for that particular connection and must not be communicated by proxies over further connections, Date, Authorization request-header field value consists of credentials containing the authentication information of the user agent, Cookie request-header field value contains a name/value pair request-header contains an Internet e-mail address for the human user who controls the requesting user agent, Host request-header is used to specify the Internet host and the port number of the resource being requested, Proxy-Authorization request-header allows the client to identify itself to a proxy which requires authentication, User-Agent request-header contains information about the user agent, Location response-header is used to redirect the recipient to a location other than the Request-URI for completion, Server response-header contains information about the software used by the origin server to handle the request.



The general format is similar to the format used for e-mail messages: headers, an empty line and then a message body. All text lines are terminated with the standard “CRLF” control character sequence; the empty line contains just those two characters and nothing else. The headers are always sent as regular text; the body, however, may be either text or 8-bit binary information, depending on the nature of the data to be sent. The start line is a special text line that conveys the nature of the message. In a request, this line indicates the nature of the request, in the form of a method, as well as specifying a URI to indicate the resource that is the object of the request. Responses use the start line to indicate status information in reply to a request.

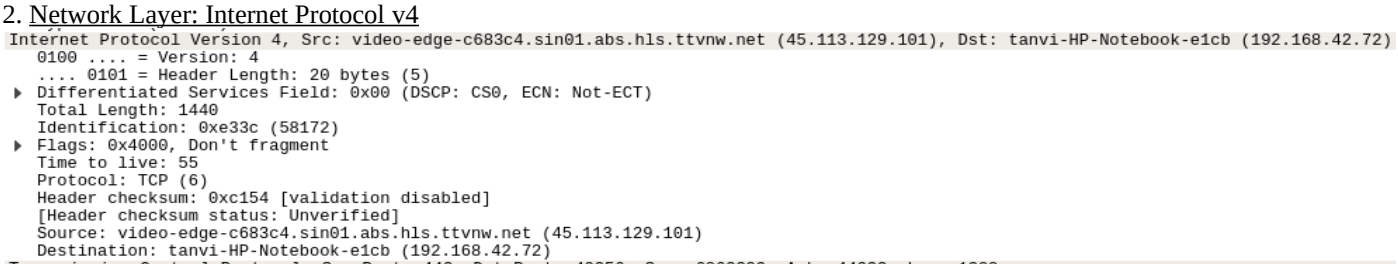
Q2.

1. Link Layer: Ethernet II



Destination contains MAC Address of the destination endpoint. Source contains the MAC address of the sending device, which is my PC. Both addresses are unicast and locally administered. Type indicates which protocol is encapsulated in the payload of the frame.

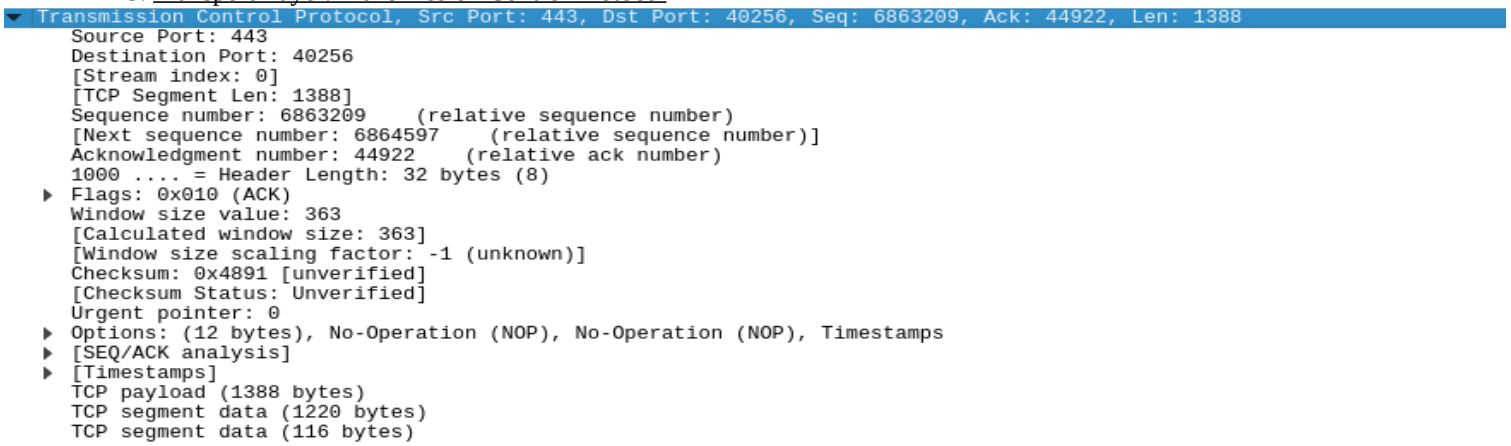
2. Network Layer: Internet Protocol v4



Version is 4 as Ipv4 is being used. Header length is 20 bytes and as it counts in 4 bytes, its value is 5. DCP: CS0 implies best-effort delivery service and ECN: Not-ECT implies Non ECN-Capable Transport. Total Length of the packet is 1440 bytes. Flag is set to “Don’t Fragment” which is an instruction not to break this packet into smaller

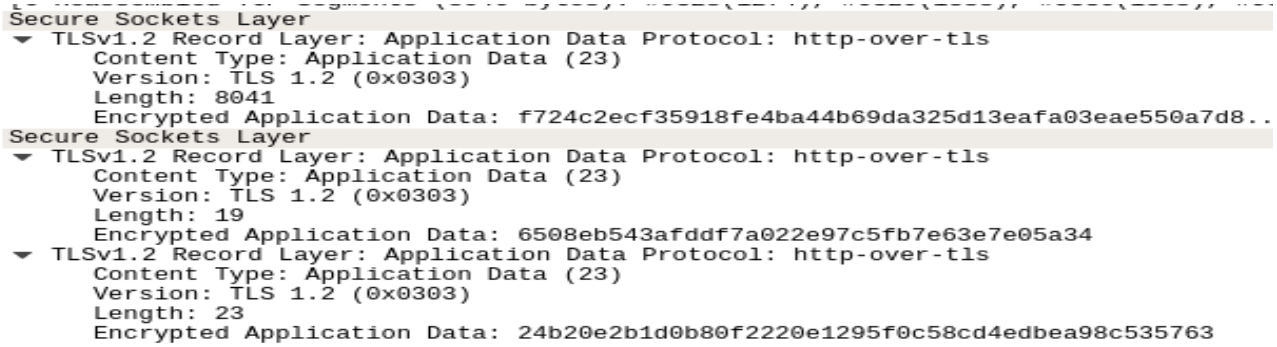
fragment. TTL is 55 hops. Protocol used at the layer above is TCP. Header Checksum is used for error detection of the packet header. Source and Destination contains the IP address of the website and my PC respectively.

3. Transport Layer: Transmission Control Protocol



The Source and Destination Ports are the port numbers on source and destination among which the communication is happening. Header Length is 32 bytes, so value 8 is stored (as counting happens in 4 bytes). Flags ACK is set which means that the machine sending the packet with ACK is acknowledging data that it had received from the other machine. The Window Size Value indicate how much buffer space is available on source for receiving packets. Checksum is used for error detection in the entire packet. Scaling Factor is the number by which the window size displayed is to be scaled.

4. Application Layer: Secure Sockets Layer



The Application-Data-Protocol indicates the protocol used at the application layer, which is http (secured with tls) here. Content Type indicates the type of payload, which is the Application Data here. Version tells the protocol version used for securing the communication, which is TLS 1.2 here. Length is the length of the message. Encrypted Application Data is unreadable by anyone who does not have the required key.

5. Application Layer: Hypertext Transfer Protocol

GET request method is being used to retrieve the stream from the server. Host specifies the address where the resource is located. Connection: keep-alive tells the server to maintain the connection. User-Agent indicates the software program that is acting on behalf of the user.

Q3.

Data Privacy and Integrity is of utmost concern in today's day and age. An SSL(Secure Sockets Layer) certificate is a bit of code on your web server that provides security for online communications. When a web browser contacts your secured website, the SSL certificate enables an encrypted connection. TLS (Transport Layer Security) is used to encrypt the data between the communication endpoints. Hence, these two provide data privacy and data integrity.

UDP is faster, which is a desirable quality for a live streaming website. However, TCP is more reliable and compatible with HTTP than UDP. Hence, Twitch uses TCP at Transport Layer. Less speed is compensated because of increased bandwidths during distribution (from CDN (Content Delivery Network) where the data is first sent).
At network layer, IP (Internet Protocol) is used because TCP is only compatible with it. TCP ensures reliable delivery and correct order of data.
Ethernet II is used at link layer as it ensures reliable data transfer, proper error handling (CRC), synchronization (preamble) and flow control.
Internet Control Message Protocol version 6 (ICMPv6) is a supporting protocol and used by networks devices for sending the error messages and operations information. Since IP does not have a inbuilt mechanism for sending error and control messages, it depends on Internet Control Message Protocol(ICMP) to provide an error control. It is used for reporting errors and management queries.

Streaming and Pausing Functionalities: On playing a video on the application, the server begins to send Application Data to the PC (HTTP encrypted by TLS used), these packets are acknowledged by a packet from my PC (TCP protocol is used). Sometimes, acknowledgment packets may carry data, called piggy-backing (identified by non-zero length).
While the video is paused, KEEP-ALIVE messages are periodically exchanged to keep the connection alive (TCP protocol is used for this).

Q4.

Streaming and Pausing Functionalities: On playing a video on the application, the server begins to send Application Data to the PC, these packets are acknowledged by a packet from my PC. Sometimes, acknowledgment packets may carry data, called piggy-backing (identified by non-zero length).

37	0.000069918	video-weaver.sin01.…	tanvi-HP-Notebook-d…	TCP	1454 443 → 60584 [ACK] Seq=8658 Ack=1231 Win=93 Len=1388 TSval=3431548666 TSecr=2479319354 [TCP …
38	0.000009507	tanvi-HP-Notebook-d…	video-weaver.sin01.…	TCP	66 60584 → 443 [ACK] Seq=1231 Ack=10046 Win=1339 Len=0 TSval=2479319534 TSecr=3431548666
39	0.000027443	video-weaver.sin01.…	tanvi-HP-Notebook-d…	TLSv1.2	764 Application Data
40	0.000010874	tanvi-HP-Notebook-d…	video-weaver.sin01.…	TCP	66 60584 → 443 [ACK] Seq=1231 Ack=10744 Win=1361 Len=0 TSval=2479319534 TSecr=3431548666
41	0.006772664	tanvi-HP-Notebook-d…	video-edge-6ea6a0.m…	TLSv1.2	1185 Application Data
42	0.273797748	tanvi-HP-Notebook-d…	server-13-227-143-5…	TCP	66 55958 → 443 [ACK] Seq=1 Ack=1 Win=477 Len=0 TSval=2100264439 TSecr=1169264316

Streaming and pausing lead to communication between different ports (eg. 60584 and 47928 respectively, at 5pm) on my PC and the same port 443 on the twitch server.
While the video is paused, KEEP-ALIVE messages are periodically exchanged to keep the connection alive.

	Time	Source	Destination	Protoco	Length	Info
655	0.089959982	tanvi-HP-Notebook-d…	static.twitchcdn.net	TCP	66	[TCP Keep-Alive] 47804 → 443 [ACK] Seq=1092 Ack=582 Win=31360 Len=0 TSval=1918868328 TSe
656	0.229696952	static.twitchcdn.net	tanvi-HP-Notebook-d…	TCP	66	[TCP Keep-Alive ACK] 443 → 47804 [ACK] Seq=582 Ack=1093 Win=30720 Len=0 TSval=2019735928
657	0.496539505	_gateway	tanvi-HP-Notebook-d…	ARP	42	Who has 192.168.42.116? Tell 192.168.42.129
658	0.000021258	tanvi-HP-Notebook-d…	_gateway	ARP	42	192.168.42.116 is at 16:f9:c5:58:d1:8b
659	1.833685903	tanvi-HP-Notebook-d…	twitch.map.fastly.n…	TCP	66	[TCP Keep-Alive] 47926 → 443 [ACK] Seq=1050 Ack=733 Win=31488 Len=0 TSval=160660825 TSec
660	0.551391633	twitch.map.fastly.n…	tanvi-HP-Notebook-d…	TCP	66	[TCP Keep-Alive ACK] 443 → 47926 [ACK] Seq=733 Ack=1051 Win=30720 Len=0 TSval=938699360
661	0.668435686	tanvi-HP-Notebook-d…	safebrowsing.google…	TLSv1.3	105	Application Data
662	0.120895289	safebrowsing.google…	tanvi-HP-Notebook-d…	TLSv1.3	105	Application Data

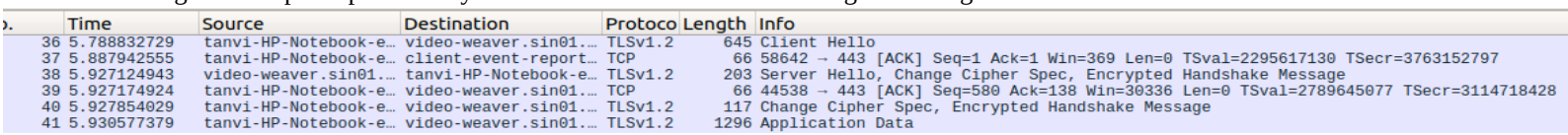
Handshaking Sequences:
-> DNS querying: First, when the site is loaded, DNS querying is done by the browser. A series of messages are exchanged, so that the browser may learn the IP address of the website.

320	0.086260992	tanvi-HP-Notebook.l…	2401:4900:38cf:a756…	DNS	93 Standard query 0xeb16 A www.twitch.tv
321	0.000209904	tanvi-HP-Notebook.l…	2401:4900:38cf:a756…	DNS	93 Standard query 0x77a8 AAAA www.twitch.tv
322	0.080021405	2401:4900:38cf:a756…	tanvi-HP-Notebook.l…	DNS	216 Standard query response 0xeb16 A www.twitch.tv CNAME twitch.map.fastly.net A 151.101.38.167 N
323	0.116545359	2401:4900:38cf:a756…	tanvi-HP-Notebook.l…	DNS	189 Standard query response 0x77a8 AAAA www.twitch.tv CNAME twitch.map.fastly.net SOA ns1.fastly.
324	0.000370915	tanvi-HP-Notebook.l…	2401:4900:38cf:a756…	DNS	101 Standard query 0x0c6b AAAA twitch.map.fastly.net
325	0.001751472	2401:4900:38cf:a756…	tanvi-HP-Notebook.l…	DNS	101 Standard query response 0x0c6b AAAA twitch.map.fastly.net
326	0.000464269	tanvi-HP-Notebook-d…	twitch.map.fastly.n…	TCP	74 48898 → 443 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=1099549661 TSecr=0 WS=128

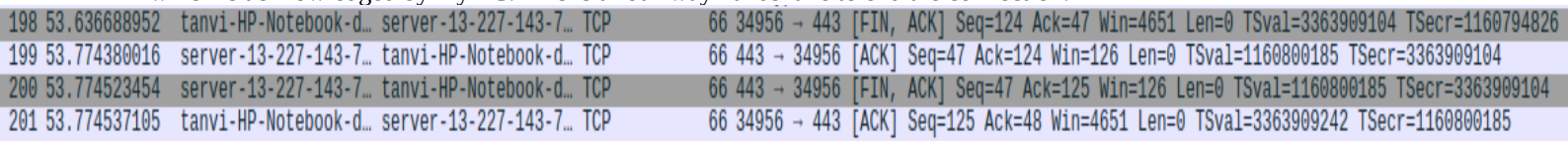
-> TCP Handshaking: A ‘three-way handshake’ between the source (my PC) and the destination (the website server). Port 44538 on my computer and port 443 on the destination system are being used. First, a SYN packet is sent from my PC to the server to establish connection and to synchronize the sequence number. The server replies with a SYN-ACK, acknowledging that is has received the initial packet, and in the same packet, it tell the source to synchronize with its sequence number.

	Time	Source	Destination	Protoco	Length	Info
33	5.618712379	tanvi-HP-Notebook-e…	video-weaver.sin01.…	TCP	74	44538 → 443 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2789644768 TSecr=0 WS=128
34	5.784772464	video-weaver.sin01.…	tanvi-HP-Notebook-e…	TCP	74	443 → 44538 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1400 SACK_PERM=1 TSval=3114718282 TSe
35	5.784842225	tanvi-HP-Notebook-e…	video-weaver.sin01.…	TCP	66	44538 → 443 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=2789644934 TSecr=3114718282

-> TLS Handshaking: To make the connection secure, TLSv1.2 is being used. The first message in the TLS Handshake is the Client Hello which is sent by the client to initiate a session with the server. The server responds with Server Hello. The change cipher spec message, transmitted by both the client and the server, defines the re-negotiated cipher spec and keys that will be used for all the messages exchanged henceforth.



-> Connection Termination: When the communication is over, there is an exchange of messages to terminate the TCP connection. My PC first sends a FIN-ACK message. The server acknowledges it and sends its own FIN-ACK message, which is acknowledged by my PC. This is a four-way handshake to end the connection.



Q5.

->Streaming Video

Time	Throughput	RTT (ms)	Packet Size	Lost Packets	TCP packets	UDP packets	Responses received
8 am	5105k bits/s	1.826	1008 B	1 (0.0%)	21141 (97.6%)	502 (2.3%)	0.67
6 pm	4124k bits/s	1.792	948 B	0 (0.0%)	16473 (95.6%)	736 (4.3%)	0.69
12 am	3759k bits/s	2.142	920 B	5 (0.0%)	14163 (96.7%)	475 (3.2%)	0.70

->Paused Video

Time	Throughput	RTT (ms)	Packet Size	Lost Packets	TCP packets	UDP packets	Responses received
8 am	538k bits/s	3.704	786 B	2 (0.0%)	6041 (90.5%)	608 (9.1%)	0.70
6 pm	281k bits/s	5.279	707 B	1 (0.0%)	7036 (90.9%)	657 (8.5%)	0.66
12 am	425k bits/s	4.911	651 B	1 (0.0%)	5339 (90.5%)	536 (9.1%)	0.86

Q6.

The IP addresses of the source server are different at different times. This maybe because of different network traffic at different times. Hence, some load balancing takes place, where some requests are redirected to a different server, thus reducing network congestion.

Time	IP Address
8 am	185.42.207.190
6 pm	185.42.207.125
12 am	45.113.129.78

->Drive Link for Wireshark Traces: *https://drive.google.com/open?id=1jOAVw7mkYc-bYDZKvwBZrSuEokLMljHW*