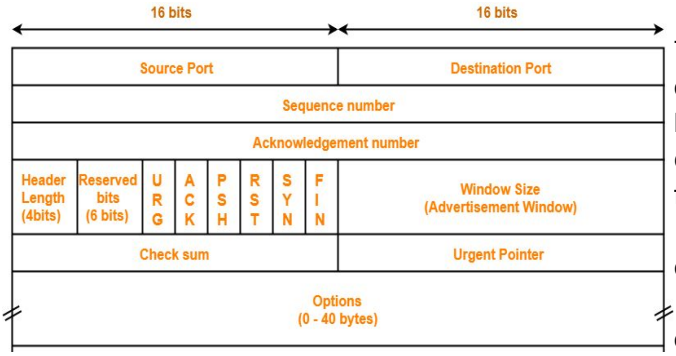
CS-349 | NETWORKS LAB
ASSIGNMENT-2 | Network Protocol Analysis Using Wireshark | COURSERA VIDEOS

VAKUL GUPTA | 170101076

Google drive link for traces :- https://drive.google.com/open?id=1jJpJ4gmhkh4WPx4JdqX-xNlJAE6SYlzK

QUESTION-1 : PROTOCOLS USED BY THE APPLICATION AT DIFFERENT LAYERS :-

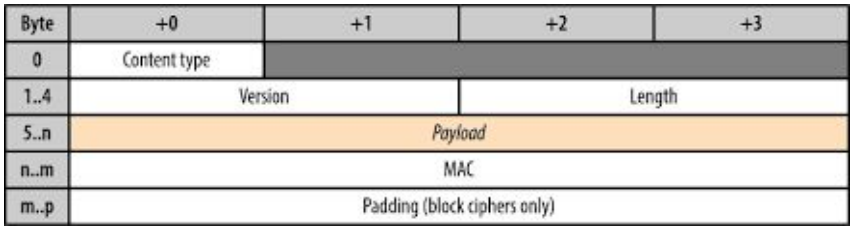
a) TCP (Transmission Control Protocol, Transport Layer) :-



TCP provides reliable, ordered, and error-checked delivery of a stream of octets (bytes) between applications running on hosts communicating by an IP network. A TCP packet is a sequence of octets (bytes) and consists of a header followed by a body. The body contains the data from the layer above it.**The header describes the following :-**

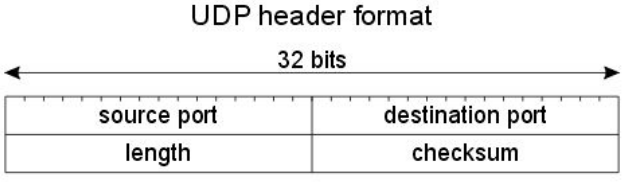
- **Source and Destination Port (2 bytes each)**:- communication endpoints for sending and receiving devices.
- **Sequence Number and Acknowledgment Number (4 bytes each)**:- Sequence numbers of the packets.
- **TCP data offset (4 bits):-** the total size of a TCP header in multiples of four bytes.
- **Reserved data (3 bits):-** This field serves the purpose of aligning the total header size as a multiple of four bytes.
- **Control flags (up to 9 bits):-** to manage data flow in specific situations.
- **Window size (2 bytes):-** to regulate how much data they send to a receiver before requiring an acknowledgment in return
- **TCP checksum (2 bytes):-** It verifies the integrity of data in the TCP payload.
- **Error Control Urgent pointer (2 bytes):-** to regulate how much data they send to a receiver before requiring an acknowledgment in return.
- **TCP optional data (0-40 bytes):-** to regulate how much data they send to a receiver before requiring an acknowledgment

b) TLS (Transport layer Security, Application Layer):-



- TLS protocol sits between the Application Layer and the Transport Layer.
- It aims primarily to provide privacy and data integrity between two communicating computer applications.
- The connection is private because symmetric cryptography is used to encrypt the data transmitted.
- The keys for this symmetric encryption are generated uniquely for each connection and are based on a shared secret negotiated at the start of the session(TLS Handshake).
- The connection ensures integrity because each message transmitted includes a message integrity check using a message authentication code to prevent undetected loss or alteration of the data during transmission.
- **The TLS header comprises three fields, necessary to allow the higher layer to be built upon it:**
  - ➢ **Byte 0:** TLS record type(example : CHANGE_CIPHER_SPEC , ALERT, HANDSHAKE,APPLICATION DATA)
  - ➢ **Bytes 1-2:** TLS version (major/minor) (example : TLSv1.0, TLSv1.1, TLSv1.2)
  - ➢ **Bytes 3-4:** Length of data in the record (excluding the header itself). The maximum supported is 16384 (16K). The TLS record contains the TLS protocol for example: Handshake protocol or changecipherspec protocol.

c) UDP (User Datagram Protocol, Transport Layer):

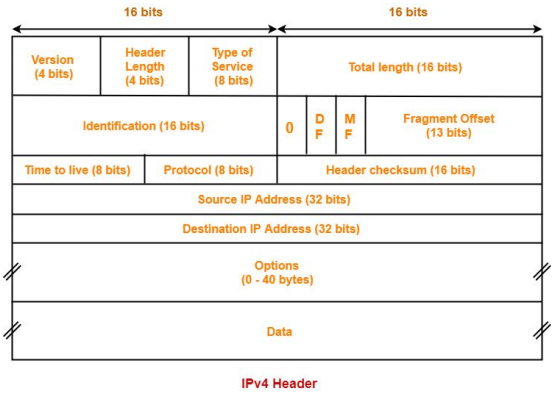


It is the simplest transport layer protocol. It simply takes the datagram from the network layer, attaches its header and sends it to the user. **The header contains only 4 fields:**

- **Source Port:** It is a 16-bit field and identifies the port of sender application

- **Destination Port:** It identifies the port of receiver application.
- **Length:** It identifies the combined length of UDP Header and Encapsulated data.
- **Checksum:** It is calculated on UDP Header, encapsulated data and IP pseudo-header and used for error control.

d) IPv4 (Internet Protocol, Network Layer):-

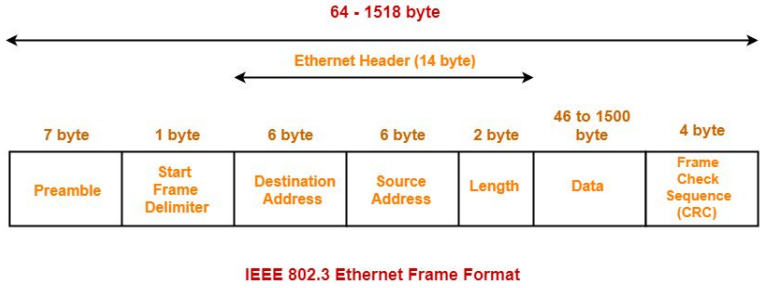


IPv4 Header

It is one of the core protocols of standards-based internetworking methods on the Internet.
IP is responsible for delivering data packets from the source host to the destination host.
IPv4 is a connectionless protocol for use on packet-switched networks. **IP headers contain:**

- **Version:** Indicates the IP version used.
- **Header Length:** Contains the length of the IP header.
- **Types of Services:** Used for Quality of Service(QoS).
- **Total Length:** It is a 16-bit field that contains the total length of the datagram (in bytes).
- **Identification:** It is used for the identification of the fragments of an original IP datagram.
- **DF/MF bits:** DF bit stands for Do Not Fragment and MF stands for More Fragment bits.
- **Time to Live:** It indicates the maximum number of hops a datagram can take to reach the destination.
- **Protocol:** It tells the network layer at the destination host to which protocol the IP datagram belongs.
- **Source/Destination IP address:** It contains the logical address of the sender and receiver of the datagram.

e) Ethernet II (Data Link layer):



IEEE 802.3 Ethernet Frame Format

Data link layer protocols are point to point protocols. The data link layer is concerned with local delivery of frames between devices on the same LAN.
Ethernet Frame Format:

- **Preamble(7 bytes) :** This is a stream of bits (alternating 1's and 0's) used to allow the transmitter and receiver to synchronize their communication.
- **SFD(1 bytes) :** This is always 10101011 and is used to indicate the beginning of the frame information.
- **Destination/Source MAC(6 bytes each):** This is the MAC address of the machine receiving/sending data. The MAC address is 6bytes long. Initial 3 bytes are unique to each manufacturer. Last 3 bytes are unique to each hardware.
- **Length(2 bytes):** This is the length of the entire Ethernet frame in bytes.
- **Data (Payload):** The data is inserted here. This is where the data from the IP layer is placed.
- **FCS(4 bytes):** This field contains the Frame Check Sequence (FCS) which is calculated using a Cyclic Redundancy Check (CRC). The FCS allows Ethernet to detect errors in the Ethernet frame and reject the frame if it appears damaged.

QUESTION-2 : OBSERVED FIELD VALUES FOR THE PROTOCOLS USED :-

The protocols used by different functions of the **application Coursera Videos** were :- **Ethernet II, IPV4, TCP & TLS**. Laptop was connected to the **JIO hotspot.**

```
▸ Ethernet II, Src: Cisco_74:60:42 (ec:44:76:74:60:42), Dst: Dell_03:5f:a3 (d8:d0:90:03:5f:a3)
▸ Internet Protocol Version 4, Src: 172.217.26.174, Dst: 10.19.2.123
▸ Transmission Control Protocol, Src Port: 443, Dst Port: 34706, Seq: 1, Ack: 1, Len: 56
▸ Transport Layer Security
```

Ethernet-II:-

```
▾ Ethernet II, Src: Cisco_74:60:42 (ec:44:76:74:60:42), Dst: Dell_03:5f:a3 (d8:d0:90:03:5f:a3)
  ▸ Destination: Dell_03:5f:a3 (d8:d0:90:03:5f:a3)
  ▸ Source: Cisco_74:60:42 (ec:44:76:74:60:42)
    Type: IPv4 (0x0800)
```

- **Destination: Dell_03:5f:a3 (d8:d0:90:03:5f:a3) :-** Hardware MAC address of my Laptop (Source).
- **Source: Cisco_74:60:42 (ec:44:76:74:60:42) :-** Hardware MAC address of destination.
- **Type :-** The last part of this header indicates which protocol is encapsulated in the payload of the frame.**It had the value 0x0800.**

IPV4:-

```
Internet Protocol Version 4, Src: 172.217.26.174, Dst: 10.19.2.123
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 108
    Identification: 0xef98 (61336)
  ▸ Flags: 0x0000
    ...0 0000 0000 0000 = Fragment offset: 0
    Time to live: 62
    Protocol: TCP (6)
    Header checksum: 0xb8de [validation disabled]
    [Header checksum status: Unverified]
    Source: 172.217.26.174
    Destination: 10.19.2.123
```

- The **version of IP is 4**(IPv4). The header length is **20 bytes** and each word contains 4 bytes, hence a total **5 words** are present in the header length.
- **CS0 implies** that the service type is the best effort and **Not-ECT implies** that the packet is not using ECN Transport.
- Total Length of the packet is **108 bytes.**
- The **fragment offset** is 0 and all **Don't Fragment, Reserve Bit and More Fragment** are set to 0.
- The value of **Time to live(TTL)** is **62** hops.
- **Header checksum (0xb8de)** is used to detect any error.
- **Source and destination** contain the IP address of sender and receiver machine.

TCP :-

```
Transmission Control Protocol, Src Port: 443, Dst Port: 34706, Seq: 1, Ack: 1, Len: 56
    Source Port: 443
    Destination Port: 34706
    [Stream index: 27]
    [TCP Segment Len: 56]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 57    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    1000 .... = Header Length: 32 bytes (8)
  ▸ Flags: 0x018 (PSH, ACK)
    Window size value: 343
    [Calculated window size: 343]
    [Window size scaling factor: -1 (unknown)]
    Checksum: 0x88c4 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▸ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  ▸ [SEQ/ACK analysis]
  ▸ [Timestamps]
    TCP payload (56 bytes)
```

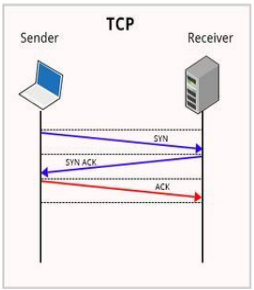
- The **source and destination ports** are the numbers of ports of the server and my laptop between which communication is happening.
- The **header length is 32 bytes** and hence a **total of 8 words** are present.
- Here both the **sequence** and **acknowledgement** number is 1.
- Total of two flags is set (**Push and Acknowledgment**). Here **ACK** means that the machine sending the packet is acknowledging data that is received from another end. **PSH** is an indication to push the entire buffer immediately to the receiving application.
- The **Window Size Value** on packets from A to B indicates how much buffer space is available on A for receiving packets and its current value is **343.**
- The **checksum** is again used for error detection in the entire packet.
- **Scaling Factor** is the number by which the window size displayed is to be scaled.
- Also, the value of the urgent pointer is 0 meaning no data bytes are urgent in the current segment.

TLSv1.2:-

```
Transport Layer Security
  TLSv1.2 Record Layer: Application Data Protocol: http-over-tls
      Content Type: Application Data (23)
      Version: TLS 1.2 (0x0303)
      Length: 51
      Encrypted Application Data: 2c83da89cedb4d9d6277a7038aa28f1338d530b17e2172e6…
```

- The **content type** of the fragment is **application data.**
- The **version of TLS is v1.2**.It is the newest SSL protocol version.
- The total length of the fragment is **51 bytes.**
- **Data encryption** translates data into another form, or code, so that only people with access to a secret key (formally called a decryption key) or password can read it. Encrypted data is commonly referred to as ciphertext.

QUESTION-3 : The RELEVANCE OF THE PROTOCOLS FOR THE APPLICATION:-



Protocols used by our network to **Play, Download the Coursera Videos** are explained below with their functions. The protocols used were **TCP, IPV4, Ethernet -II and TLSv1.2**.

- **TCP:**

  ➔ Its responsibility includes **end-to-end message transfer** independent of the underlying network and structure of user data, along with **error control, segmentation, flow control**, and helps to minimize traffic congestion control.

  ➔ It is a **connection-oriented** protocol that addresses numerous reliability issues in providing a reliable byte stream: **data arrives in-order**, data has minimal error (i.e., correctness), duplicate data is discarded and lost or discarded packets are resent.

- ➔ TCP is optimized for **accurate delivery** rather than timely delivery, as the correct sequence of buffer to be fetched. TCP's **bandwidth probing** and congestion control will attempt to use all of the available bandwidth between the server and client.

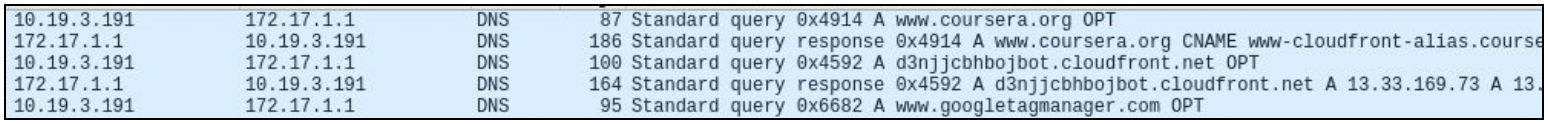
- **TLSv1.2:**
  - ➔ TLS was designed to operate on **top of a reliable transport protocol** such as TCP. Web servers use **cookies** to identify the web user. They are small pieces of data stored into the web user's disk. TLS is used to **protect session cookies** on the rest of the sites from being intercepted to protect user accounts.
  - ➔ The TLS protocol aims primarily to provide **privacy** and **data integrity** between two communicating computer applications. SSL and TLS are both cryptographic protocols utilizing X.509 certificates, public/private key encryption that provides authentication and data encryption between servers, machines and applications operating over a network.
  - ➔ TLS provides verification of identity of server, which is as important as **encryption**.
  - ➔ The goals of the TLS protocol are **cryptographic security, extensibility, and relative efficiency**. These goals are achieved through implementation of the TLS protocol on two levels: the TLS Record protocol and the TLS Handshake protocol.
  - ➔ TLS allows the peers to negotiate a shared secret key without having to establish any prior knowledge of each other, and to do so over an unencrypted channel, Client-server applications use the TLS protocol to communicate across a network in a way designed to prevent eavesdropping and tampering.
- **Ethernet II:**
  - ➔ It is used at the link layer as it ensures **reliable data transfer** between two nodes and involves proper error handling and flow control mechanisms to minimize errors.
  - ➔ Other features include CRC for **error detection** and preamble for **synchronization**.
- **IPv4:**
  - ➔ The IP has the job of delivering packets using the IP headers from the source to the destination.
  - ➔ Existing in the network layer, IPv4 connection is a **hop to hop** connection.

QUESTION-4 : THE SEQUENCE OF MESSAGES EXCHANGED:-

Note:- The laptop (source) was connected to **JIO hotspot** for performing the following experiment.

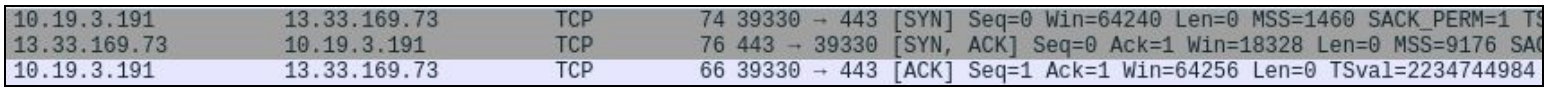
Messages exchanged by the application:

- **DNS querying:**
  First, when the site is loaded, DNS querying is done by the browser. DNS querying is a demand for information sent from the user's computer (DNS client) to a DNS server to ask for the IP address associated with the corresponding domain name. As we can see, the querying is for 'A' record type, which is used to relate IP addresses with domain names.



10.19.3.191	172.17.1.1	DNS	87	Standard query 0x4914 A www.coursera.org OPT
172.17.1.1	10.19.3.191	DNS	186	Standard query response 0x4914 A www.coursera.org CNAME www-cloudfront-alias.course
10.19.3.191	172.17.1.1	DNS	100	Standard query 0x4592 A d3njjcbhbojbot.cloudfront.net OPT
172.17.1.1	10.19.3.191	DNS	164	Standard query response 0x4592 A d3njjcbhbojbot.cloudfront.net A 13.33.169.73 A 13.
10.19.3.191	172.17.1.1	DNS	95	Standard query 0x6682 A www.googletagmanager.com OPT

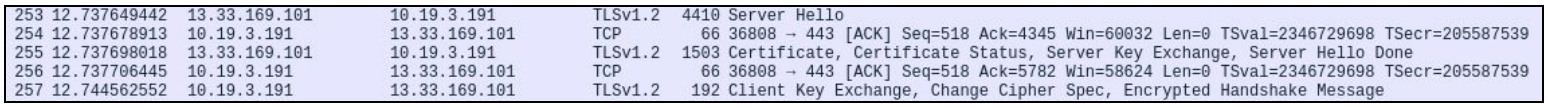
- **TCP connection Establishment:**
  To establish a connection, **TCP uses a three-way handshake**. Before a client attempts to connect with a server, the server must first bind to and listen at a port to open it up for connections: this is called a **passive open**. Once the passive open is established, a client may initiate an active open. **The three-way handshake:**
  - ➔ **SYN:** The active open is performed by the client sending a SYN to the server. The client sets the segment's sequence number to some value A.
  - ➔ **SYN-ACK:** In response, the server replies with a SYN-ACK (acknowledging our SYN request). The ack number is set to one more than the received sequence number (A + 1), and the sequence number that the server chooses for the packet is another number, B.
  - ➔ **ACK:** Finally, the client sends an ACK back to the server to acknowledge the SYN-ACK packet from the server. The sequence number is set to the received ack value i.e. A + 1, and the ack number is set to one more than the received sequence number i.e. B + 1.



10.19.3.191	13.33.169.73	TCP	74 39330 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TS
13.33.169.73	10.19.3.191	TCP	76 443 → 39330 [SYN, ACK] Seq=0 Ack=1 Win=18328 Len=0 MSS=9176 SAC
10.19.3.191	13.33.169.73	TCP	66 39330 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2234744984

- **TLS Handshaking:**
  The TLS Handshake Protocol is responsible for the authentication and key exchange necessary to **establish or resume secure sessions**. When establishing a secure session, the **Handshake Protocol manages the following**:-

- ➔ The client sends a "**Client hello**" message to the server, along with the client's random value and supported cipher suites.
- ➔ The server responds by sending a "**Server hello**" message to the client, along with the server's random value.
- ➔ The server sends its **certificate to the client for authentication** and may request a certificate from the client.
- ➔ The server sends the "**Server hello done**" message.
- ➔ The client sends a "**Change cipher spec"** notification to the server to indicate that the client will start using the new session keys for hashing and encrypting messages. Client also sends a "**Client finished**" message.
- ➔ Server receives "**Change cipher spec**" and switches its record layer security state to symmetric encryption using the session keys. Server sends "**Server finished**" message to the client.
- ➔ Clients and servers can now exchange application data over the secured channel they have established.

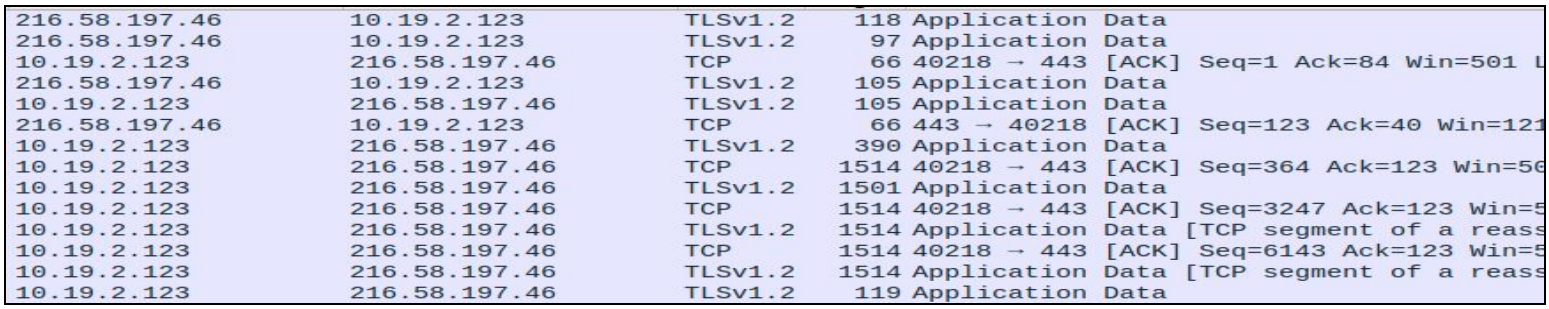


- **Request for Resources**: After handshaking is over and communication is established, the browser sends a CONNECT request to the server and asks for the web page. The server responds by sending the desired resource.

Data flow for different functionalities:-

- **Streaming Videos**

  Once the video starts streaming, the server (Coursera website) sends application data to the client, each of which is Acknowledged by a message from the client by the ACK flag. For multiple TCP segments of a reassembled PDU(Protocol Data Unit) sent from the server, the client sends a cumulative ACK back indicating the first-byte number that it expects from the server. Once all these segments arrive, they are re-assembled and then data is sent to the application layer.

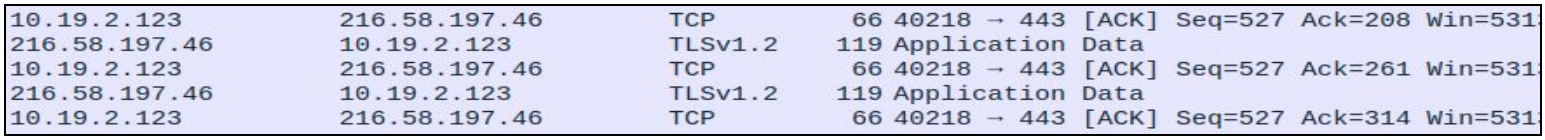


  Operations like playing, pausing etc while streaming can lead to communication between a different port on the client and the same 443 Coursera server port. In this communication, a message will be sent from the client which is replied by the server along with ACK (piggy-backing protocol). When the video is paused, Keep-Alive messages are exchanged to keep the connection alive.

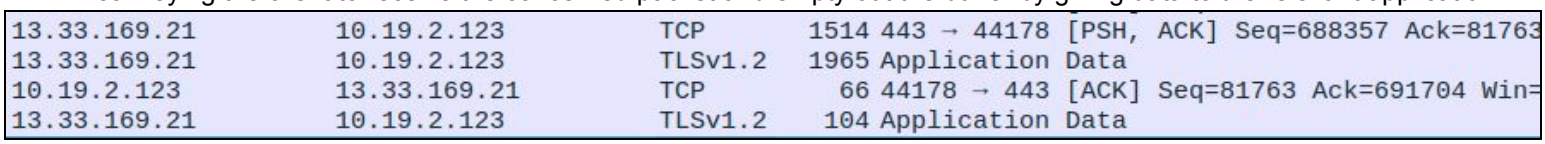
  Pausing a video :- When a playing live video stream was paused, the current ongoing TCP connection was closed and we received a FIN packet from the server.

- **Downloading videos**

  While downloading the videos, the same sequence of messages as streaming was observed. However, the client (my laptop) does not send any message to the server on a different port.



  Also when the video is about to be completely downloaded, the server sends a PSH packet (Push Flag) and hence, conveying the client to receive the concerned packet and empty out the buffer by giving data to the relevant application



QUESTION-5 : TRACE STATISTICS:-

Note:- My laptop was connected to **IITG_Wifi** while taking the following readings for the experiment.
Number of UDP and TCP Packets were taken from **Protocol Hierarchy** under the Statistics Tab.
Avg. Throughput and Avg. Packet Size was taken from **Capture File Properties** under the Statistics Tab.
Packets Lost was counted by using the filter **tcp.analysis.lost_segment**.
Avg. RTT was calculated using **tshark** command.
Responses per request can be calculated by choosing **all addresses in the IPV4 statistics**.

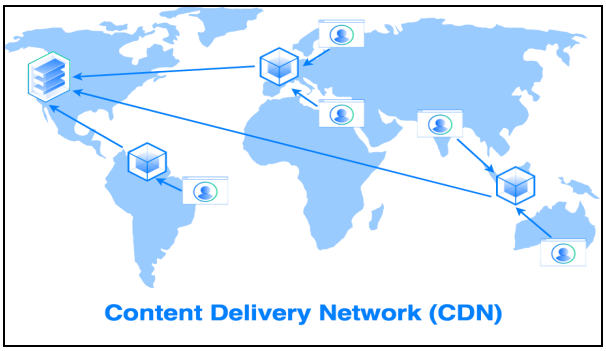
a) Streaming Videos:-

Time of the Day	Avg. Throughput (Bytes/sec)	Avg. RTT (sec)	Avg. Packet Size (Bytes)	Packets Lost	UDP Packets	TCP Packets	Responses per Request
8am	167K	0.00826849	1270	2	132	7560	1
2pm	219K	0.00969073	1248	2	534	7452	1
2am	257K	0.00942703	1530	12	93	12538	1

b) Downloading Videos:-

Time of the Day	Avg. Throughput (Bytes/sec)	Avg. RTT (sec)	Avg. Packet Size (Bytes)	Packets Lost	UDP Packets	TCP Packets	Responses per Request
8am	466K	0.0152688	1518	9	80	22027	1
2pm	423K	0.014734	1523	10	909	20169	1
2am	472K	0.037067	1788	14	20	11263	1

QUESTION-6 : POSSIBILITY OF MULTIPLE IPs:-



My laptop was connected to the **JIO hotspot** and I was able to observe multiple content providers (IP addresses) for the **Coursera Website** which I found from the "**Resolved Addresses**" section under the "**Statistics**" Tab. There were many connection requests to **google ad services** and to **amazon web services**. Some of the IPs were:-
d3njjcbhbojbot.cloudfront.net -> 13.33.169.73, 13.33.169.106, 13.33.169.25
clients.l.google.com -> 172.217.160.142
ib.sin1.geoadnxs.com -> 103.43.90.180, 103.43.90.19, 103.43.90.20
dual-a-0001.a-msedge.net -> 204.79.197.200
www-cloudfront-alias.coursera.org -> 13.33.169.70, 13.33.169.21, 13.33.169.115
Amazon web services :- tracking-1659963975.ap-southeast-1.elb.amazonaws.com -> 52.221.30.100
Collector-low-pri-kafka-1101170048.us-east-1.elb.amazonaws.com -> 34.236.102.49
Google ad services :- www-googletagmanager.l.google.com -> 172.217.163.40

Reasons for Multiple IP:-

- Since modern day websites deploy **load balancing** on servers, data is sent to clients from various IP in most efficient manner.
- **Round Robin DNS mechanism** for faster fetching of relevant pages by balancing the page requests across many servers may lead to multiple IP.
- It may be possible that video is being streamed from one IP and other site data is being loaded from another server with different IP. Since video and html data might be present on different servers and hence different IP.
- Multiple servers help in **reducing network congestion** and **increase reliability**, as there is **no single point of error**. If one server is unable to provide the resource, another server can step in.
