NETWORKS LAB

ASSIGNMENT – 2

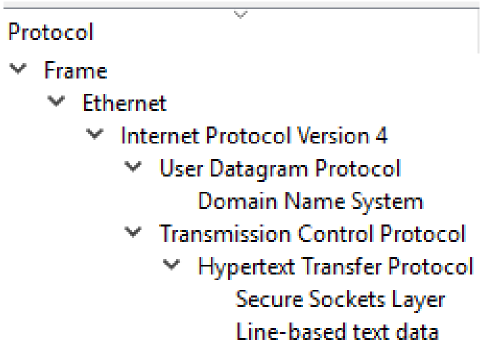
Anirudh Sharma

150101006

I traced the following NPTEL course for the Assignment: http://nptel.ac.in/courses/106106126/ and the traces are in the link given below

https://drive.google.com/open?id=1Ll_l9A1s-J0zR6rjtJZfz9WVMN3SHAE-

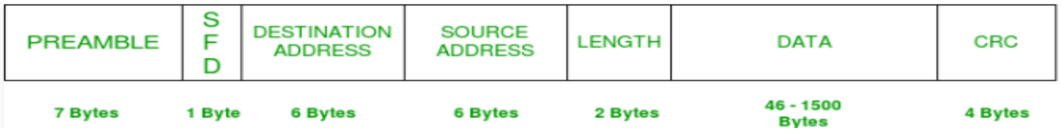
ANS 1 Protocol hierarchy of the protocol used by NPTEL.



```
Protocol
⌄ Frame
  ⌄ Ethernet
    ⌄ Internet Protocol Version 4
      ⌄ User Datagram Protocol
          Domain Name System
      ⌄ Transmission Control Protocol
        ⌄ Hypertext Transfer Protocol
            Secure Sockets Layer
            Line-based text data
```

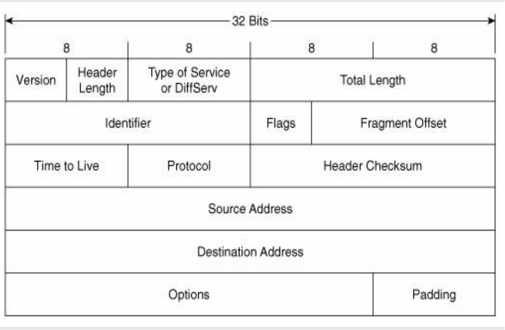
DATA LINK-LAYER (ETHERNET): Packet format:

An Ethernet frame is preceded by a preamble and start frame delimiter (SFD). An Ethernet packet contains the information about the Destination MAC address (6 bytes) and the Source MAC Address (6 bytes) . The middle section of the frame is payload data including any headers for other protocols (for example, Internet Protocol) carried in the frame. The frame ends with a frame check sequence (FCS), which is a 32-bit cyclic redundancy check used for error detection.



PREAMBLE	SFD	DESTINATION ADDRESS	SOURCE ADDRESS	LENGTH	DATA	CRC
7 Bytes	1 Byte	6 Bytes	6 Bytes	2 Bytes	46 - 1500 Bytes	4 Bytes

NETWORK LAYER, Internet Protocol version 4(IPv4): IPv4 frame has relevant information like version number (which is 4 here , Protocol , source(32 bits) and destination address(32 bits) . If IP packet is fragmented then all fragments will have same identification. Header checksum is for error correction. Options includes different option of IPv4 packets like Security, record Route, time-Stamp.

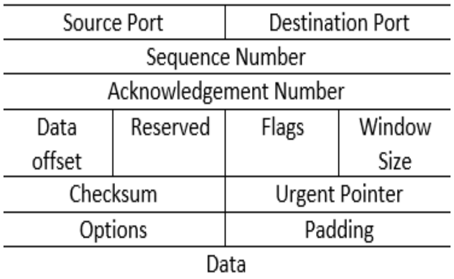


32 Bits

8		8	8	8
Version	Header Length	Type of Service or DiffServ	Total Length	
Identifier			Flags	Fragment Offset
Time to Live		Protocol	Header Checksum	
Source Address				
Destination Address				
Options				Padding

IPv4 Header

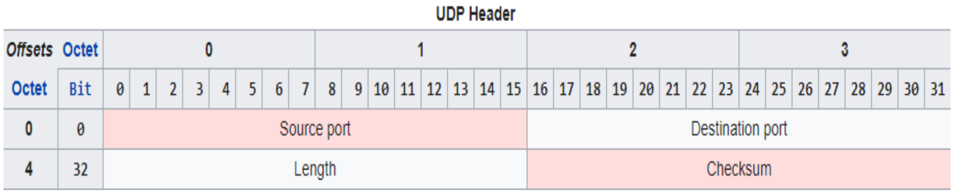
TRANSPORT LAYER, Transmission control protocol (TCP):



Source Port		Destination Port	
Sequence Number			
Acknowledgement Number			
Data offset	Reserved	Flags	Window Size
Checksum		Urgent Pointer	
Options		Padding	
Data			

On Left is the format of TCP header. Source and destination port identifies the end points of the connection. Sequence number specifies the number assigned to the first byte of data in the current message. Data offset tells how many 32 bits words are there in TCP packet. Acknowledgement is the sequence number of next frame which the source is expecting to receive. Checksum is used for error detection.

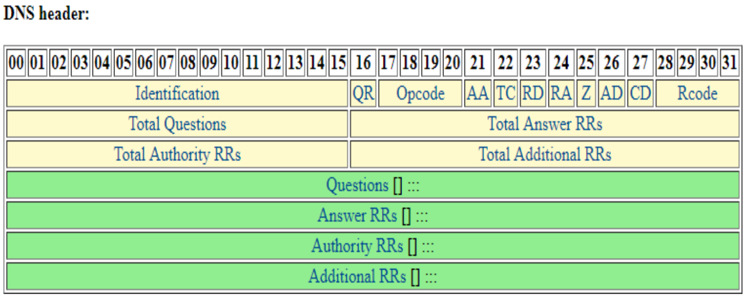
TRANSPORT LAYER, User datagram Protocol (UDP): The UDP header consists of 4 fields, each of which is 2 bytes (16 bits) - Source port, destination port, checksum and length. Function of different part remain same as of TCP.



UDP Header

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Length																Checksum															

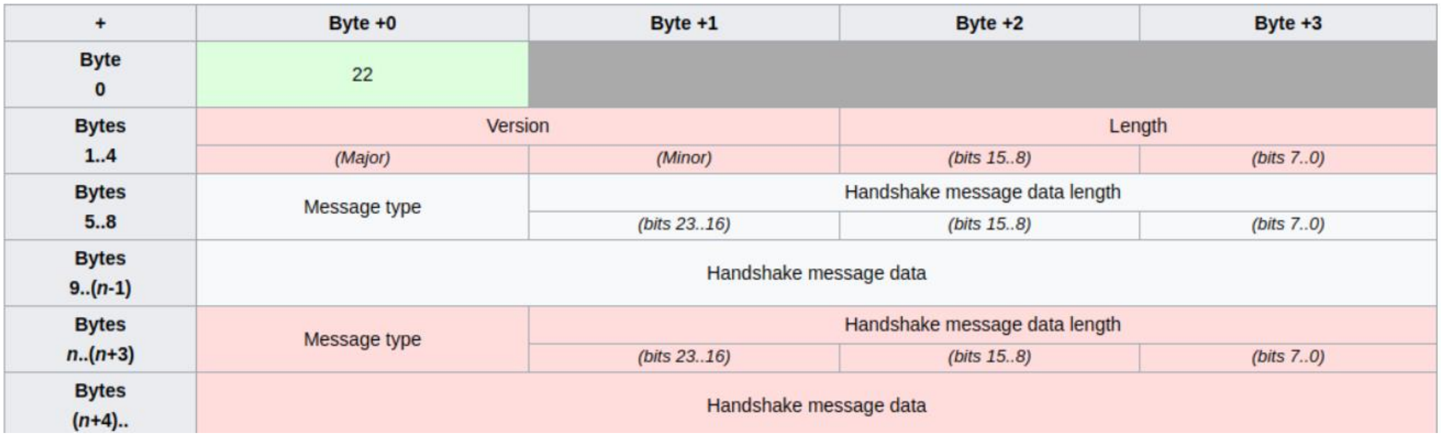
Application layer, Domain name system (DNS): It translates more readily memorized domain names to the numerical IP addresses. Format of DNS header is shown in right. Identification is used to match request replies. QR is query/response either 0 or 1. Total Questions (16 bits) (unsigned) Number of entries in the question list that were returned . Total answer return from answers resource record list returned. Query a list of 0's or more queries. Answer RR, Authority RR and Additional RR a list of 0's or Answer record list structure, Authority record list structure and Additional record list structure respectively.



DNS header:

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Identification																QR	Opcode				AA	TC	RD	RA	Z	AD	CD	Rcode			
Total Questions																Total Answer RRs															
Total Authority RRs																Total Additional RRs															
Questions [] :::																															
Answer RRs [] :::																															
Authority RRs [] :::																															
Additional RRs [] :::																															

Application Layer , TLS / SSL :



+	Byte +0	Byte +1	Byte +2	Byte +3
Byte 0	22			
Bytes 1..4	Version		Length	
	(Major)	*(Minor)*	*(bits 15..8)*	*(bits 7..0)*
Bytes 5..8	Message type	Handshake message data length		
		(bits 23..16)	*(bits 15..8)*	*(bits 7..0)*
Bytes 9..(*n*-1)	Handshake message data			
Bytes *n*..(*n*+3)	Message type	Handshake message data length		
		(bits 23..16)	*(bits 15..8)*	*(bits 7..0)*
Bytes (*n*+4)..	Handshake message data			

The TLS protocol comprises two layers: the **TLS record protocol** and the **TLS handshake protocol**. Version: Contains the version of TLS we are using. Length: contains the length of the total TLS packets received.

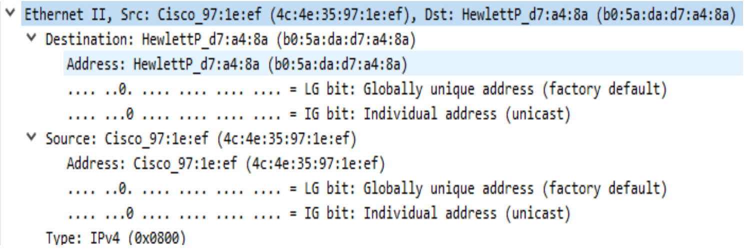
Message Type: there could be four types of TLS packets application layer, handshake protocol, change cipher spec protocol and alert protocol. Message data length: contains the length of the message. Handshake message data is the encrypted data.

Application layer , HTTP : A typical HTTP request contains many fields like Accept (to specify media types), accept-charset, accept-encoding and accept-language. It also contains fields for proxy authorization, range, host and User- agent. An 'expires' field is also present which specifies the time afterwhich the response is considered stale **.**

__ANS 2__. Values obtained for different protocol in wireshark are:

ETHERNET II: Destination: b0:5a:da:d7:a4:8a Mac address of my PC .

Source: Cisco_97:le:ef MAC address of source .



```
v Ethernet II, Src: Cisco_97:1e:ef (4c:4e:35:97:1e:ef), Dst: HewlettP_d7:a4:8a (b0:5a:da:d7:a4:8a)
  v Destination: HewlettP_d7:a4:8a (b0:5a:da:d7:a4:8a)
      Address: HewlettP_d7:a4:8a (b0:5a:da:d7:a4:8a)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  v Source: Cisco_97:1e:ef (4c:4e:35:97:1e:ef)
      Address: Cisco_97:1e:ef (4c:4e:35:97:1e:ef)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
```

Type (16 bits): Indicate what upper layer protocol should be used.

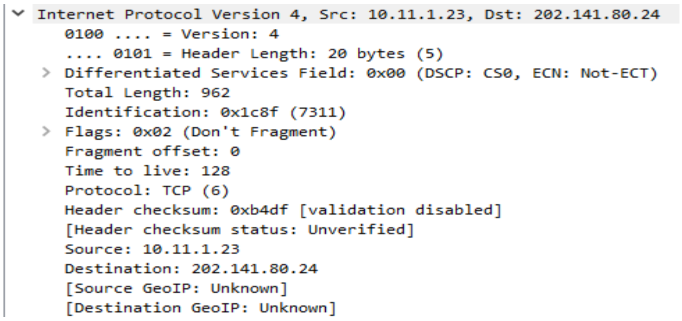
NETWORK LAYER (IPv4): Version: 4bits (0100)

Differentiated services field: this determines the level of service requested for the packet, like high priority or best effort delivery.

Total length: header length + Packet length (962 bytes)

Flags: tells if IP packet can be fragmented or not, here not fragment.

Fragment offset: position of fragmentation of IP packet, 0 if not fragment is selected.



```
v Internet Protocol Version 4, Src: 10.11.1.23, Dst: 202.141.80.24
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 962
    Identification: 0x1c8f (7311)
  > Flags: 0x02 (Don't Fragment)
    Fragment offset: 0
    Time to live: 128
    Protocol: TCP (6)
    Header checksum: 0xb4df [validation disabled]
    [Header checksum status: Unverified]
    Source: 10.11.1.23
    Destination: 202.141.80.24
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
```

Time to live: remaining hops packet can do before reaching destination here 128.

Protocol: which Upper protocol does this packet belong, here TCP.

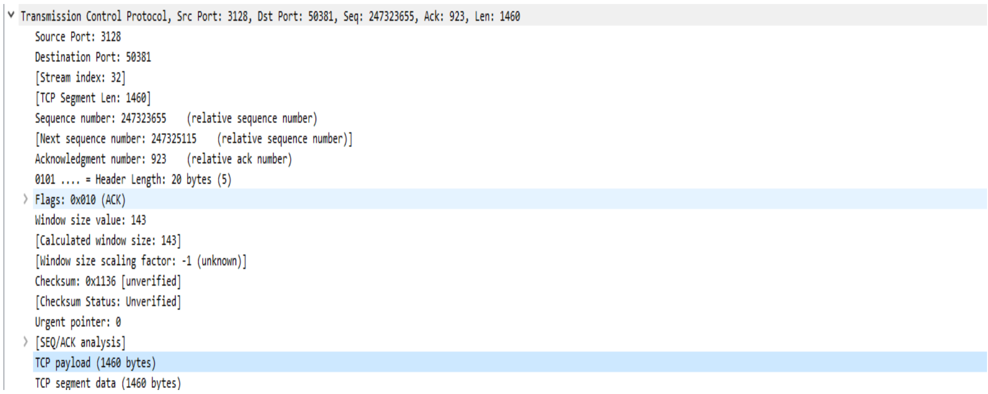
Header checksum: for error detection usually CRC is used. Here validation is disables of checksum is not verified. Source: IP address from which packet is send (Umiam hostel IP 10.11.23), Destination: IITG proxy server, 202.141.80.24

TCP: Source port: 3128 which is proxy port

Destination port: port of receiver, here random created by my PC.

Sequence Number: data is divided while sending in IP packets. This gives relative numbering of packets.

Acknowledgement Number : To acknowledge the packet which is received .



```
v Transmission Control Protocol, Src Port: 3128, Dst Port: 50381, Seq: 247323655, Ack: 923, Len: 1460
    Source Port: 3128
    Destination Port: 50381
    [Stream index: 32]
    [TCP Segment Len: 1460]
    Sequence number: 247323655    (relative sequence number)
    [Next sequence number: 247325115    (relative sequence number)]
    Acknowledgment number: 923    (relative ack number)
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x010 (ACK)
    Window size value: 143
    [Calculated window size: 143]
    [Window size scaling factor: -1 (unknown)]
    Checksum: 0x1136 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  > [SEQ/ACK analysis]
    TCP payload (1460 bytes)
    TCP segment data (1460 bytes)
```

Header length : length TCP header(20 bytes)

Flags : tells which type pf packet it is currently Acknowledgement .

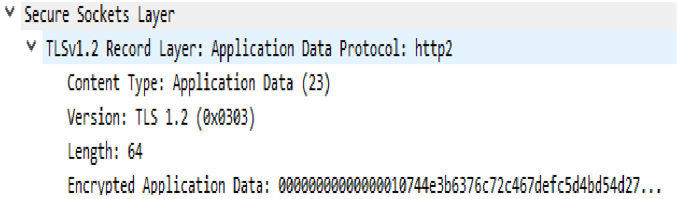
Window size : indicates buffer space for reciveing packets .

Urgent pointer : if some packet has to reach server as soon as possible .   TCP payload : size of data (1460 bytes)**.**

SSL Protocol : content type -> there are four type of protocol application layer , handshake protocol , change cipher spec protocol and alert protocol , here we have Application protocol .

Version : version of TLS we are using that is 1.2 .

Length: length of protocol message , mac and padding which is 64.



```
Secure Sockets Layer
  TLSv1.2 Record Layer: Application Data Protocol: http2
    Content Type: Application Data (23)
    Version: TLS 1.2 (0x0303)
    Length: 64
    Encrypted Application Data: 0000000000000001 0744e3b6376c72c467defc5d4bd54d27...
```

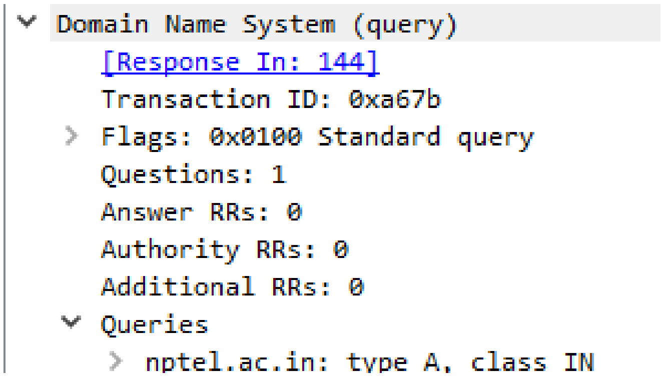
Data received is also encrypted to increase security .

DNS : Transaction ID : a 16 bit identification created by the device that creates the DNS quesry . here it is 0xa67b .

Flags: Various type of flags to indicate whether it is a query, response ,opcode .

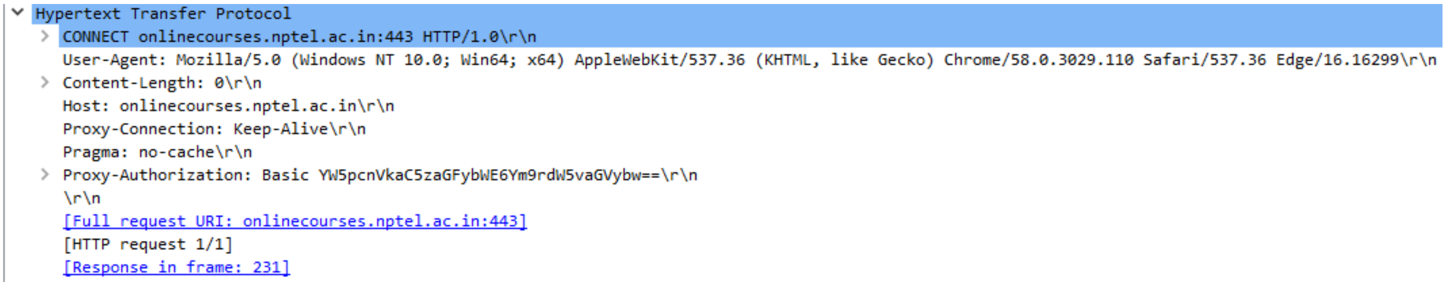
Questions /Answer/Authority RR's/Additional RR's: number of queries / answer / authority response / additional response . this is a query hence number of Questions is 1 while others are 0 .

Queries /Answers : whatever query or answer sent . Here we are asking IP for nptel.ac.in therefore it is in query .



```
Domain Name System (query)
    [Response In: 144]
    Transaction ID: 0xa67b
  > Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
    Queries
      > nptel.ac.in: type A, class IN
```

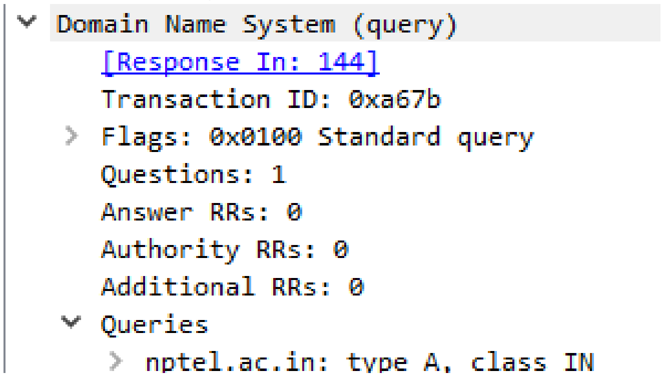
HTTP :



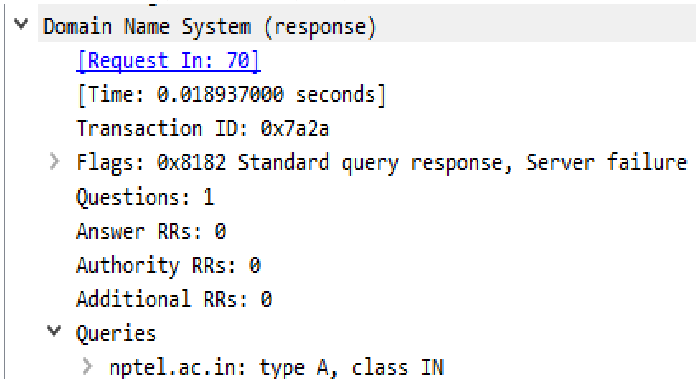
```
Hypertext Transfer Protocol
  > CONNECT onlinecourses.nptel.ac.in:443 HTTP/1.0\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.36 Edge/16.16299\r\n
  > Content-Length: 0\r\n
    Host: onlinecourses.nptel.ac.in\r\n
    Proxy-Connection: Keep-Alive\r\n
    Pragma: no-cache\r\n
  > Proxy-Authorization: Basic YW5pcnVkaC5zaGFybWE6Ym9rdW5vaGVybw==\r\n
    \r\n
    [Full request URI: onlinecourses.nptel.ac.in:443]
    [HTTP request 1/1]
    [Response in frame: 231]
```

HTTP gives the hostname which is onlinecourses.nptel.ac.in our case. It also gives information about proxy username and authorization other details include user-agent (the application process responsible for the packet), referrer (address of the webpage), accept encoding (listing all accepted encodings), accept language (all languages accepted) and X- Requested-With (used for streaming).

ANS 3. Right after entering: http://nptel.ac.in/courses/106106126/ browser sends a **DNS query** to the DNS server asking for the IP of the domain name entered. The DNS server reply to the query as shown:

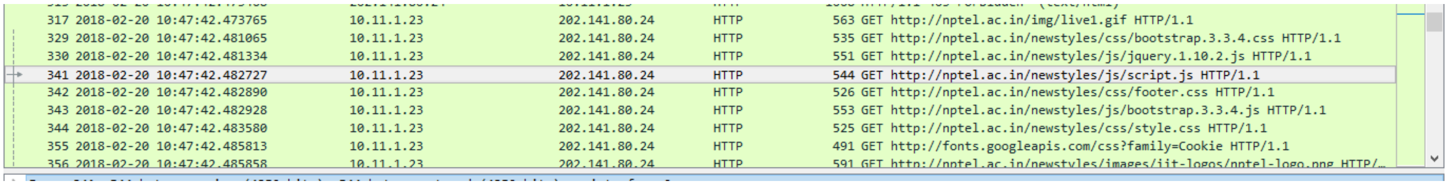


```
Domain Name System (query)
    [Response In: 144]
    Transaction ID: 0xa67b
  > Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
    Queries
      > nptel.ac.in: type A, class IN
```



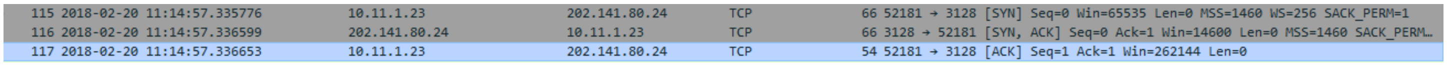
```
Domain Name System (response)
    [Request In: 70]
    [Time: 0.018937000 seconds]
    Transaction ID: 0x7a2a
  > Flags: 0x8182 Standard query response, Server failure
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
    Queries
      > nptel.ac.in: type A, class IN
```

HTTP request: many http GET commands were send to load the page stylesheet, script and contents, for example :



```
317 2018-02-20 10:47:42.473765    10.11.1.23     202.141.80.24   HTTP   563 GET http://nptel.ac.in/img/live1.gif HTTP/1.1
329 2018-02-20 10:47:42.481065    10.11.1.23     202.141.80.24   HTTP   535 GET http://nptel.ac.in/newstyles/css/bootstrap.3.3.4.css HTTP/1.1
330 2018-02-20 10:47:42.481334    10.11.1.23     202.141.80.24   HTTP   551 GET http://nptel.ac.in/newstyles/js/jquery.1.10.2.js HTTP/1.1
341 2018-02-20 10:47:42.482727    10.11.1.23     202.141.80.24   HTTP   544 GET http://nptel.ac.in/newstyles/js/script.js HTTP/1.1
342 2018-02-20 10:47:42.482890    10.11.1.23     202.141.80.24   HTTP   526 GET http://nptel.ac.in/newstyles/css/footer.css HTTP/1.1
343 2018-02-20 10:47:42.482928    10.11.1.23     202.141.80.24   HTTP   553 GET http://nptel.ac.in/newstyles/js/bootstrap.3.3.4.js HTTP/1.1
344 2018-02-20 10:47:42.483580    10.11.1.23     202.141.80.24   HTTP   525 GET http://nptel.ac.in/newstyles/css/style.css HTTP/1.1
355 2018-02-20 10:47:42.485813    10.11.1.23     202.141.80.24   HTTP   491 GET http://fonts.googleapis.com/css?family=Cookie HTTP/1.1
356 2018-02-20 10:47:42.485858    10.11.1.23     202.141.80.24   HTTP   591 GET http://nptel.ac.in/newstyles/images/iit-logos/nptel-logo.png HTTP/…
```

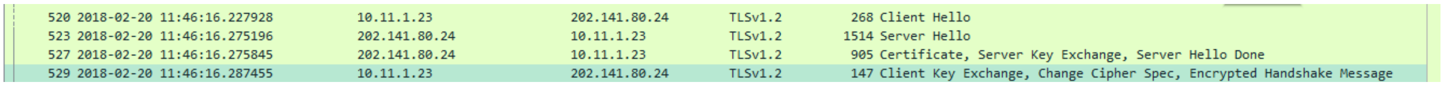
Establishment of connection: (**PLAY / PAUSE ACTION**)As soon as play button is pressed a 3 way TCP handshake takes place .



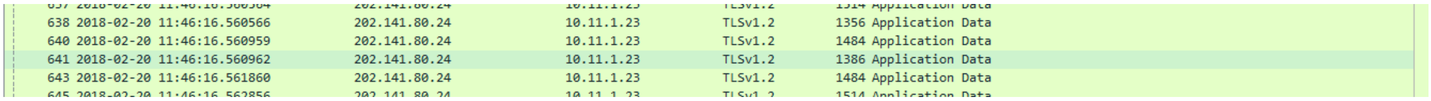
```
115 2018-02-20 11:14:57.335776    10.11.1.23       202.141.80.24   TCP   66 52181 → 3128 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
116 2018-02-20 11:14:57.336599    202.141.80.24    10.11.1.23      TCP   66 3128 → 52181 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM…
117 2018-02-20 11:14:57.336653    10.11.1.23       202.141.80.24   TCP   54 52181 → 3128 [ACK] Seq=1 Ack=1 Win=262144 Len=0
```

1. First client (10.11.1.23) send a SYN segment . This request is for the server to synchronize the sequence number.
2. Server sends an SYN and ACK segment . Server is acknowledging client of its sequence number and at the same time server is also sending its sequence number for client to synchronize .
3. Finally client is sending an ACK to acknowledge server request to synchronize the segment number . hence a reliable connection is set .

After this, the TCP protocol uses the positive acknowledgment technique to receive the video content. As the data packet is received by the client host, it sends an acknowledgement with the sequence number of next packet which it is expecting. If a particular packet is lost, the acknowledgement is sent again with the same sequence number. The client can also acknowledge for multiple packets together .TLSv1.2 uses handshake to establish connection. Once clients have exchanged keys and encrypted handshake message is done video transmission can take place .



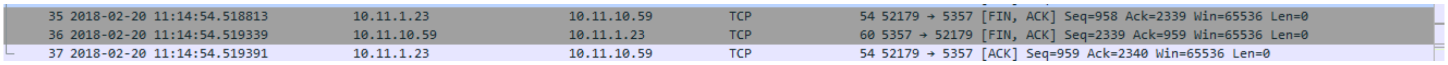
```
520 2018-02-20 11:46:16.227928    10.11.1.23       202.141.80.24   TLSv1.2    268 Client Hello
523 2018-02-20 11:46:16.275196    202.141.80.24    10.11.1.23      TLSv1.2   1514 Server Hello
527 2018-02-20 11:46:16.275845    202.141.80.24    10.11.1.23      TLSv1.2    905 Certificate, Server Key Exchange, Server Hello Done
529 2018-02-20 11:46:16.287455    10.11.1.23       202.141.80.24   TLSv1.2    147 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
```



```
638 2018-02-20 11:46:16.560566    202.141.80.24    10.11.1.23      TLSv1.2   1356 Application Data
640 2018-02-20 11:46:16.560959    202.141.80.24    10.11.1.23      TLSv1.2   1484 Application Data
641 2018-02-20 11:46:16.560962    202.141.80.24    10.11.1.23      TLSv1.2   1386 Application Data
643 2018-02-20 11:46:16.561860    202.141.80.24    10.11.1.23      TLSv1.2   1484 Application Data
645 2018-02-20 11:46:16.562856    202.141.80.24    10.11.1.23      TLSv1.2   1514 Application Data
```

PAUSE action : Pause action has no effect as the TLS packets keep on coming till the whole video is not transmitted . Pause only stops the video on the user side but has no effect on delivery of packets .

Termination of the Connection :



```
35 2018-02-20 11:14:54.518813    10.11.1.23     10.11.10.59    TCP   54 52179 → 5357 [FIN, ACK] Seq=958 Ack=2339 Win=65536 Len=0
36 2018-02-20 11:14:54.519339    10.11.10.59    10.11.1.23     TCP   60 5357 → 52179 [FIN, ACK] Seq=2339 Ack=959 Win=65536 Len=0
37 2018-02-20 11:14:54.519391    10.11.1.23     10.11.10.59    TCP   54 52179 → 5357 [ACK] Seq=959 Ack=2340 Win=65536 Len=0
```

1. The client send a FIN which indicates it has no more data to send, while ACK identifies the connection between them .
2. Server acknowledge the FIN of client by ACK and send its own FIN .
3. Client acknowledges server FIN , and hence the server is relaxed .

Since entire communication is over TLS , we have no way of distinguishing the data since it is encrypted .

ANS 4. Importance and function of different layer in NPTEL video transmission are:

Hypertext transfer protocol (HTTP) : HTTP is an application layer protocol which uses a request-response protocol in the client-server model . It is used to get various data from server like HTML documents .In NPTEL it is basically used for transfer of HTML, style-sheets , scripts and images of the NPTEL website .

Transmission control protocol (TCP): From the traces we would clearly see that NPTEL uses 3-way handshaking for establishing connection and four message to finish connection. TCP provides host to host connectivity and control. If the packet is lost, it requests for retransmission. It also uses the positive acknowledgement technique to guarantee reliability of packet transfers.

INTERNET PROTOCOL (IPv4) : Its role is to transfer packet from source to destination on the basis of IP address of source and destination . This protocol is responsible for transmission of packets to travel from NPTEL servers to IITG server than from IITG server to our PC .

TLS/SSL : NPTEL uses TLS/SSL protocol in application layer for transfer of video from source to client . TLS/SSL is an application layer protocol which encrypted transfer of data. This provides confidentiality – that is no one on the network could see what we are watching or can alter the video. Once HTML page is loaded rest of video data is send via TLS/SSL packets. TLS also does encypted handshake and client key exchange .

ANS 5 . Statistics obtained are as follows:

Time	Trace1 11:45 AM	Trace2 10:24 PM	Trace 3 9:00 AM	Trace 4 4:10 PM
Client IP (A)	10.11.1.23	10.11.1.23	10.11.1.23	172.16.114.137
Host IP (B)	202.141.80.24	202.141.80.24	202.141.80.24	202.141.80.24
Throughput(bytes/s)	1369	6597	21000	11000
RTT (in ms) = 1/(average pps)	588.2 ms	151.5 ms	41.8 ms	75.18 ms
Packet Size (B)	813.5	1006.5	910.5	891.5
No. of Packet loss%	0%	0%	0%	0%
No. of UDP packets DNS server (D): 202.141.81.2	A->D = 0 D->A = 0	A->D = 1 D->A= 1	A-> D = 0 D ->A = 0	A -> D =7 D -> A = 6
No. of TCP packets	A->B = 53 B->A = 24	A->B = 10 B->A= 11	A->B = 42 B->A = 19	A -> B = 12 B -> A =6
No. of response/request sent	24/53 = .452	11/10 = 1.1	19/42 = .452	6/12 = .5

ANS 6 . The IP was found the same due to the reason that we are using IITG proxy server . Hence all IP will be 202.141.80.24 . But http://nptel.ac.in/ has an IP 14.139.160.71 (checked online) while http://nptel.ac.in/courses/106106126/ has an IP address 14.139.160.69 which is a different server .

The reason for multiple sources to balance the page request and serve the end user faster . The website may also use a CDN to host static content .A CDN is a system of distributed networks that deliver webpages and other Web content to a user based on the geographic locations of the user, the origin of the webpage and a content delivery server. This helps in speeding the delivery of content with traffic rates.