CS-349 NETWORKS ASSIGNMENT-4

-MAYANK WADHWANI (170101038) ABHISHEK JAISWAL (170101002) ANKUR INGALE (170122130)

#### TOPIC ASSIGNED: VIDEO STREAMING ON HOTSTAR.COM

#### TRACES: bit.ly/397rhsc

# QUESTION-1 PROTOCOLS USED IN DIFFERENT LAYERS AND THEIR PACKET FORMATS

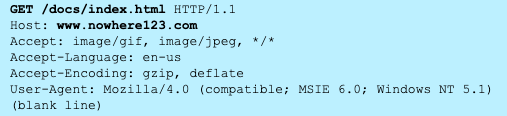
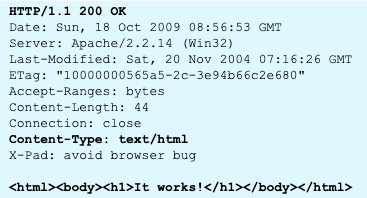
1. **APPLICATION LAYER:**
2. **HTTP**: HTTP stands for **HyperText** **Transfer** **Protocol** and it is used in the application layer. There are two types of messages that are sent between the web server and the web client: namely **HTTP request** and **HTTP** **response**. An HTTP request consists of the request line which specifies what type of request is asked for(it can be GET, POST, PUT, DELETE or HEAD) and the HTTP version and the headers which contain additional information like language accepted. It may also contain an HTTP Body to send additional information. The web server responds to the HTTP request by generating an HTTP response. HTTP response comprises of an initial status line(like 404 means not found), header lines and an entity body.

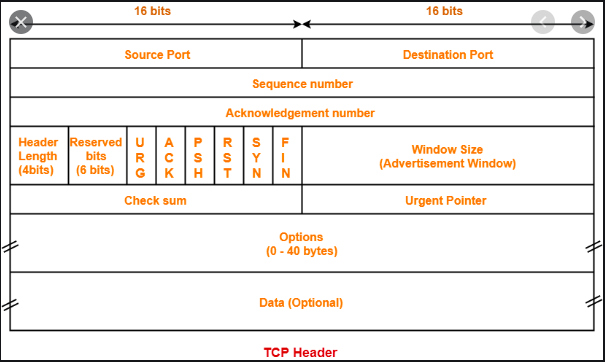
Figure 2 An HTTP request

Figure 1 An HTTP response

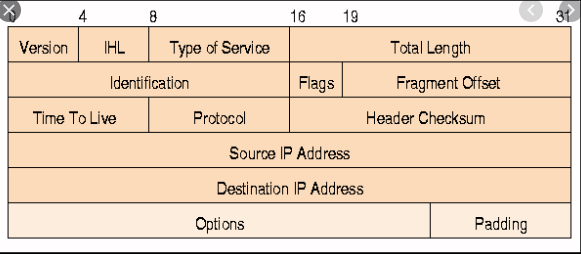
1. **SESSION LAYER**

**TLS**: The TLS layer sits between the Application layer and the transport layer. It provides security in transmission by encrypting the data to be sent. The basic unit of SSL is a **record**. Each record consists of a five-byte record header, followed by data. The record format is Type ( Handshake, Application Data, Alert and Change Cipher Spec), Version and Length. It can be used by any protocol that used TCP as the transport layer

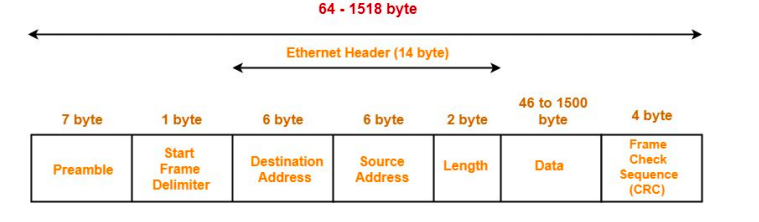
1. **TRANSPORT LAYER**

**TCP**: The **Transmission** **Control** **Protocol** is a standard that defines how to establish and maintain a network connection between application programs that wish to send data over the network. The TCP packet header consists of the following fields: **Source Port** (16bits) and **Destination Port** (16 bits) to correctly identify the address of the communicating hosts. **Sequence Number** and **Acknowledgement** **Number** (32 bits each) are used to maintain the ordering of the packets being transferred. **Header Length** (4 Bits) specifies the total size of the TCP header in multiples of 4 bytes. **Reserved bits** (3 bits) serves the purpose of aligning the total header size so that it becomes a multiple of 4. **Window size** (32 bits) is used to regulate the amount of data to be sent to a receiver before requiring an acknowledgement in return. The **Check Sum** (2 bits) is used for error detection. The **Urgent** **Pointer** field is used to point to data that is urgently required and is often set to zero and ignored. **Data** (variable length) contains upper layer information.

1. **NETWORKS LAYER**

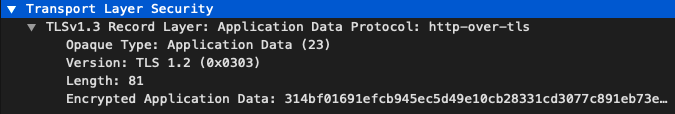
**IPV4** is a connectionless protocol or use on packet switched networks. An IP packet can contain up to 60 bytes of IP header data. The following are the fields one observes in an IP packet header. The **Version** (4 bits) specifies the version no of the internet protocol used (4 in this case). The **Internet Header Length (IHL**)(4 bits) represents the length of the header. The **Type of service** field itself contains 2 fields viz. the **Differentiated Services Code Point(DSCP)** (6 bits) which is used to define differentiated services like VoIP (Voice over IP) and **Explicit Congestion Notification(ECN)** (2 bits) allows end-to-end notification of network congestion without dropping any packets. **Total length (**16 bits) field defines the entire length of the IP packet. The **Identification** (16 bits) field is used to uniquely identify group of fragments for a single IP datagram. The **Fragment Offset (**13 bits) is used to tell the exact position of the fragment in the original IP Packet. The **TTL or Time To Live (**8 bits) tells how many routers at maximum the packet can hop through. So at each router, its value is decremented by 1 and as a result of which if we exceed to max routers allowed the value becomes 0 and the packet is destroyed. The **Protocol (**8 bits**)** specifies the protocol used in the next layer. The **Header Checksum(**16 bits) is used for error detection just as in the TCP packet. The **Source and Destination IP Address (**16 bits each) specifies the IP address of the client and the server in our case i.e. the source and the destination respectively. The **Options** field is used for additional information.

1. **LINK LAYER**

**Ethernet II** is used at the link layer. A data unit on an Ethernet Link transports an Ethernet frame as its payload. An Ethernet frame consists of the following: A **Preamble** and a **Start Frame Delimiter(SFD**) to mark the starting of the frame. Then **Destination** **Address** and **Source** **Address** follows which specified the MAC Addresses of the client and the host in our case which is the source and the destination respectively. The **Data** also calledas the **Payload**  is inserted next. In the end comes the **Frame Check Sequence (FCS)** which is used to detect errors, if any using the **Cyclic Redundancy Check (CRC).**

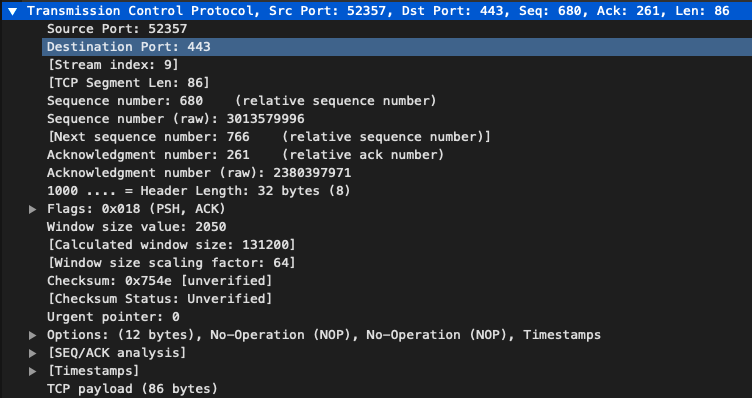
# QUESTION-2 OBSERVED VALUES

The protocols whose values were observed and listed below are **TLS, TCP** and **Ethernet II.**

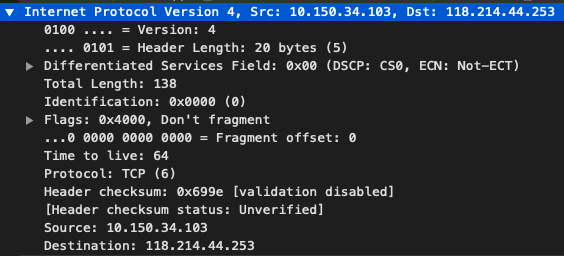
1. **TLS**:

|  |  |  |
| --- | --- | --- |
| Opaque Type | Application Data | Refers to the type of the content being transferred. 23 is an identifier for the same and it means that Application Data will be transferred |
| Version | TLS 1.2 | Refers to the version of the protocol that is used |
| Length | 81 | The Length of the application data being transferred, excluding the protocol headers and including the MAC and padding trailers |
| Encrypted Application Data | 314bf0169… | The encrypted form of the data that is sent over the network to provide security. |
| Application Data Protocol | http-over-tls | The protocol ensures secure communication over the network by encrypting the data and sending it through http requests. |

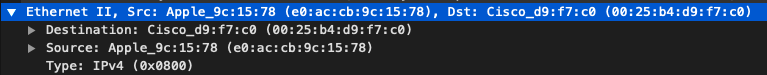
1. **TCP**

****

|  |  |  |
| --- | --- | --- |
| Source Port | 52357 | The source port number is used by the sending host to help keep track of new incoming connections and existing data streams. |
| Destination Port | 443 | Similar to the source port, the destination port is used by the receiver to keep track of new incoming connections. |
| Sequence Number | 680 | The number assigned to the packet relative to the advent of the TCP connection. So we can say that 679 packets have been sent since the TCP connection has been set up. |
| Acknowledgement Number | 261 | The acknowledgement number is the sequence number of the next byte the receiver expects to receive |
| Checksum | 0x754e | The checksum field is used for error detection. |
| Flags | PSH,ACK | PSH or the PUSH flag is an option provided by TCP that allows the sending application to start sending the data even when the buffer is not full. The ACK which stands for “Acknowledgment”, is used to acknowledge the successful receipt of a packet |
| Urgent Pointer | 0 | The Urgent Pointer field is used to point to data that is urgently required. Here there is no such requirement and so its value is set to 0. |
| Header Length | 8 | Header length is 32 bytes so the value 8 is stored as the counting happens in multiples of 4 bytes. |

1. **IPv4**

|  |  |  |
| --- | --- | --- |
| Version | 4 | The version of internet protocol used |
| Header Length | 5 | Header length is 20 bytes so the value is 5 as counting is done in multiples of 4 bytes |
| Src | 10.150.34.103 | The IP address of the sender, in this case my laptop |
| Dst | 118.214.44.253 | The IP address of the receiver, in this case the hotstar server |
| Flags | Don’t Fragment | If this flag is set and fragmentation is required to route the packet, then the packet is simply dropped. |
| TTL | 64 | This represents the maximum hops the packet can make through the routers. So after 64 hops the packet will be dropped. |
| DSCP | CS0 | This means standard service is applied in the network and this will have undifferentiated applications. |
| ECN | Not-ECT | This means not-ECN capable transport i.e. ECN is not used in the connection. |

1. **Ethernet II**

|  |  |  |
| --- | --- | --- |
| Destination | 00:25:b4:d9:f7:c0 | Refers to the MAC address of the destination server, in this case the server of hotstar.com |
| Source | E0:ac:cb:9c:15:78 | Refers to the MAC address of the source, in this case my laptop |
| Type | IPv4 | Means that the upper layer protocol used is IPv4 |

# Question-3 : Protocols helping in various functionalities

**TLS:**

1. TLS is used for prevention of unwanted **eavesdropping** and modification on internet traffic.
2. The protocol helps to provide security to the website from **external hackers** as now the data is encrypted so even if someone breaks into the website and gets access to the data, he/she will not be able to decipher it.
3. So whenever we are entering, let’s say login credentials on a website, in this case I entered my credentials on hotstar.com, the password and username are safely transmitted from my machine to the server by using TLS.

**TCP:**

1. It was the only used protocol in this case i.e. hotstar.com only uses TCP for transmission of data and not UDP. There are several reasons for the same.
2. It provides **reliability** of data i.e. if we send a packet using TCP, then we can be absolutely sure that the packet will reach the destination safely without any loss. UDP does not guarantee this, it is not as reliable as TCP.
3. It works by creating a TCP connection. Also, the TLS protocol discussed above works only for the applications that send their data through TCP. So if we are using UDP for transmission then we will not get **security** which is a major disadvantage. So TCP is both secure and reliable.
4. Since the user should be given the facility to go back and watch that part of the clip, this means that the application can’t afford to lose any information. This can be ensured by TCP as there is a process of **hand shaking** done at the start. As a result of which, the error rates are minimized.

**IPv4:**

1. **IPv4** is a connection less protocol used for packet-switched networks like the Internet.
2. It delivers packets using IP headers from source to the destination.
3. This protocol is used in the **networks** layer.

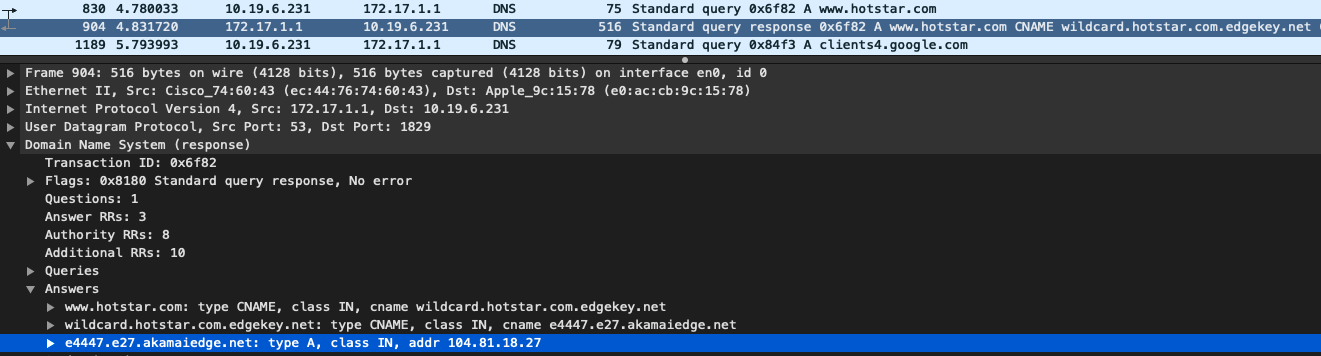
**Ethernet:**

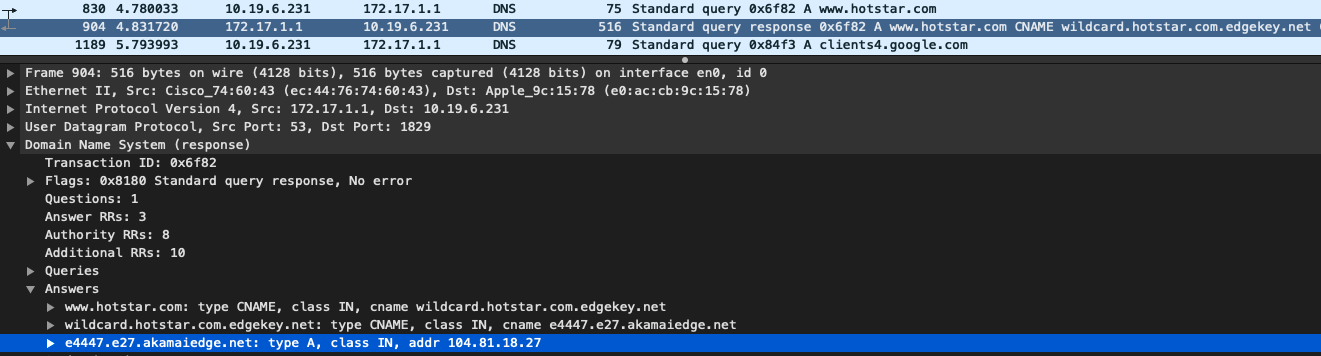
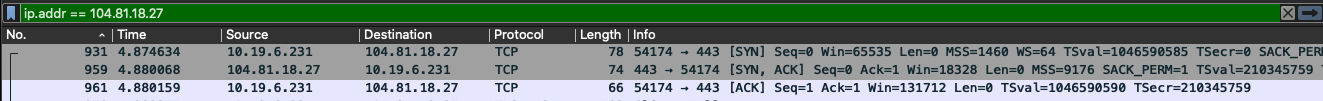
1. Ethernet is used in the **data link** layer.
2. It stores the actual physical MAC Addresses of the source and the destination hosts.
3. Lying on the data link layer, it is responsible for **error detection** and correctness. Various algorithms and techniques exist for the same including **CRC** (Cyclic Redundancy Check).

# QUESTION-4 SEQUENCE OF MESSAGES

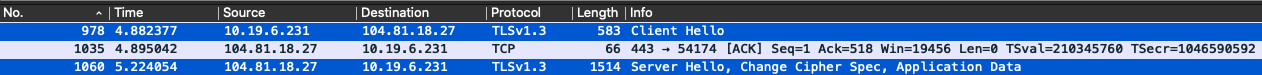
The sequence of messages were observed for various functionalities like playing a video, pausing it, closing the web browser, etc. All these have been listed below:

1. **DNS Querying**

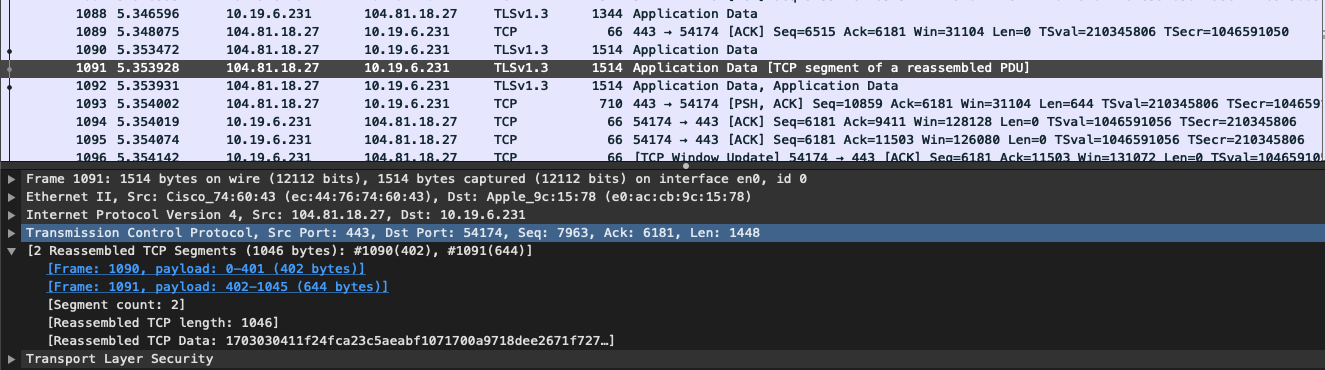
When we open the site, DNS querying is done by the browser, here Google Chrome. A series of messages are exchanged so that the browser can know the IP Address of the server ie hotstar.com. In the attached screenshot, it can be seen that a query was made to hotstar.com and the response contains the IP address of the server. **The answer part is selected(blue colour) in the screenshot**. This IP was then used as a filter to get all the messages.

1. **TCP Handshaking**

After DNS querying, we get the IP address of the server. Following this, there is a **three-way-handshake** between the source, the destination. We have already seen In the previous question that port **541474** is being used in my laptop and port **443** is being used by hotstar.com. The handshaking process advents when the host sends a **SYN** packet to the destination. This SYN packet helps in **SYNchronizing** the sequence number. The destination then responds to this packet by sending an **ACK** (acknowledgement) and another SYN packet which asks the source to SYNchronize the packet number with its sequence number. In the end, the source sends a final ACK packet and with this, we say that the handshaking process is successful.

1. **TLS Handshaking**

Once TCP connection is successfully established, we move on to TLS establishment. The TLS protocol send a ‘Client Hello’ message to the server. This can be seen from the first line of the screenshot where the source is my laptop and destination is server’s IP address ie. 104.81.18.27. The server responds to this message by a ‘Server Hello’ message and a Server Certificate(used primarily for authentication). The Server Hello signifies that the server is now ready to take requests from the client. We say that the TLS session is established and communication can take place successfully.

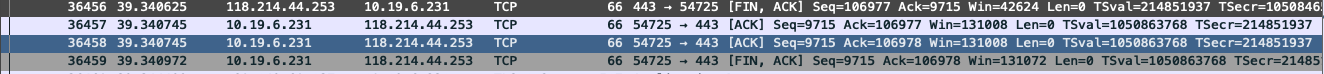
1. **Streaming Videos**

When we play the video on hotstar.com, packets are sent from the server to the client. Now these packets may take different paths to reach the client. These paths may depend on a lot of factors like load balancing. So it is possible that these packets arrive out of order. But when the packets arrive out of order, they are not directly passed to the application layer. We wait for the rest packets to arrive. So in this situation, we can see in the screenshot that the packet number 1091 and 1090 were reassembled so that the data becomes in order. Once reassembling is done, we send the data to the Application layer.

1. **Pausing A Video**

When we pause the video, some data still comes at the client end till the receiving buffer is

full. Once it is full, the client sends a FIN piggybacked by an ACK to the server to stop the packet flow and the server acknowledges. But since the user has only paused the video and can play it anytime in the future, the client keeps sending Keep-Alive messages to the server notifying it to not close the connection. Once the user presses the play button again, the client sends a SYN packet to notify the server to start sending the packets again. In this manner the pause functionality works and the TCP connection stays established.

1. **Closing the website**

When the user closes the website, a 4-way termination handshake takes place. First the client sends a FIN packey piggybacked by an ACK packet. The server responds to this by sending an ACK packet from its end. It also sends a FIN packet to the client. The client acknowledges it by sending the final ACK packet to the server after which the connection successfully closes.

# QUESTION-5 STATISTICS FROM TRACES

Following are the required statistics:

|  |  |  |  |
| --- | --- | --- | --- |
| Property | **Time 1 : 5:00 AM (CSE LAB)** | **Time 2: 2:30 PM (Lohit Hostel)** | **Time 3: 7:00 PM (Library)** |
| Avg. Throughput (in Kilo Bytes per second) | 254 | 212 | 593 |
| RTT | 0.00059272 | 0.0010439 | 0.0003713 |
| Avg. Packet Size (in Bytes) | 757 | 771 | 592 |
| Number of packets lost | 0 | 0 | 0 |
| Number of UDP packets | 142 | 1488 | 5301 |
| Number of TCP packets | 34343 | 17666 | 76188 |
| Number of responses per request | 23644/10699 (2.2099) | 9439/8227 (1.1473) | 47647/28541 (1.6694) |

The UDP packets seen in the table can be attributed mainly to DNS and other activities performed by the browser. However, hotstar.com does not use UDP for transmission of packets.

# QUESTION-6 CONTENT SOURCES

**For the Source:**

It was observed that there was a change in source IP while sending the data. When the messages were sent in the morning from the lab, the observed IP address was 10.150.33.220. Whereas in the afternoon from my hostel (Lohit), observed IP was 10.19.6.231 and lastly observed IP address from the library in the evening was 10.150.34.103. For all these cases, I used IIT\_CONNECT for internet connection. The main reason for difference in these IP’s can be because at different locations, different routers have been set up. These routers have different IP’s. So when I connect my laptop to these different routers, I observed different IP addresses for the sources. We can conclude that source IP depends on the **location** of the laptop.

**For the Destination:**

For this part, it was observed that the destination IP address changed at different times of the day. In the morning, the destination address was 104.81.18.27, whereas in the evening the observed IP was 180.149.60.168. Finally, in the afternoon, observed IP was 118.214.44.253. We observe different values because of:

1. **Load** **balancing**: Most of the websites have several servers set up across the world. This helps in **load balancing** i.e. if one particular server receives a lot of requests at one time, the next request is sent to some other server by the router. This helps to keep the network traffic stable.
2. **Reliability**: If due to any unforeseen reasons, the server fails, then there should be other servers which respond to the clients. If however we are using only one server, then the website crashes if that server goes down. So using multiple servers helps to provide **reliability**.