# CSE 421

Lab 2: Introduction to Packet Analysis and Network Simulation

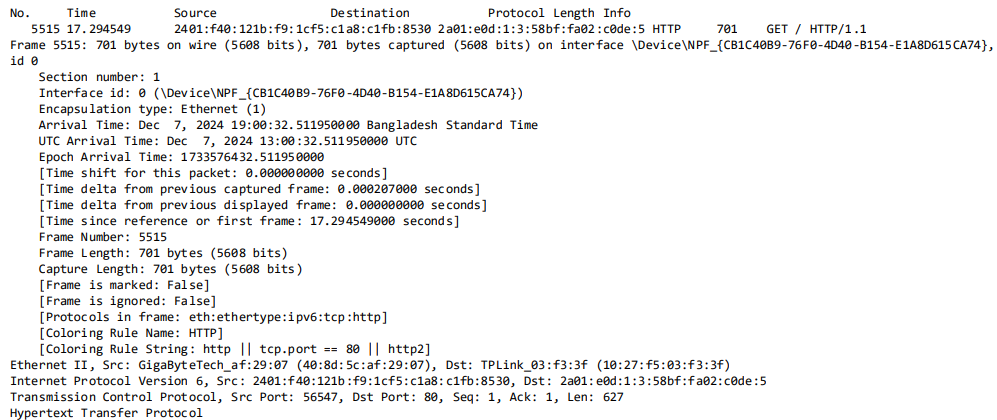
ID: 22241042

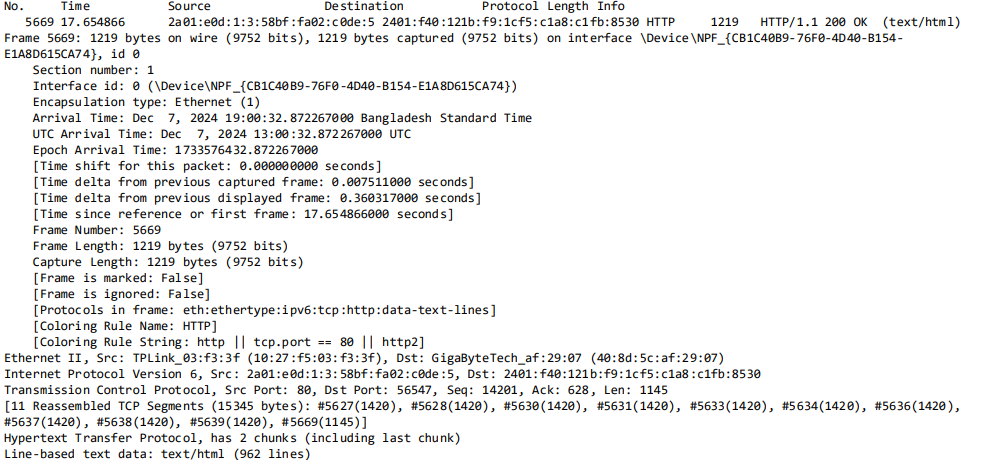
**Question:** Visit any http website while monitoring the network traffic. Now filter only the http packets. Observe the headers of any http request and response packet and explain how the layers are operating in this case.

Answer - For this task, I have chosen a website that does not have https. The website I have chosen is http://www.videolan.org/, which only has some text and images on the website. Here is the request captured from the wireshark while loading this website:

### **Packet Analysis**

**Request:**



**Response:  
**

#### **HTTP Request (GET)**

* **Packet Details:**
  + **Packet Number:** 5515
  + **Source IP:** 2401:f40:121b:f9:1cf5:c1a8:c1fb:8530 (Client)
  + **Destination IP:** 2a01:e0d:1:3:58bf:fa02:c0de:5 (Server)
  + **Source MAC:** 40:8d:5c:af:29:07 (GigaByteTech)
  + **Destination MAC:** 10:27:f5:03:f3:3f (TPLink)
  + **Length:** 701 bytes

#### **HTTP Response (200 OK)**

* **Packet Details:**
  + **Packet Number:** 5669
  + **Source IP:** 2a01:e0d:1:3:58bf:fa02:c0de:5 (Server)
  + **Destination IP:** 2401:f40:121b:f9:1cf5:c1a8:c1fb:8530 (Client)
  + **Source MAC:** 10:27:f5:03:f3:3f (TPLink)
  + **Destination MAC:** 40:8d:5c:af:29:07 (GigaByteTech)
  + **Length:** 1219 bytes

### **OSI Layer-wise Analysis**

#### **1. Application Layer (HTTP)**

* **Request:**
  + **Method:** GET
  + **Path:** /
  + **Protocol Version:** HTTP/1.1
  + The client initiates a request to retrieve the root resource of the server.
* **Response:**
  + **Status:** 200 OK
  + **Content-Type:** text/html
  + The server successfully delivers the requested content in HTML format, which is chunked into multiple segments for transmission.

#### **Application Layer Operations:**

The client sends a GET request to the server, specifying the resource to fetch. The server then responds with a 200 OK status and the content type, indicating the successful retrieval of the resource (HTML page).

#### **2. Transport Layer (TCP)**

* **Request:**
  + **Source Port:** 56547
  + **Destination Port:** 80 (HTTP default)
  + **Sequence Number:** 1
  + **Acknowledgment Number:** 1
  + **Payload Length:** 627 bytes
  + Reliable transmission is ensured through sequence and acknowledgment numbers.
* **Response:**
  + **Source Port:** 80 (Server-side)
  + **Destination Port:** 56547 (Client-side)
  + **Sequence Number:** 14201
  + **Acknowledgment Number:** 628
  + **Payload Length:** 1145 bytes

#### **Transport Layer Operations:**

The transport layer ensures reliable data delivery. Each segment includes sequence numbers to indicate the order of data and acknowledgment numbers to confirm the receipt of packets. This guarantees that data is correctly assembled and reliably transferred from the client to the server and vice versa.

#### **3. Network Layer (IPv6)**

* **Request:**
  + **Source IP:** 2401:f40:121b:f9:1cf5:c1a8:c1fb:8530 (Client)
  + **Destination IP:** 2a01:e0d:1:3:58bf:fa02:c0de:5 (Server)
  + The IP layer is responsible for routing the packets across networks.
* **Response:**
  + **Source IP:** 2a01:e0d:1:3:58bf:fa02:c0de:5 (Server)
  + **Destination IP:** 2401:f40:121b:f9:1cf5:c1a8:c1fb:8530 (Client)
  + The reverse route is used for server responses.

#### **Network Layer Operations:**

The network layer handles the routing of data packets across networks using IP addresses. It ensures that the data reaches the correct destination by encapsulating the data in IP packets, which are forwarded through intermediate routers. The source and destination IP addresses are used to determine the route.

#### **4. Data Link Layer (Ethernet)**

* **Request:**
  + **Source MAC:** 40:8d:5c:af:29:07 (GigaByteTech)
  + **Destination MAC:** 10:27:f5:03:f3:3f (TPLink)
  + The data link layer encapsulates the data into frames and facilitates communication between devices within the same local network.
* **Response:**
  + **Source MAC:** 10:27:f5:03:f3:3f (TPLink)
  + **Destination MAC:** 40:8d:5c:af:29:07 (GigaByteTech)
  + The reverse encapsulation takes place as the server responds to the client.

#### **Data Link Layer Operations:**

The data link layer is responsible for transferring data between devices that are directly connected on the same physical network. It encapsulates the data into frames, adding MAC addresses for proper delivery on the local network segment. This layer ensures the correct addressing and delivery of frames between network devices like the client and router.

### **Summary**

The analysis shows a typical HTTP interaction where the client sends a GET request, and the server responds with a 200 OK status, delivering HTML content. The OSI model ensures a structured encapsulation-de-encapsulation process, facilitating smooth communication. The geographic and hardware details provide further context to the source and destination of the packets.