**Don Bosco Institute of Technology, Kurla**
**Academic Year 2023-24**

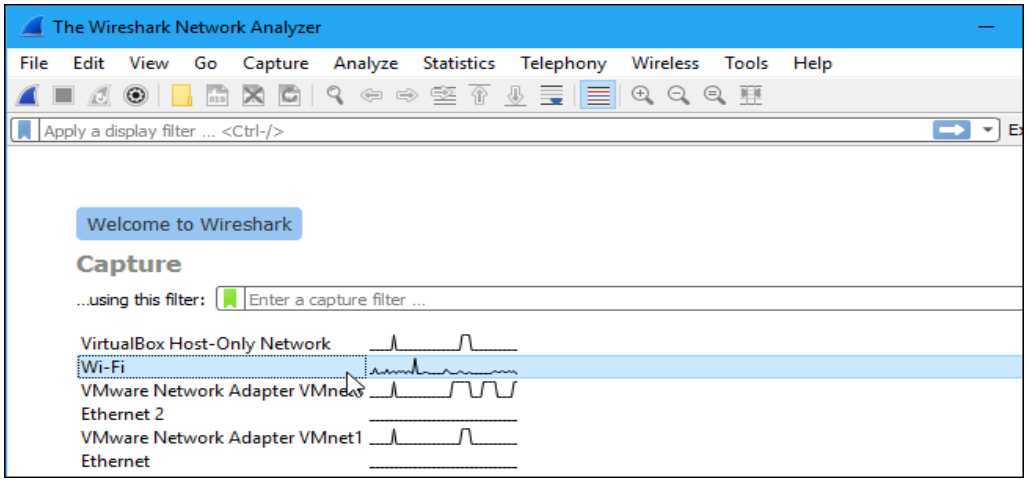
**EXPERIMENT NO. 6**

**SEMESTER: V** **DATE OF PERFORMANCE: 28th August 2024**

**SUBJECT: CN Lab** **DATE OF SUBMISSION: 01st September 2024**

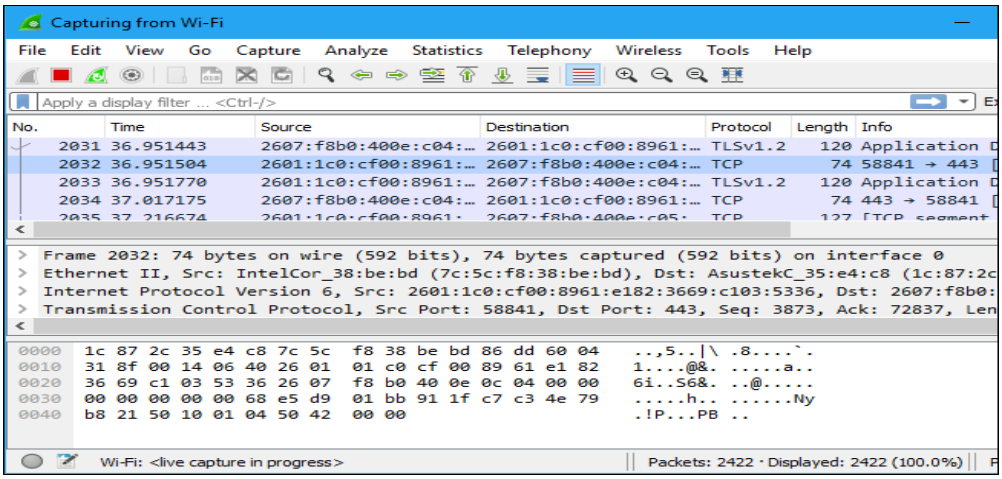
**NAME OF THE STUDENT: Dwayne George Nixon** **ROLL NO.: 21**

| | |
|---|---|
| **AIM** | Use wire shark to understand the TCP/IP operations.Ethernet Layer: Frame header, Frame size etc.<br>● Data Link Layer: MAC address, ARP (IP and MAC address binding)<br>● Network Layer: IP Packet (header, fragmentation), ICMP (Query andEcho)<br>● Transport Layer: TCP Ports, TCP handshake segments etc.<br>● Application Layer: DHCP, FTP, HTTP header formats |
| **LEARNING OBJECTIVE** | Students will be able to sniff and find protocol stack details from packets<br>Students will be able to use Wireshark filters effectively. |
| **LEARNING OUTCOME** | The student will illustrate the use of Wireshark tool to understand the operations ifTCP/IP layers. |
| **COURSE OUTCOME** | CSL502.5: Review various operations of TCP/IP layers using Wire shark. |
| **PROGRAM OUTCOME** | PO1,PO2,PO3,PO4,PO5,PO9,PO10,PSO1,PSO2,PSO3 |
| **BLOOM'S TAXONOMY LEVEL** | Evaluate |
| **THEORY** | Wireshark, a network analysis tool formerly known as Ethereal, captures packetsin real time and display them in human-readable format. Wireshark includes filters, color coding, and other features that let you dig deep into network traffic and inspect individual packets.<br>**Capturing Packets**<br>After downloading and installing Wireshark, you can launch it and double-clickthe name of a network interface under Capture to start capturing packets on thatinterface. For example, if you want to capture traffic on your wireless network, click your wireless interface. You can configure advanced features by clicking Capture > Options, but this isn't necessary for now. |

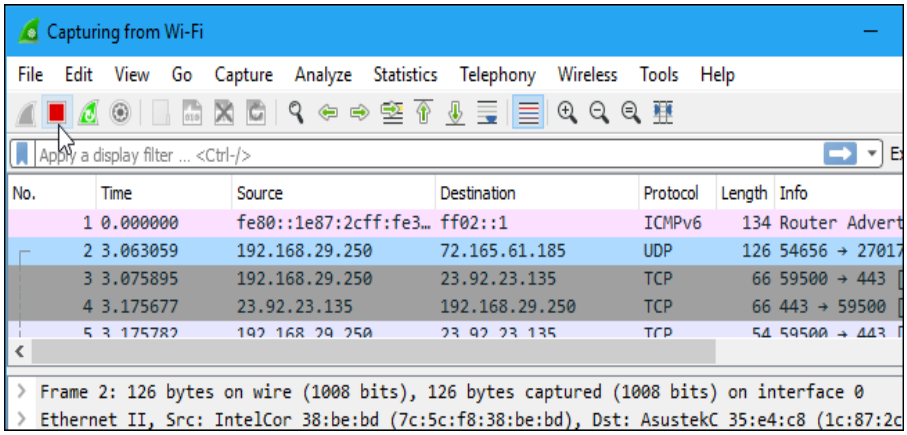



As soon as you click the interface's name, you'll see the packets start to appearin real time. Wireshark captures each packet sent to or from your system.

If you have promiscuous mode enabled—it's enabled by default—you'll also see all the other packets on the network instead of only packets addressed to your network adapter. To check if promiscuous mode is enabled, click Capture
> Options and verify the "Enable promiscuous mode on all interfaces" checkboxis activated at the bottom of this window.



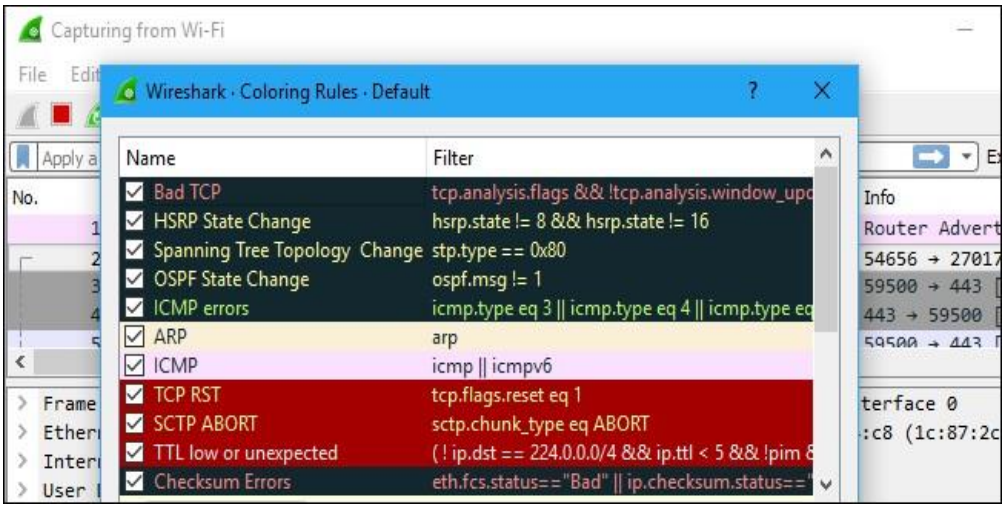

Click the red "Stop" button near the top left corner of the window when you wantto stop capturing traffic.




## Color Coding

You'll probably see packets highlighted in a variety of different colors. Wireshark uses colors to help you identify the types of traffic at a glance. By default, light purple is TCP traffic, light blue is UDP traffic, and black identifiespackets with errors—for example, they could have been delivered out of order.

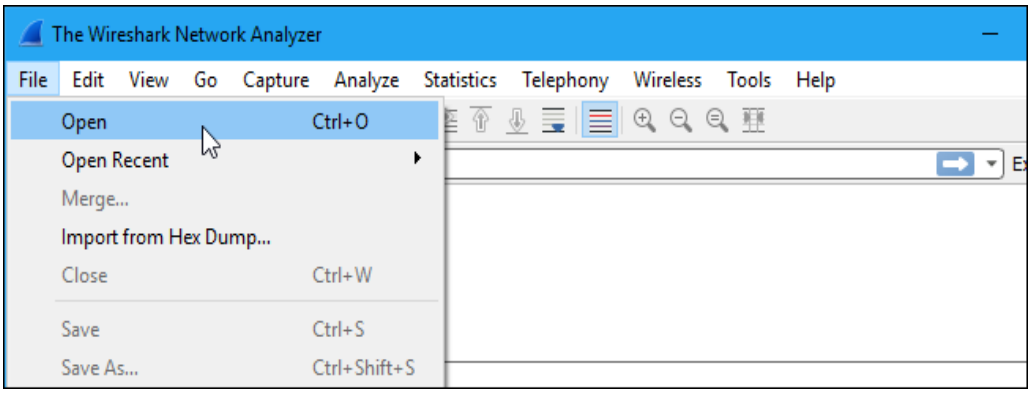
To view exactly what the color codes mean, click View > Coloring Rules. Youcan also customize and modify the coloring rules from here, if you like.




## Sample Captures

If there's nothing interesting on your own network to inspect, Wireshark's wikihas you covered. The wiki contains a page of sample capture files that you canload and inspect. Click File > Open in Wireshark and browse for your downloaded file to open one.

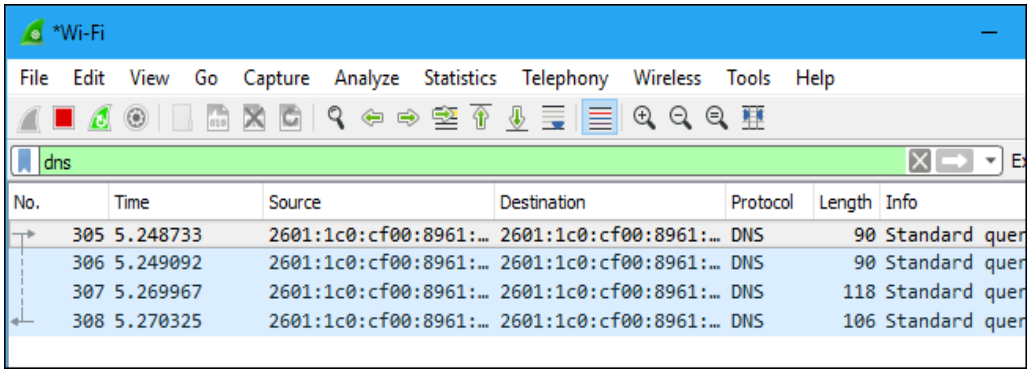
You can also save your own captures in Wireshark and open them later. ClickFile > Save to save your captured packets.




## Filtering Packets

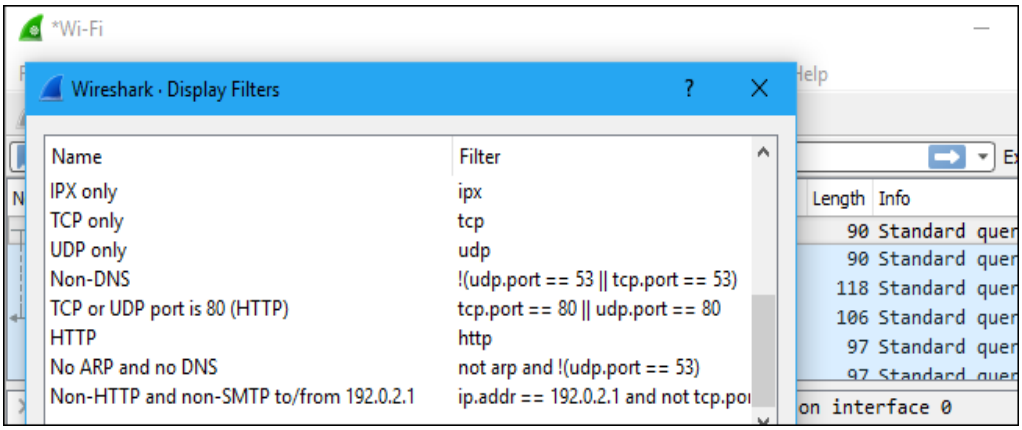
If you're trying to inspect something specific, such as the traffic a program sends when phoning home, it helps to close down all other applications using the network so you can narrow down the traffic. Still, you'll likely have a largeamount of packets to sift through. That's where Wireshark's filters come in.

The most basic way to apply a filter is by typing it into the filter box at the top of the window and clicking Apply (or pressing Enter). For example, type "dns"and you'll see only DNS packets. When you start typing, Wireshark will help you autocomplete your filter.



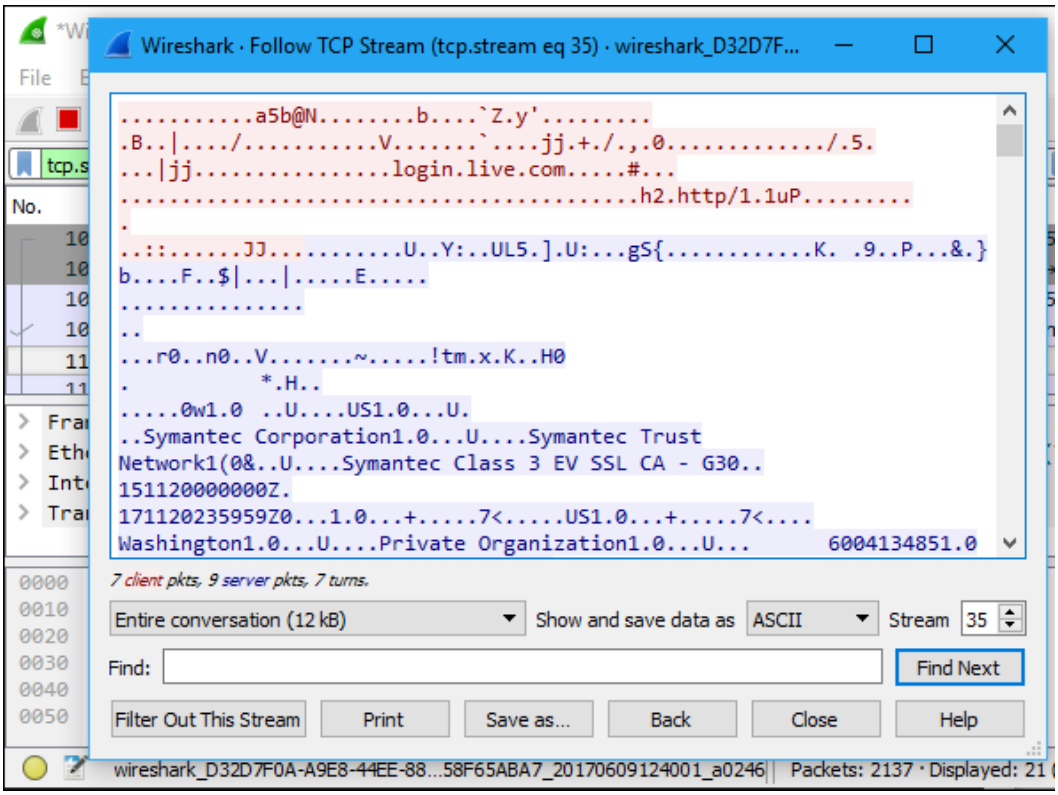

You can also click Analyze > Display Filters to choose a filter from among the default filters included in Wireshark. From here, you can add your own custom filters and save them to easily access them in the future.

For more information on Wireshark's display filtering language, readthe Building display filter expressions page in the official Wireshark documentation.

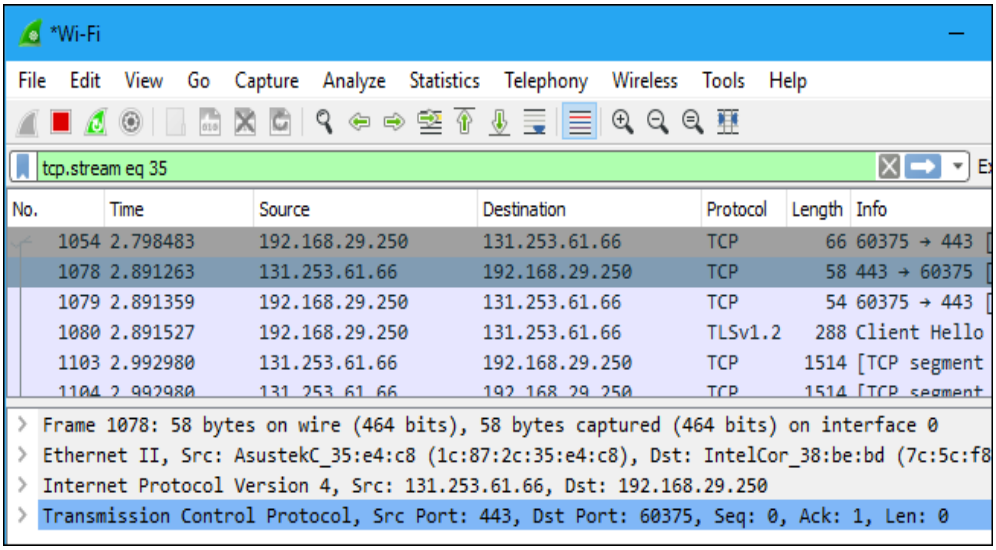



Another interesting thing you can do is right-click a packet and select Follow >TCP Stream.

You'll see the full TCP conversation between the client and the server. You canalso click other protocols in the Follow menu to see the full conversations for other protocols, if applicable.

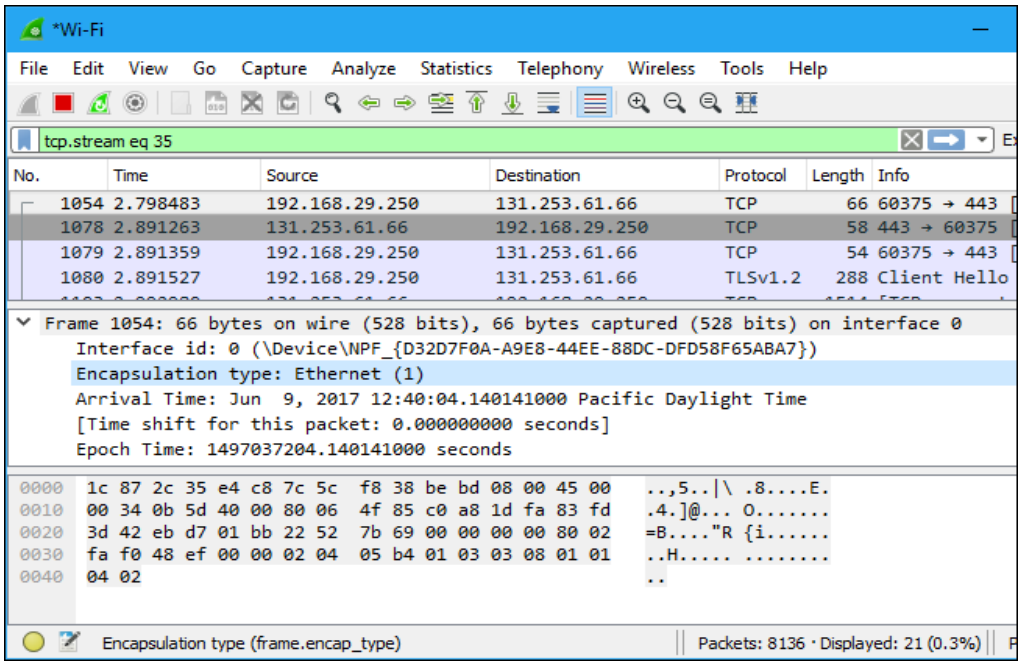


Close the window and you'll find a filter has been applied automatically.Wireshark is showing you the packets that make up the conversation.

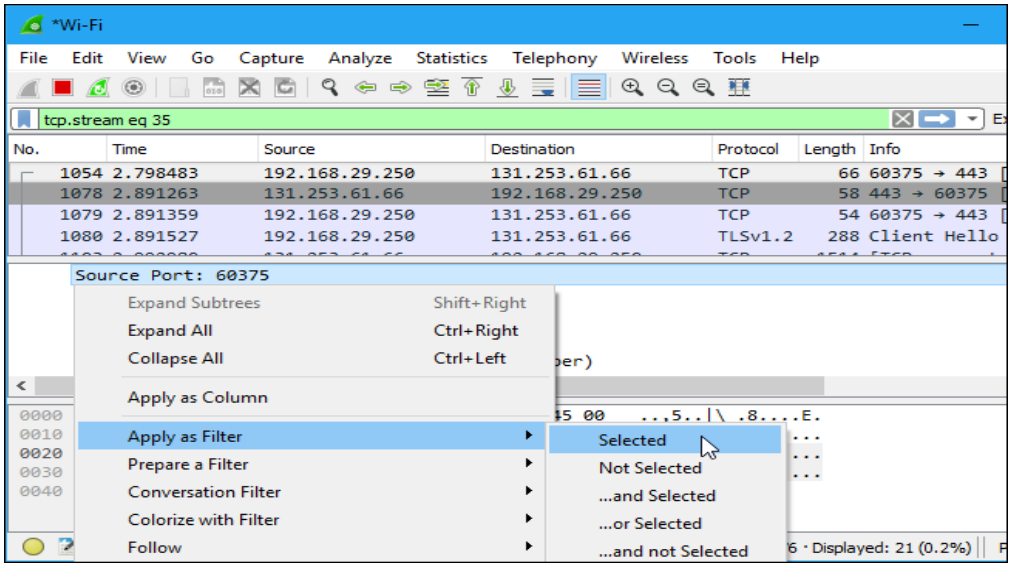



## Inspecting Packets

Click a packet to select it and you can dig down to view its details.



You can also create filters from here — just right-click one of the details and usethe Apply as Filter submenu to create a filter based on it.




Wireshark is an extremely powerful tool, and this tutorial is just scratching thesurface of what you can do with it. Professionals use it to debug network protocol implementations, examine security problems and inspect network protocol internals.

**LAB EXERCISE**

- Trace one connection using Wireshark and answer the following questionwith screenshots

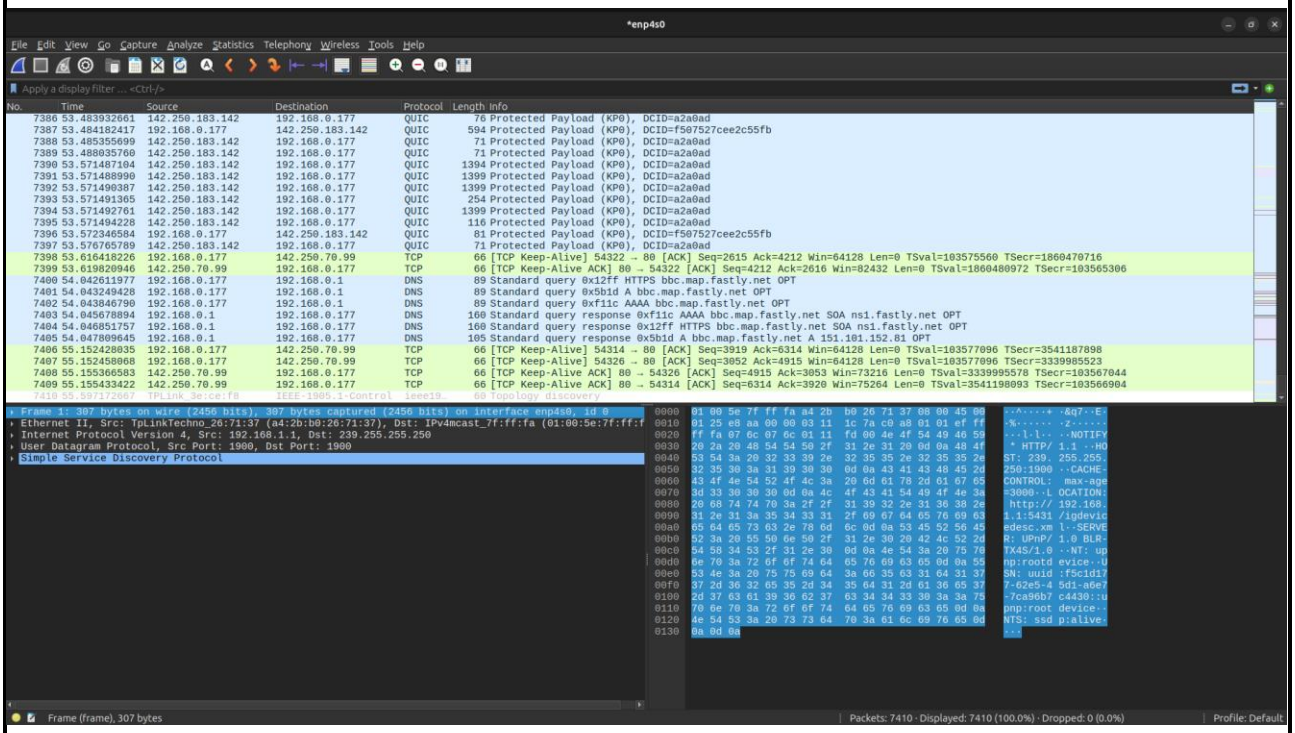
| |
|---|
| Is the frame an outgoing or an incoming frame? |
| Source IP address of the network-layer header in the frame: |
| Destination IP address of the network-layer header in the frame: |
| Total number of bytes in the whole frame: |
| Number of bytes in the Ethernet (data-link layer) header: |
| Number of bytes in the IP header: |
| Number of bytes in the TCP header: |
| Total bytes in the message (at the application layer): |

Also try:

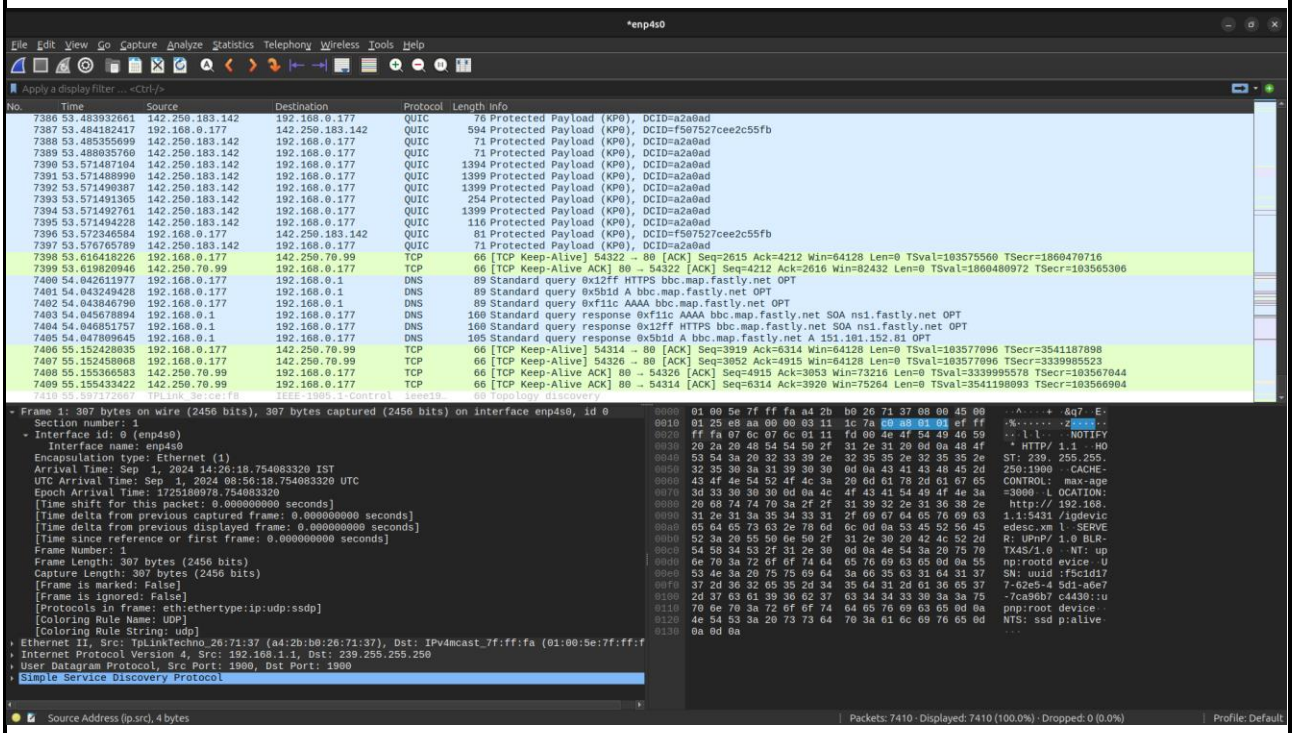
- Capture packets if you were using youtube (TCP contains youtube or ipaddress of youtube)
- Login through some unsecure website and capture the password

**LAB EXERCISE OUTCOMES:**

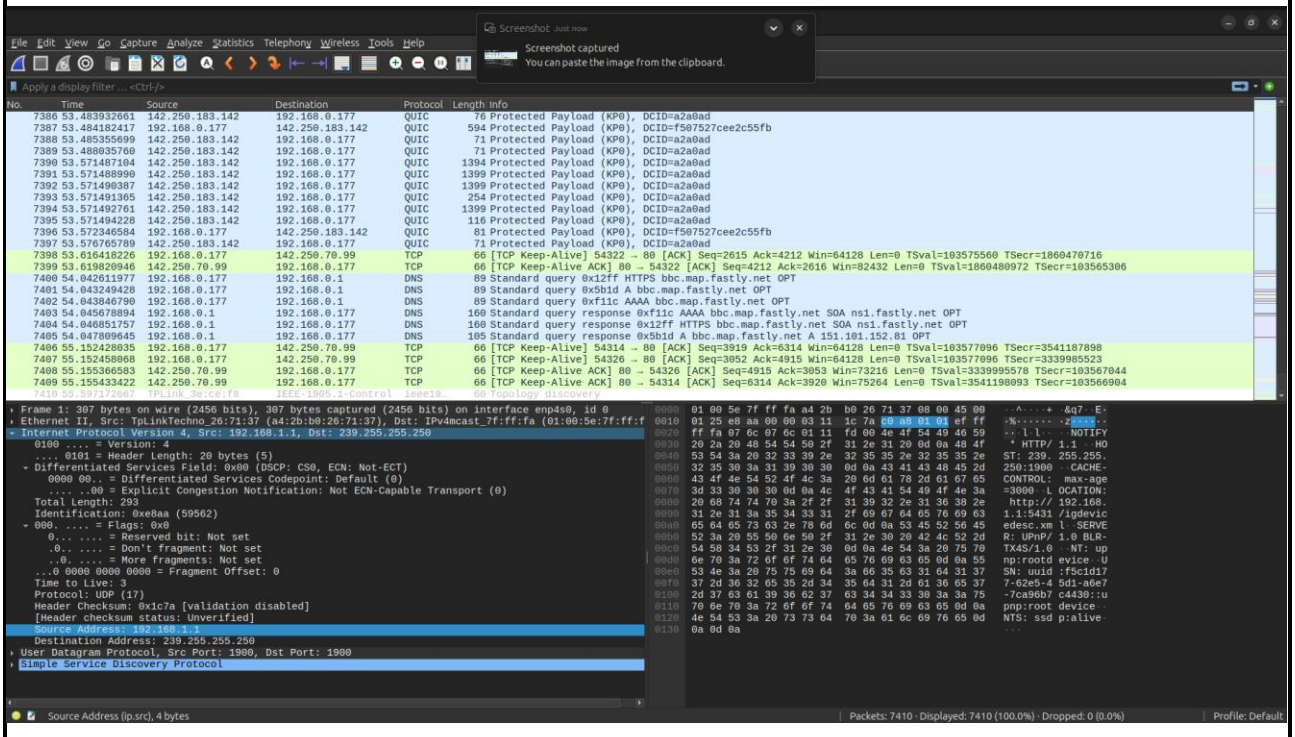
- Trace one connection using Wireshark and answer the following questionwith screenshots



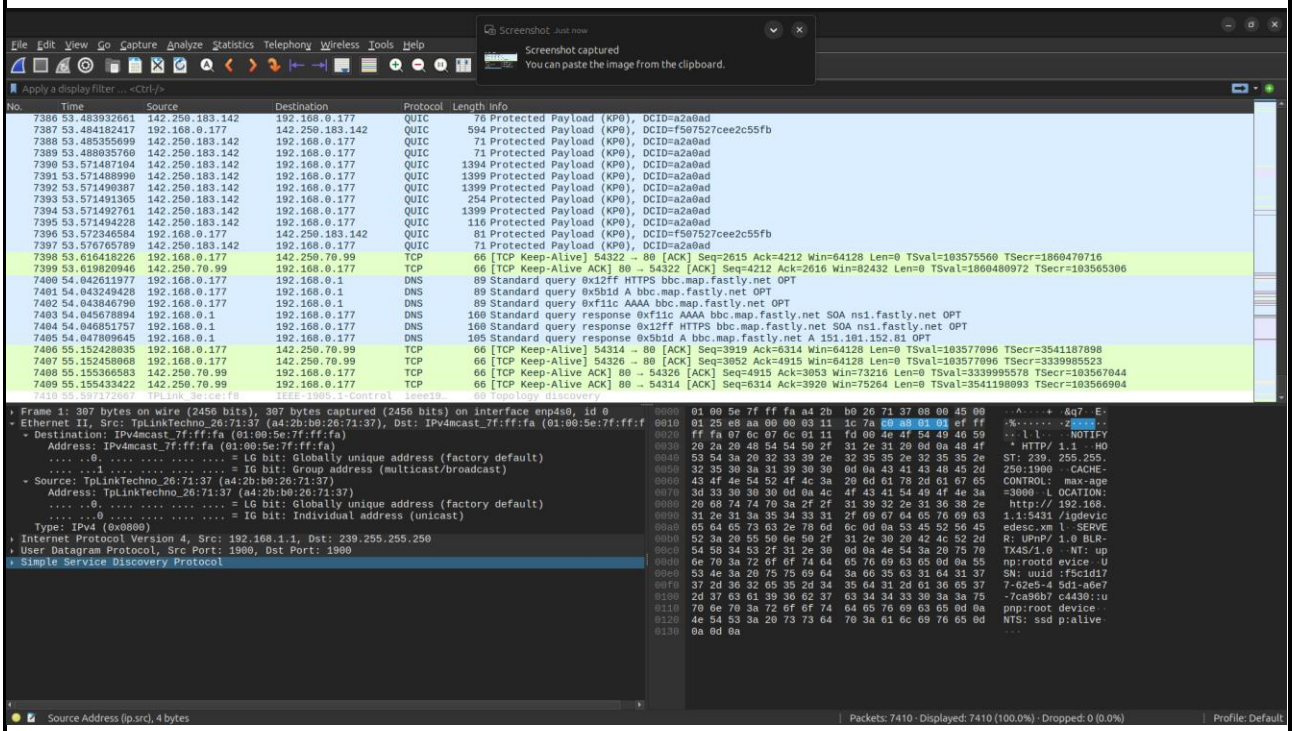

**Connection to the Internet via Ethernet**



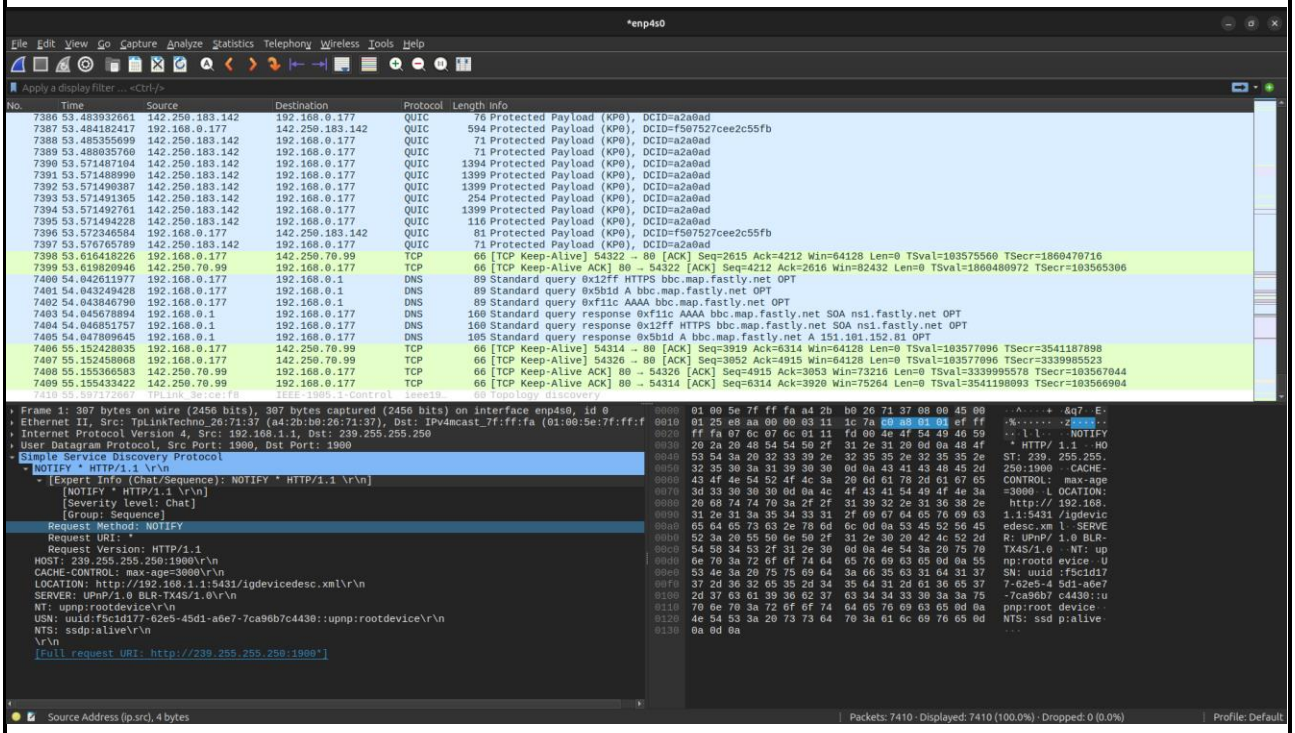

**Frame Details**



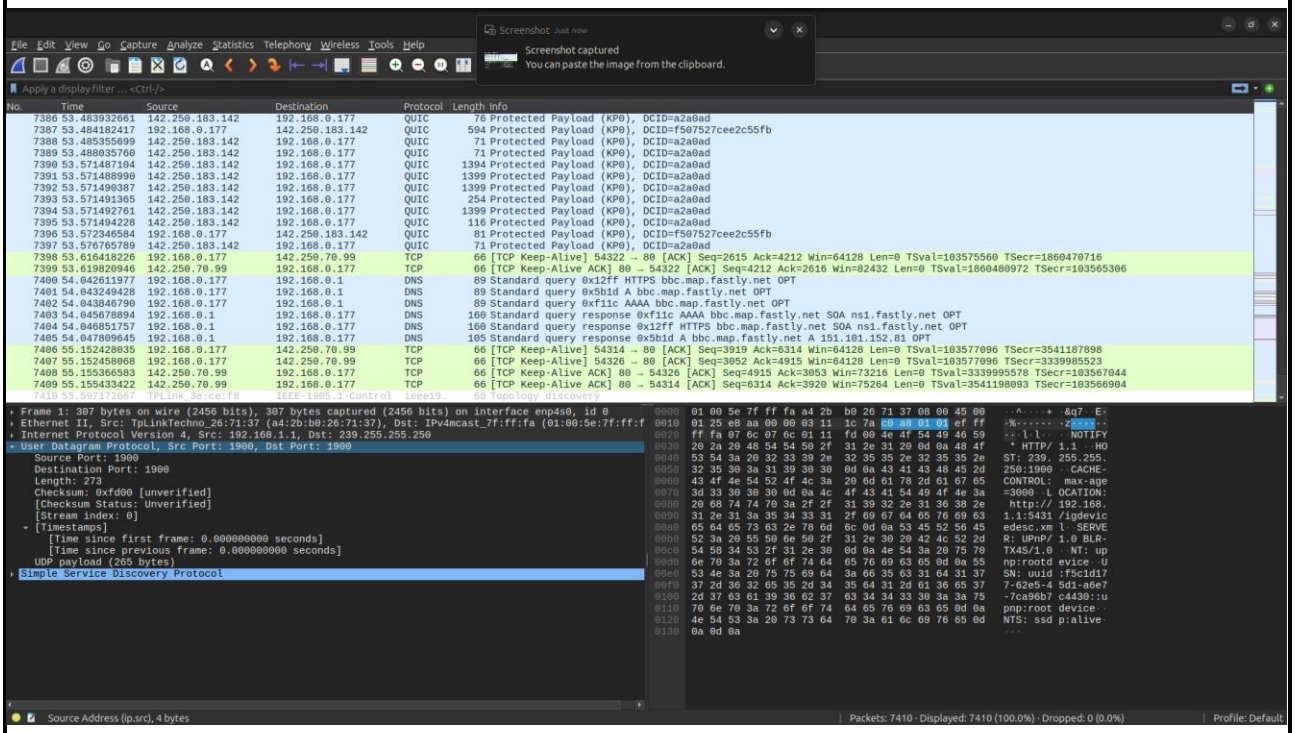

**Internet Protocol Details**




**Ethernet Details**




**Simple Service Discovery Protocol**




## User Datagram Protocol

**Questions on the Given Images:**

- **Is the frame an outgoing or an incoming frame?**
  The given frame is an incoming frame as the Destination MAC Address matches the computers MAC Address.

- **Source IP address of the network-layer header in the frame:**
  The Source Address is 192.168.1.1

- **Destination IP address of the network-layer header in the frame:**
  The Destination Address is 239.255.255.250

- **Total number of bytes in the whole frame:**
  The number of bytes in the whole frame is 307 bytes.

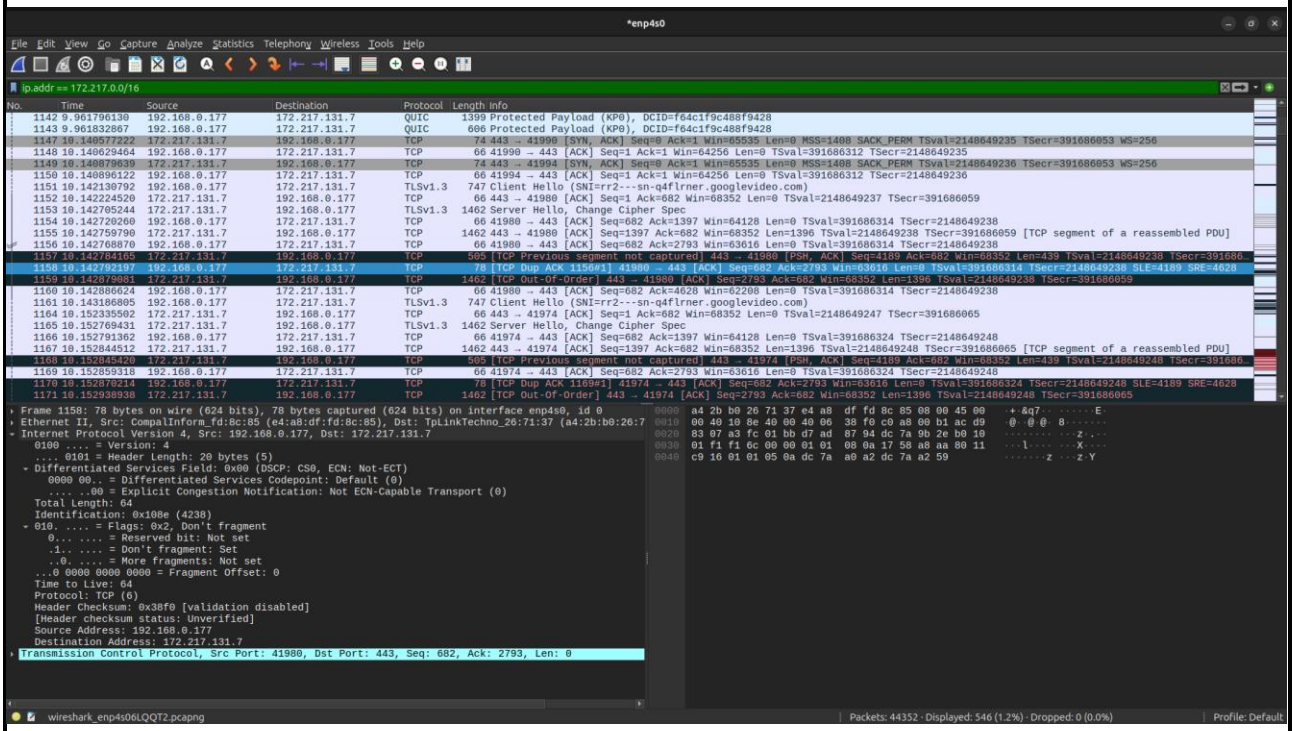
- **Number of bytes in the Ethernet (data-link layer) header:**
  The number of bytes in the Ethernet header is 14 bytes.

- **Number of bytes in the IP header:**
  The number of bytes in the IP header is 20 bytes.

- **Number of bytes in the TCP header:**
  The number of bytes in the TCP header is 20 bytes.

- **Total bytes in the message (at the application layer):**
  The total bytes in the message is 273 bytes.

- Capture packets if you were using youtube (TCP contains youtube or ipaddress of youtube)




**Watching a Youtube Video while having Wireshark enabled.**

When capturing and analyzing YouTube traffic using Wireshark, I observed the following:

**1. DNS Resolution:** My computer requested the IP address for `youtube.com`, and the DNS server responded with YouTube's IP addresses, allowing the connection to initiate.

**2. TCP Handshake:** A three-way TCP handshake was established with YouTube's server, confirming a reliable connection for data transfer.

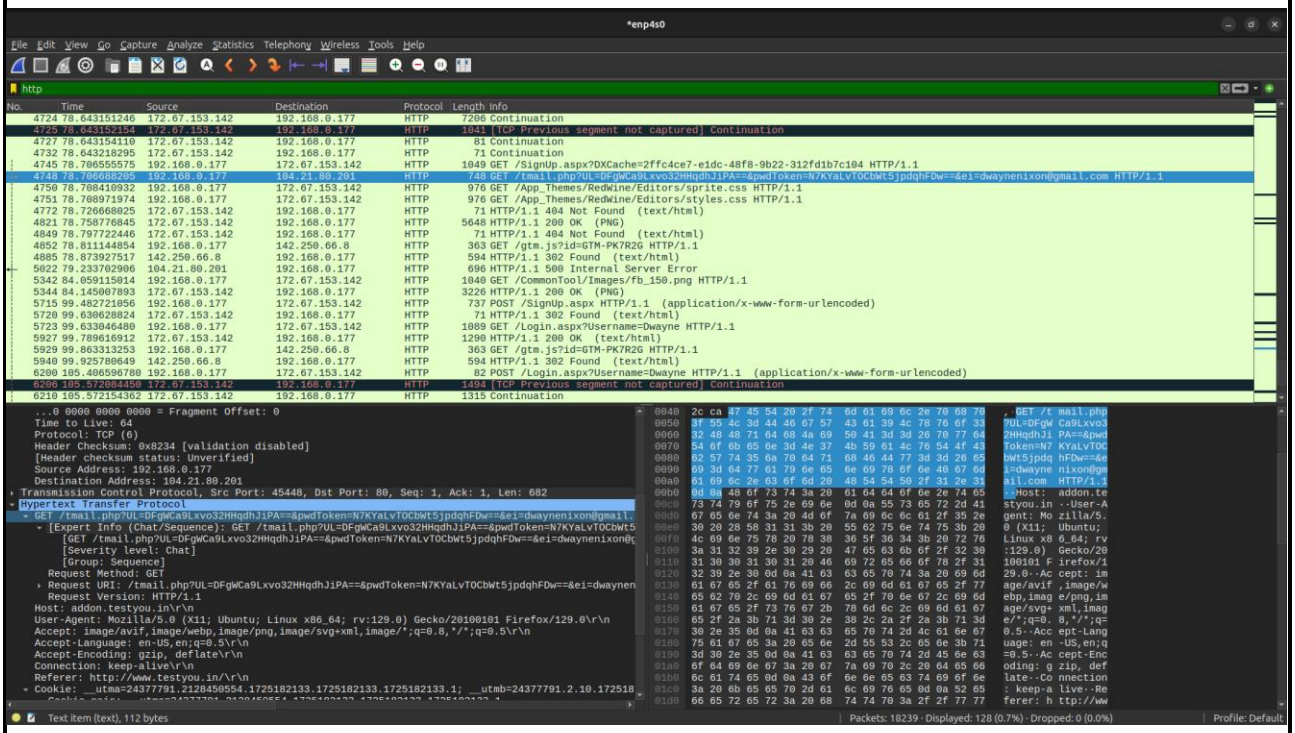
**3. TLS Handshake:** The TLS handshake followed, securing the connection through encryption, ensuring that the video data was transmitted securely.

**4. Video Data Transfer:** I observed large, continuous TCP packets representing the streaming video data. Although encrypted, the size and frequency of the packets indicated active video streaming.

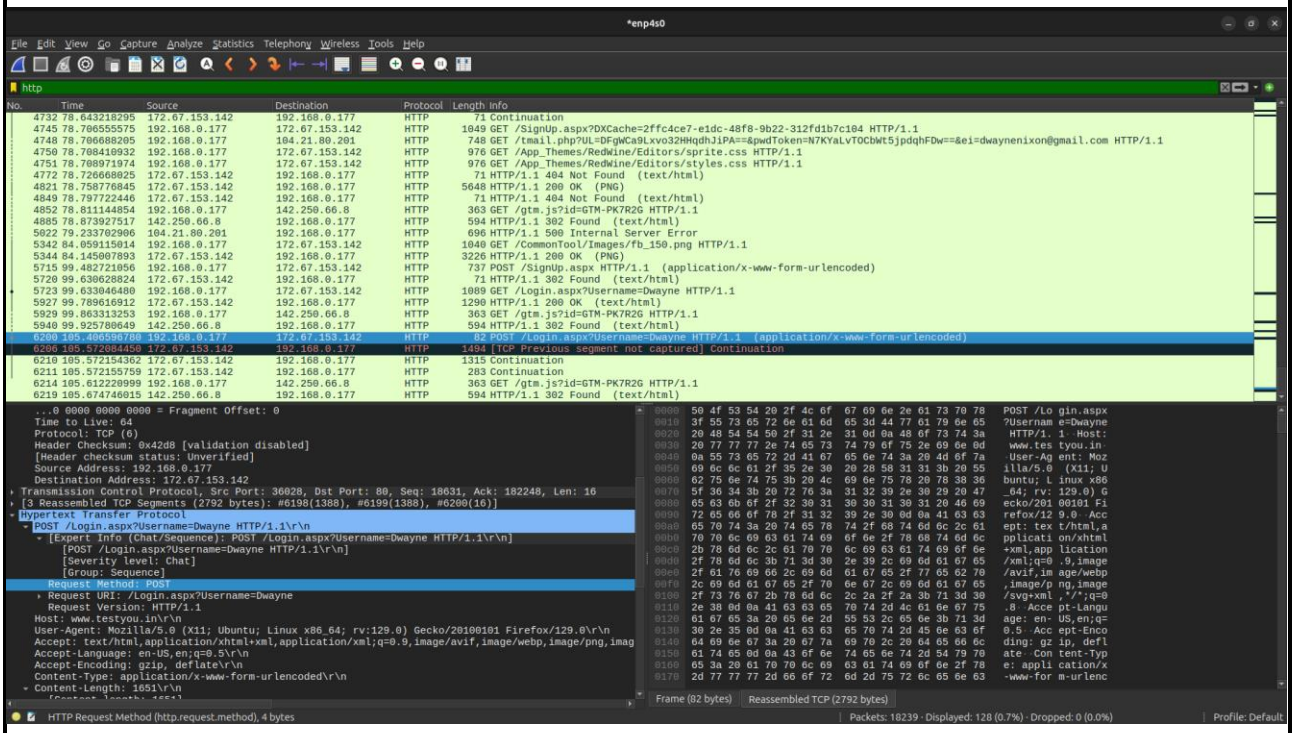
**5. Session Characteristics:** The streaming session was marked by consistent data flow and occasional retransmissions, reflecting the nature of video streaming traffic.

These results provide a clear view of how YouTube streams content, from DNS resolution to encrypted data transfer, ensuring secure and reliable video delivery.

- Login through some unsecure website and capture the password




**Creating an Account in an Unsafe Website.**




**Logging into that Account on the Same Website.**

| | |
|---|---|
| | When logging into an unsecure website and capturing the traffic with Wireshark, here's what I observed:<br><br>**1. Unencrypted Login Data:** Upon logging into the unsecure website (using HTTP), I captured the traffic and found that the login credentials, including the username and password, were transmitted in plain text.<br><br>**2. HTTP POST Request:** The credentials were included in an HTTP POST request, which was sent from my computer to the server. Since the connection was not encrypted, the data was easily visible in the packet capture.<br><br>**3. Captured Password:** By inspecting the HTTP packet, I was able to see the password and other form data directly in the packet details under the `Form Data` or `Parameters` section.<br><br>This result highlights the risks of using unsecure websites, as sensitive information like passwords can be intercepted easily by anyone capturing the network traffic. |
| **REFERENCES** | • A complete guide to Wireshark- https://www.javatpoint.com/wireshark<br>• https://www.youtube.com/watch?v=TkCSr30UojM |