Task 5 –

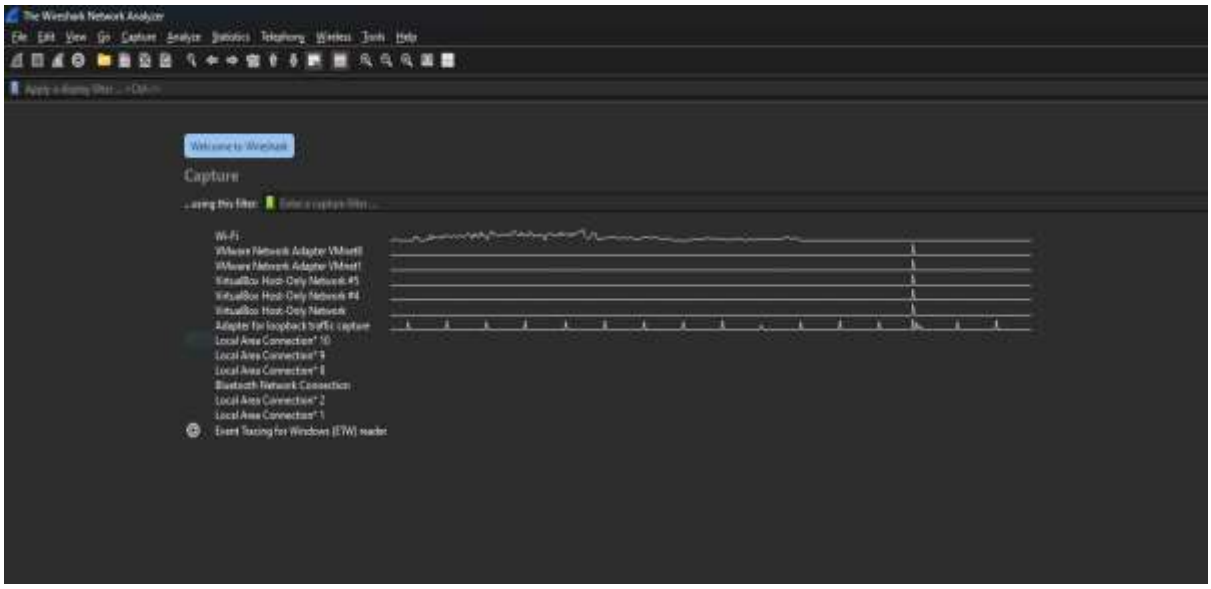
Wireshark Already installed for previous tasks

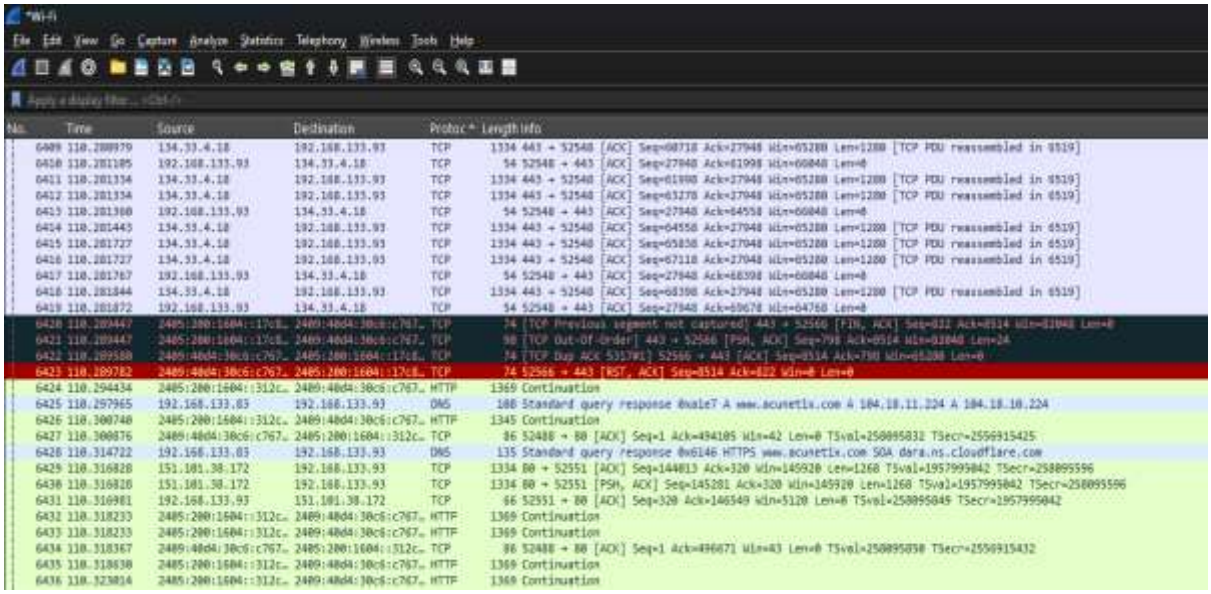
Now I need to capture live packets via available interface.

Here I am using wifi so I have to capture traffic

Before capture packets need to generate some traffic via browser

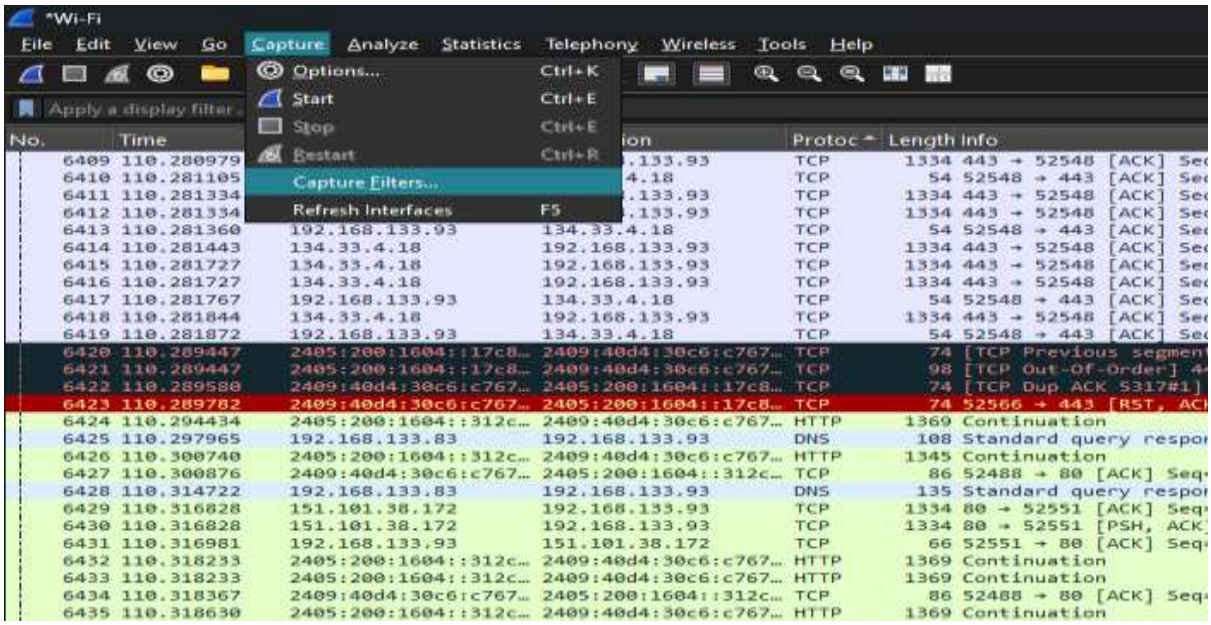
Then we capture packets of wifi

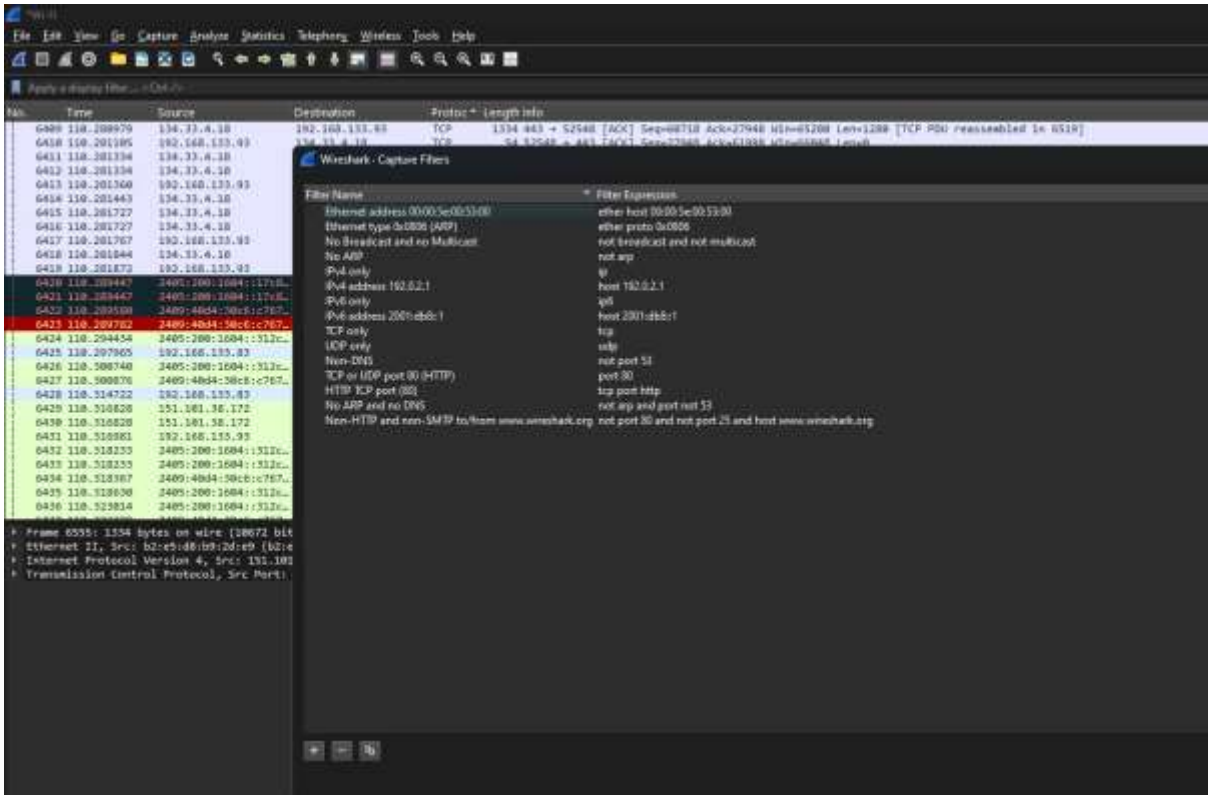




Here is some packets captured via wireshark now we can filter out it

And find out some protocols





Here some protocol mentioned above

The image displays a Wireshark "Capture Filters" window, listing various pre-defined filters and their corresponding filter expressions. These filters are used to capture specific network traffic based on different criteria. Here's a report based on the mentioned protocols:

Report on Wireshark Capture Filters by Protocol

This report analyzes the capture filters presented in the provided Wireshark screenshot, categorized by the network protocols they target.

1. Ethernet Layer Filters (Layer 2)

- **Ethernet address 00:00:5e:00:53:00:**
    - **Filter Name:** Ethernet address 00:00:5e:00:53:00
    - **Filter Expression:** ether host 00:00:5e:00:53:00
    - **Purpose:** This filter captures all Ethernet frames where the source or destination MAC address is 00:00:5e:00:53:00. This is useful for monitoring traffic to/from a specific network interface or device.
- **Ethernet type 0x0806 (ARP):**
    - **Filter Name:** Ethernet type 0x0806 (ARP)
    - **Filter Expression:** ether proto 0x0806
    - **Purpose:** This filter specifically captures Address Resolution Protocol (ARP) packets. ARP is used to resolve IP addresses to MAC addresses. This filter is crucial for troubleshooting connectivity issues at the data link layer.
- **No Broadcast and no Multicast:**
    - **Filter Name:** No Broadcast and no Multicast
    - **Filter Expression:** not broadcast and not multicast
    - **Purpose:** This filter excludes broadcast and multicast traffic, focusing only on unicast communication. This can help in reducing noise when analyzing traffic between specific hosts.
- **No ARP:**
    - **Filter Name:** No ARP
    - **Filter Expression:** not arp
    - **Purpose:** This filter excludes ARP packets from the capture. Useful when ARP traffic is not relevant to the analysis and would otherwise clutter the capture.

2. IP Layer Filters (Layer 3)

- **IPv4 only:**
    - **Filter Name:** IPv4 only
    - **Filter Expression:** ip
    - **Purpose:** This filter captures only IPv4 packets, excluding any IPv6 or other network layer protocols.
- **IPv4 address 192.0.2.1:**
    - **Filter Name:** IPv4 address 192.0.2.1
    - **Filter Expression:** host 192.0.2.1
    - **Purpose:** This filter captures all IPv4 packets where the source or destination IP address is 192.0.2.1. Essential for monitoring traffic to/from a specific host on an IPv4 network.
- **IPv6 only:**
    - **Filter Name:** IPv6 only
    - **Filter Expression:** ip6
    - **Purpose:** This filter captures only IPv6 packets, excluding IPv4 or other network layer protocols.
- **IPv6 address 2001:db8::1:**
    - **Filter Name:** IPv6 address 2001:db8::1
    - **Filter Expression:** host 2001:db8::1
    - **Purpose:** This filter captures all IPv6 packets where the source or destination IP address is 2001:db8::1. Useful for monitoring traffic to/from a specific host on an IPv6 network.

3. Transport Layer Filters (Layer 4)

- **TCP only:**
    - **Filter Name:** TCP only
    - **Filter Expression:** tcp
    - **Purpose:** This filter captures only Transmission Control Protocol (TCP) segments. TCP is a connection-oriented protocol used for reliable data transfer.

- **UDP only:**
    - **Filter Name:** UDP only
    - **Filter Expression:** udp
    - **Purpose:** This filter captures only User Datagram Protocol (UDP) datagrams. UDP is a connectionless protocol often used for speed over reliability.
- **TCP or UDP port 80 (HTTP):**
    - **Filter Name:** TCP or UDP port 80 (HTTP)
    - **Filter Expression:** port 80
    - **Purpose:** This filter captures traffic on port 80, regardless of whether it's TCP or UDP. Port 80 is primarily used for HTTP (Hypertext Transfer Protocol) traffic.
- **HTTP TCP port (80):**
    - **Filter Name:** HTTP TCP port (80)
    - **Filter Expression:** tcp port http
    - **Purpose:** This filter specifically captures TCP traffic on the standard HTTP port (80). This is a more precise way to filter for web Browse traffic.

4. Application Layer and Combined Protocol Filters

- **Non-DNS:**
    - **Filter Name:** Non-DNS
    - **Filter Expression:** not port 53
    - **Purpose:** This filter excludes DNS (Domain Name System) traffic, which typically uses port 53 (both TCP and UDP). This is useful when you want to focus on other types of network communication.
- **No ARP and no DNS:**
    - **Filter Name:** No ARP and no DNS
    - **Filter Expression:** not arp and not port 53
    - **Purpose:** This filter combines two previous exclusions, removing both ARP and DNS traffic from the capture. This helps in narrowing down the focus to application-specific data.

- **Non-HTTP and non-SMTP to/from www.wireshark.org:**
    - **Filter Name:** Non-HTTP and non-SMTP to/from www.wireshark.org
    - **Filter Expression:** not port 80 and not port 25 and host www.wireshark.org
    - **Purpose:** This advanced filter demonstrates the ability to combine multiple conditions. It captures traffic to/from www.wireshark.org but *excludes* HTTP (port 80) and SMTP (port 25) traffic. This could be used, for example, to analyze other types of communication with a specific web server, such as ICMP pings or SSH connections.

Now I can generate pcap file for save results of wireshark packets.