

# Wireshark Packet Analysis Report

---

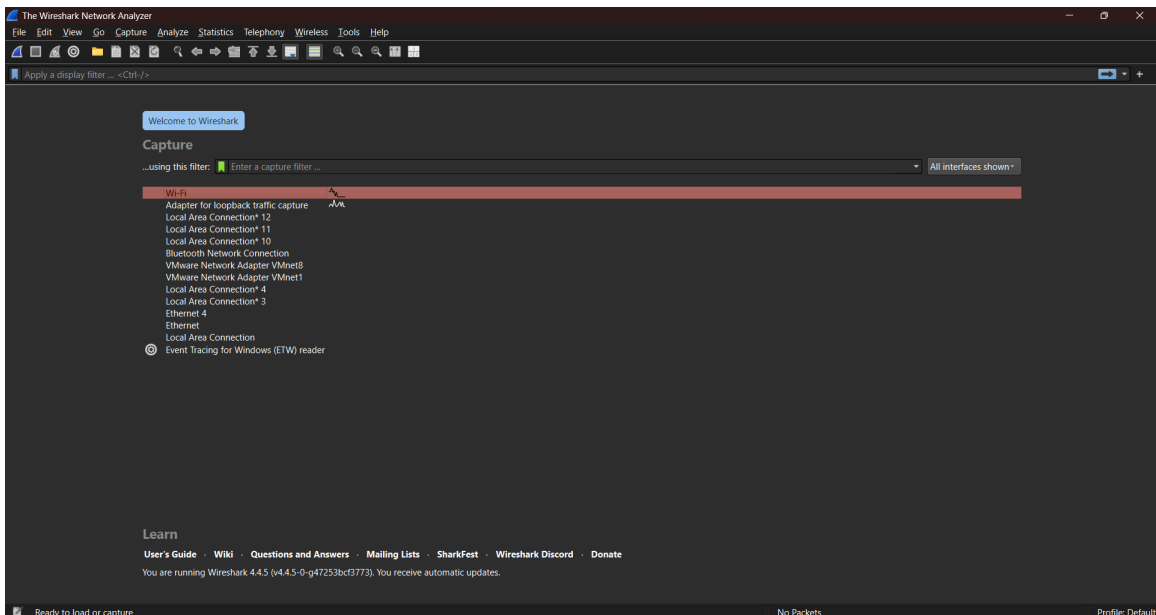
## Objective

To perform a hands-on network traffic capture using Wireshark and analyze packets across various protocols.

## Procedure

1. Installed Wireshark from the official website (<https://www.wireshark.org/>).
2. Launched Wireshark and started capturing packets on the active network interface.
3. Opened a browser and visited a website (e.g., <https://example.com>) or executed 'ping google.com' to generate network traffic.
4. After about a minute of activity, stopped the packet capture.
5. Applied display filters such as 'http', 'dns', and 'tcp' to analyze specific protocol packets.
6. Identified at least three different protocols from the captured traffic.
7. Exported the captured data as a .pcap file for documentation and future reference.

## Screenshots



## All incoming and outgoing network traffics :

The image shows a Wireshark network traffic capture. The top pane displays a list of packets, and the bottom pane shows the details of a selected packet (No. 2381).

No.	Time	Source	Destination	Protocol	Length	Info
2370	3.761342	2409:40f4:3104:a082...	64:ff9b::2baf:8ada...	TCP	74	16995 → 80 [ACK] Seq=3073 Ack=2240804 Win=524032 Len=0
2371	3.762756	64:ff9b::2baf:8ada...	2409:40f4:3104:a082...	TCP	1354	80 → 16995 [ACK] Seq=2240804 Ack=3073 Win=524288 Len=1280 [TCP PDU reassembled in 2583]
2372	3.762756	64:ff9b::2baf:8ada...	2409:40f4:3104:a082...	TCP	1354	80 → 16995 [ACK] Seq=2242184 Ack=3073 Win=524288 Len=1280 [TCP PDU reassembled in 2583]
2373	3.762756	64:ff9b::2baf:8ada...	2409:40f4:3104:a082...	TCP	1354	80 → 16995 [ACK] Seq=2243464 Ack=3073 Win=524288 Len=1280 [TCP PDU reassembled in 2583]
2374	3.762756	64:ff9b::2baf:8ada...	2409:40f4:3104:a082...	TCP	1354	80 → 16995 [ACK] Seq=2244744 Ack=3073 Win=524288 Len=1280 [TCP PDU reassembled in 2583]
2375	3.762756	64:ff9b::2baf:8ada...	2409:40f4:3104:a082...	TCP	1354	80 → 16995 [ACK] Seq=2246024 Ack=3073 Win=524288 Len=1280 [TCP PDU reassembled in 2583]
2376	3.762756	64:ff9b::2baf:8ada...	2409:40f4:3104:a082...	TCP	1354	80 → 16995 [ACK] Seq=2247304 Ack=3073 Win=524288 Len=1280 [TCP PDU reassembled in 2583]
2377	3.762756	64:ff9b::2baf:8ada...	2409:40f4:3104:a082...	TCP	1354	80 → 16995 [ACK] Seq=2248584 Ack=3073 Win=524288 Len=1280 [TCP PDU reassembled in 2583]
2378	3.762756	64:ff9b::2baf:8ada...	2409:40f4:3104:a082...	TCP	1354	80 → 16995 [ACK] Seq=2249864 Ack=3073 Win=524288 Len=1280 [TCP PDU reassembled in 2583]
2379	3.762756	64:ff9b::2baf:8ada...	2409:40f4:3104:a082...	TCP	1354	80 → 16995 [ACK] Seq=2251144 Ack=3073 Win=524288 Len=1280 [TCP PDU reassembled in 2583]
2380	3.762811	2409:40f4:3104:a082...	64:ff9b::2baf:8ada...	TCP	74	16995 → 80 [ACK] Seq=3073 Ack=2252424 Win=524032 Len=0
2381	3.763613	64:ff9b::2baf:8ada...	2409:40f4:3104:a082...	TCP	1354	80 → 16995 [ACK] Seq=2252424 Ack=3073 Win=524288 Len=1280 [TCP PDU reassembled in 2583]
2382	3.763613	64:ff9b::2baf:8ada...	2409:40f4:3104:a082...	TCP	1354	80 → 16995 [ACK] Seq=2253704 Ack=3073 Win=524288 Len=1280 [TCP PDU reassembled in 2583]
2383	3.763613	64:ff9b::2baf:8ada...	2409:40f4:3104:a082...	TCP	1354	80 → 16995 [ACK] Seq=2254984 Ack=3073 Win=524288 Len=1280 [TCP PDU reassembled in 2583]
2384	3.763613	64:ff9b::2baf:8ada...	2409:40f4:3104:a082...	TCP	1354	80 → 16995 [ACK] Seq=2256264 Ack=3073 Win=524288 Len=1280 [TCP PDU reassembled in 2583]
2385	3.763613	64:ff9b::2baf:8ada...	2409:40f4:3104:a082...	TCP	1354	80 → 16995 [ACK] Seq=2257544 Ack=3073 Win=524288 Len=1280 [TCP PDU reassembled in 2583]
2386	3.763667	2409:40f4:3104:a082...	64:ff9b::2baf:8ada...	TCP	74	16995 → 80 [ACK] Seq=3073 Ack=2258824 Win=524032 Len=0
2387	3.764004	64:ff9b::2baf:8ada...	2409:40f4:3104:a082...	TCP	1354	80 → 16995 [ACK] Seq=2258824 Ack=3073 Win=524288 Len=1280 [TCP PDU reassembled in 2583]
2388	3.764004	64:ff9b::2baf:8ada...	2409:40f4:3104:a082...	TCP	1354	80 → 16995 [ACK] Seq=2260104 Ack=3073 Win=524288 Len=1280 [TCP PDU reassembled in 2583]
2389	3.764031	2409:40f4:3104:a082...	64:ff9b::2baf:8ada...	TCP	74	16995 → 80 [ACK] Seq=3073 Ack=2261384 Win=524032 Len=0

Frame 2381: 1354 bytes on wire (10832 bits), 1354 bytes captured (10832 bits) on interface \Device\NPF...  
Ethernet II, Src: 64:ff9b::2baf:8ada, Dst: 2409:40f4:3104:a082::c32:f031:9026  
Internet Protocol Version 6, Src: 2409:40f4:3104:a082::c32:f031:9026, Dst: 2409:40f4:3104:a082::c32:f031:9026  
Transmission Control Protocol, Src Port: 80, Dst Port: 16995, Seq: 2252424, Ack: 3073, Len: 1280

Packet 6528

## Capturing the network traffic of github.com using the command :

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

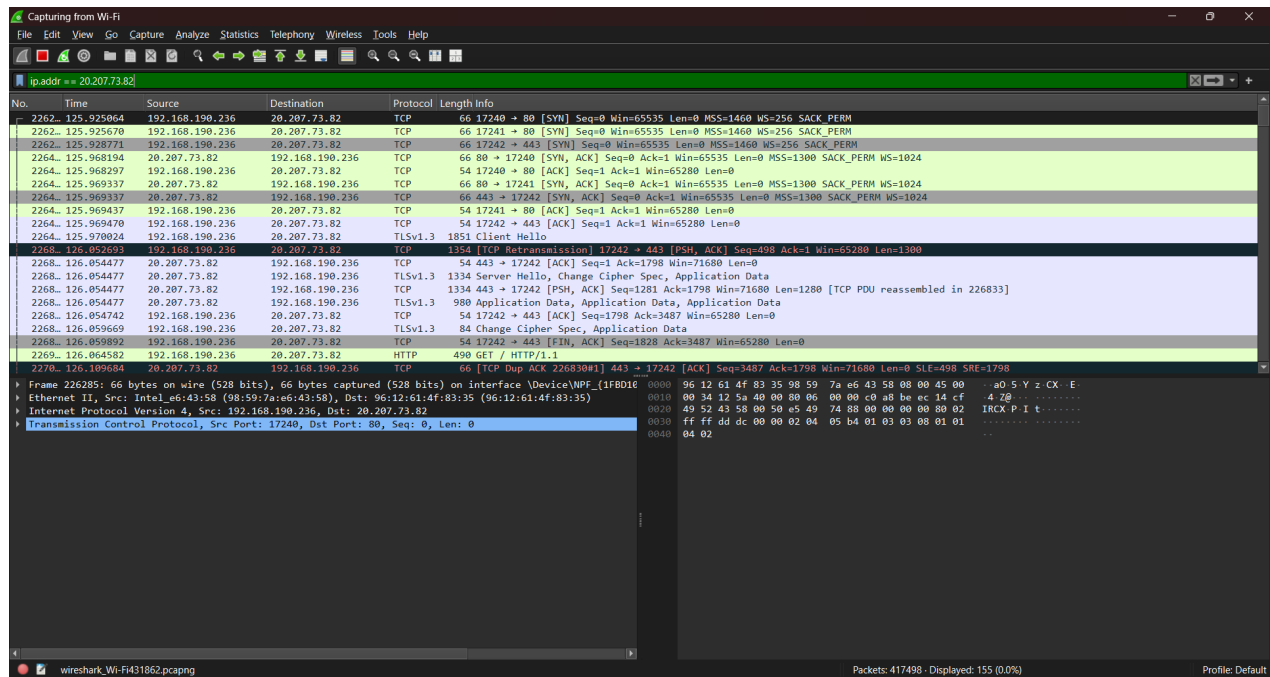
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\praga> nslookup github.com
Server: Unknown
Address: 192.168.190.116

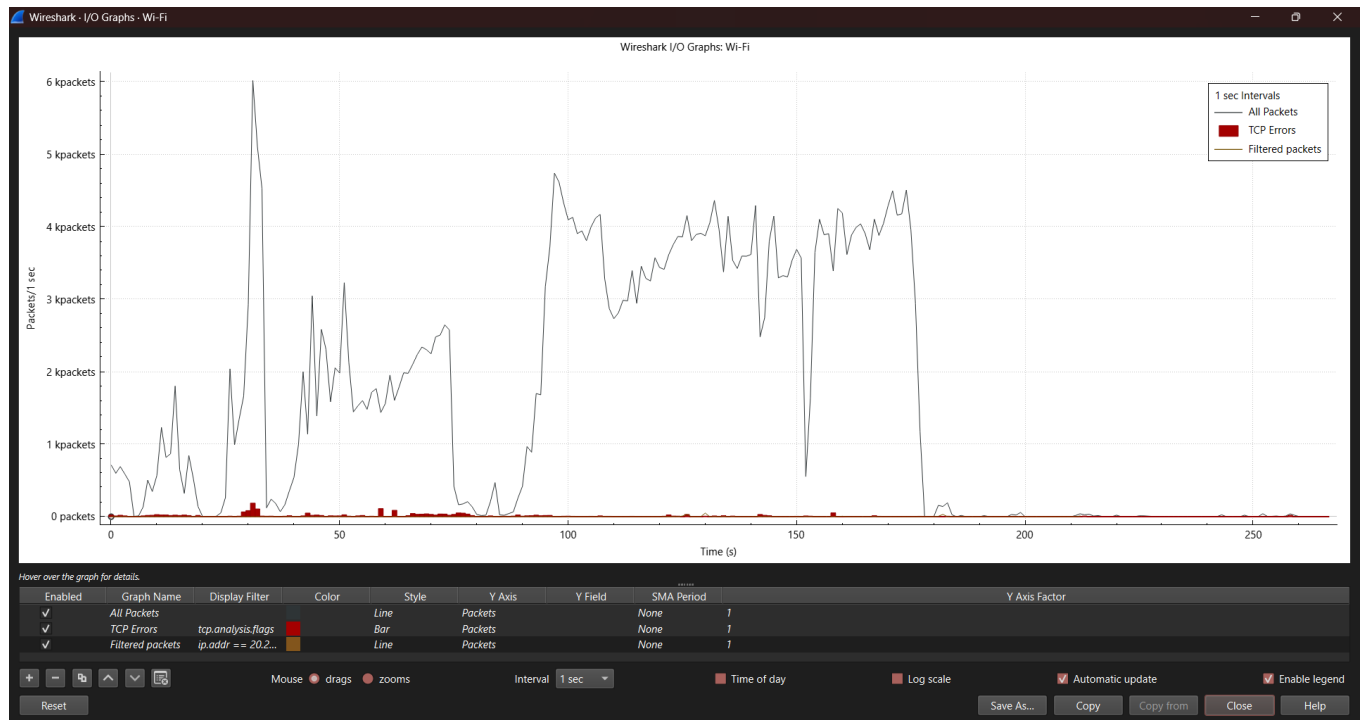
Non-authoritative answer:
Name: github.com
Addresses: 64:ff9b::14cf:4952
          20.207.73.82

PS C:\Users\praga>
```

ip.addr == 20.207.73.82



I/O graph



## Protocols Identified

From the captured network traffic, the following protocols were observed:

- HTTP (HyperText Transfer Protocol)
- DNS (Domain Name System)
- TCP (Transmission Control Protocol)

## Packet Summary

Each protocol carried specific types of data:

- HTTP packets included GET and POST requests for web content.
- DNS packets resolved domain names to IP addresses.
- TCP packets established and maintained connections with acknowledgments and sequence numbers.

## Conclusion

This hands-on task demonstrated the basics of capturing and analyzing network traffic. The identification of different protocols and packet-level inspection helped in understanding how data flows through a network.