**Lab 14: Packet Capture and Header Analysis by Wireshark (TCP, UDP, IP)**

**Theory**

**Wireshark** is a popular open-source network protocol analyzer. It captures and analyzes network traffic in real time, enabling users to inspect data packets for troubleshooting, monitoring network performance, and identifying security issues. With its graphical interface, Wireshark supports a wide range of protocols, offering detailed insights into network communications. It is an essential tool for both network administrators and cybersecurity professionals.

**TCP (Transmission Control Protocol)** is a connection-oriented protocol that ensures reliable data transmission between devices. It establishes a connection before data transfer, provides error checking, and guarantees that data packets are delivered in order and without duplication. TCP is commonly used for applications such as web browsing (HTTP/HTTPS) and email.

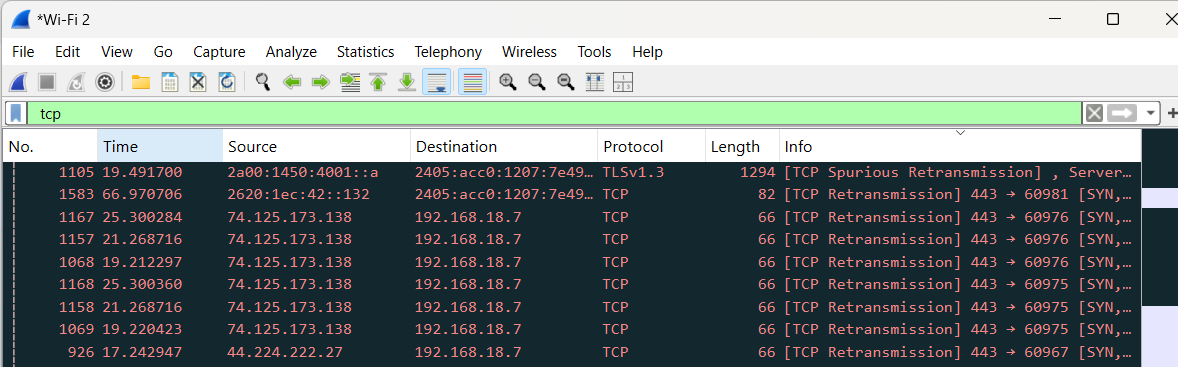
**UDP (User Datagram Protocol)** is a connectionless protocol that prioritizes speed over reliability. It sends data packets without establishing a connection or ensuring delivery, making it ideal for applications like video streaming, online gaming, and VoIP, where low latency is crucial and occasional data loss is acceptable.

**IP (Internet Protocol)** is the primary protocol for routing packets across networks. Operating at the network layer, it assigns IP addresses to devices and ensures data packets are forwarded to the correct destination through routing. IP works with both TCP and UDP to facilitate end-to-end communication over the internet.

**Network Interface Selection and Traffic Filtering**

**Steps:**

1. **Open Wireshark** and select the network interface (Wi-Fi or Ethernet) where you want to capture traffic.
2. **Click the Start button** to begin capturing live traffic.
3. **Apply a filter** to focus on specific traffic, such as tcp for TCP traffic, udp for UDP traffic, or ip for general IP traffic.

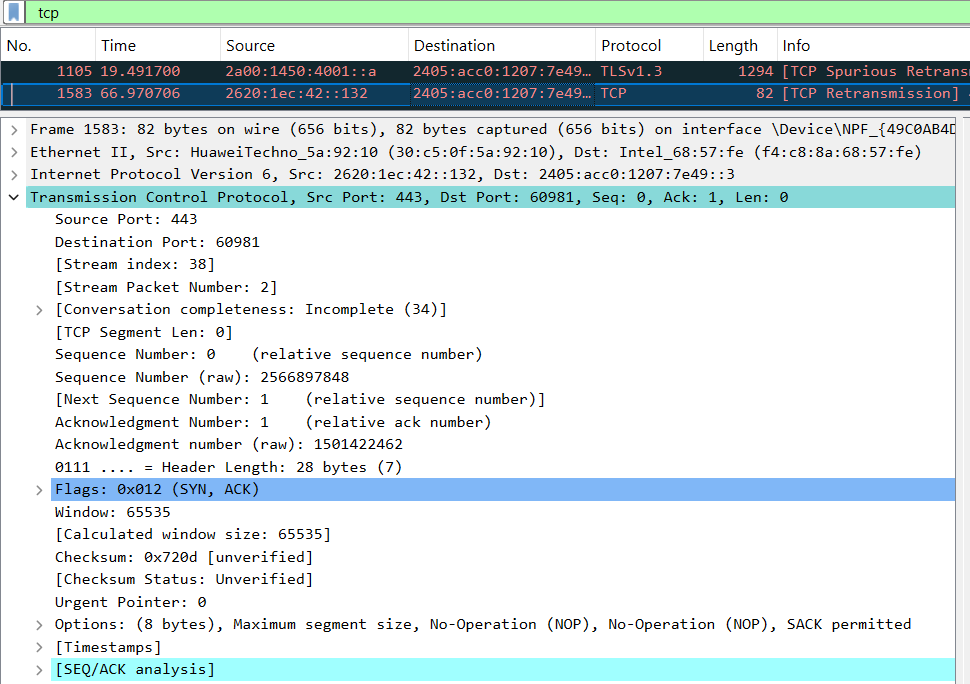


*Fig: Traffic filtering*

**TCP Header Analysis**

After capturing TCP traffic, select a TCP packet to view its header details, which include:

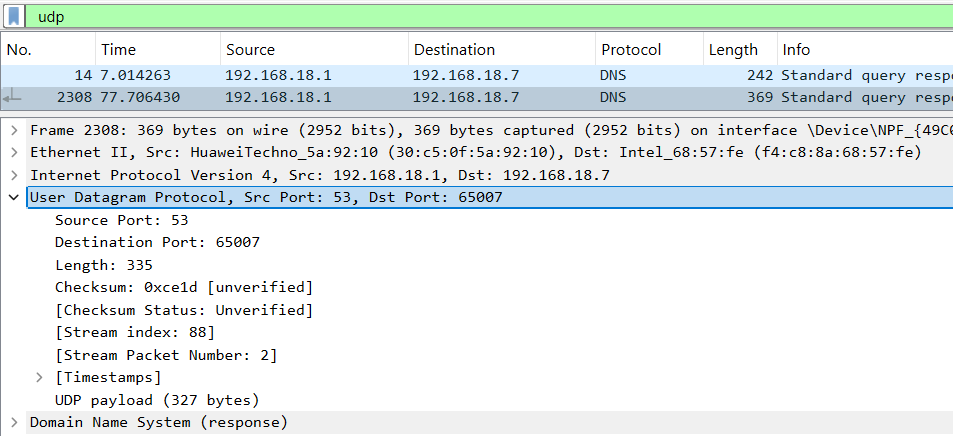
* **Source Port:** Identifies the port on the sender’s machine (e.g., port 443 for HTTPS).
* **Destination Port:** Specifies the port on the recipient’s machine.
* **Sequence Number:** Tracks the packet’s position in the communication stream.
* **Acknowledgment Number:** Confirms the receipt of previous packets.
* **Flags:** Control bits (e.g., SYN, ACK, FIN) used to manage the connection’s state.

*Fig: TCP header analysis of selected packet*

**UDP Header Analysis**

Select a UDP packet and analyze its header details:

* **Source Port:** The port on the sender’s side.
* **Destination Port:** The port on the receiver’s side.
* **Length:** Indicates the size of the UDP packet, including the header and data.
* **Checksum:** A verification field for ensuring data integrity.

*Fig:UDP header analysis of selected packet*

#### ****IP Header Analysis****

For IP packet analysis, the following fields are important:

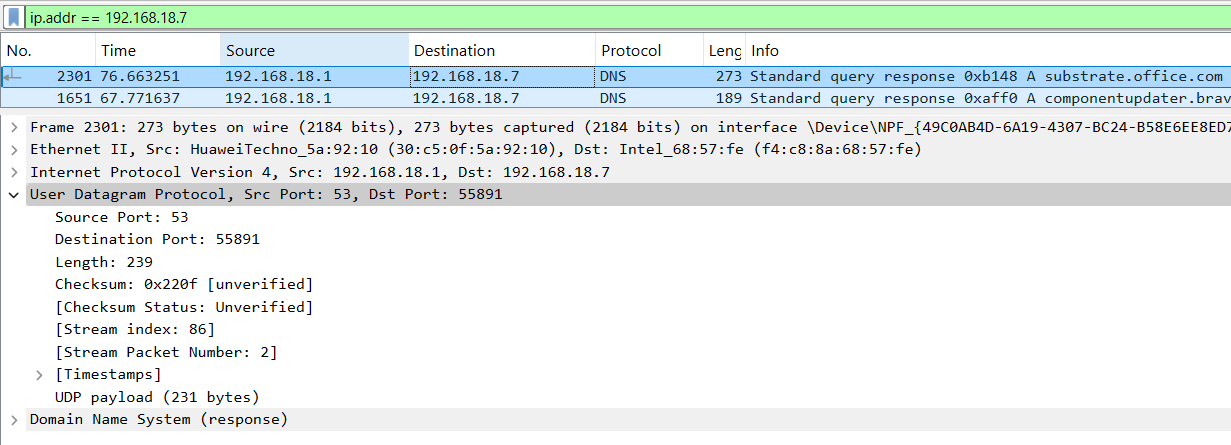
**Source IP:** The sender’s IP address.

**Destination IP:** The receiver’s IP address.

**Header Length:** Indicates the size of the IP header.

**TTL (Time to Live):** Limits the lifespan of the packet, decremented by each router.

**Protocol:** Specifies whether TCP, UDP, or another protocol is being used.

**Fragmentation:** Shows relevant information if the packet is fragmented.

*Fig:IP header analysis of selected packet*

**Conclusion**

In this lab, we delved into Packet Capture and Header Analysis using Wireshark, focusing on key protocols: TCP, UDP, and IP. By capturing live network traffic and analyzing packet headers, we gained practical insights into data transmission and network management. This hands-on experience is vital for understanding protocol operations, troubleshooting network issues, optimizing performance, and enhancing security by examining packet-level details such as ports, IP addresses, and control flags.