**Lab Number: 15 Date: 2025/08/30**

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

**Theory:**

1. **Wireshark:** Wireshark is a powerful open-source network protocol analyzer that enables real-time capture and analysis of network traffic. Wireshark helps identify issues such as network congestion, security vulnerabilities, misconfigurations, and protocol errors. By allowing users to visualize traffic patterns and analyze the behavior of network protocols, it aids in diagnosing problems efficiently and ensuring optimal network performance. Its ability to capture and decode a wide range of protocols makes it invaluable in both network administration and cybersecurity.
2. **TCP, UDP, IP**

**TCP (Transmission Control Protocol)** is a connection-oriented protocol designed to ensure reliable data transmission between devices. It establishes a connection before data transfer begins, incorporates error checking mechanisms, and guarantees that data packets are delivered in sequence and without duplication. TCP is widely utilized in applications such as web browsing (HTTP/HTTPS) and email.

**UDP (User Datagram Protocol)** is a connectionless protocol that prioritizes speed over reliability. It transmits 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 some data loss is tolerable.

**Internet Protocol (IP)** is the primary protocol for routing packets across networks. Operating at the network layer, it assigns IP addresses to devices and ensures that data packets reach their destinations through routing, working alongside both TCP and UDP for end-to-end communication over the Internet.

**Implementation**

1. **Capture the Network Interface of Choice and Filter the traffic**

**Steps:**

1. Open Wireshark and select the network interface (Wi-Fi or Ethernet) where traffic is to be captured.
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, or ip for general IP traffic.
4. **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:** Keeps track of the packet’s position in the communication stream.

**Acknowledgement Number:** Confirms the receipt of previous packets.

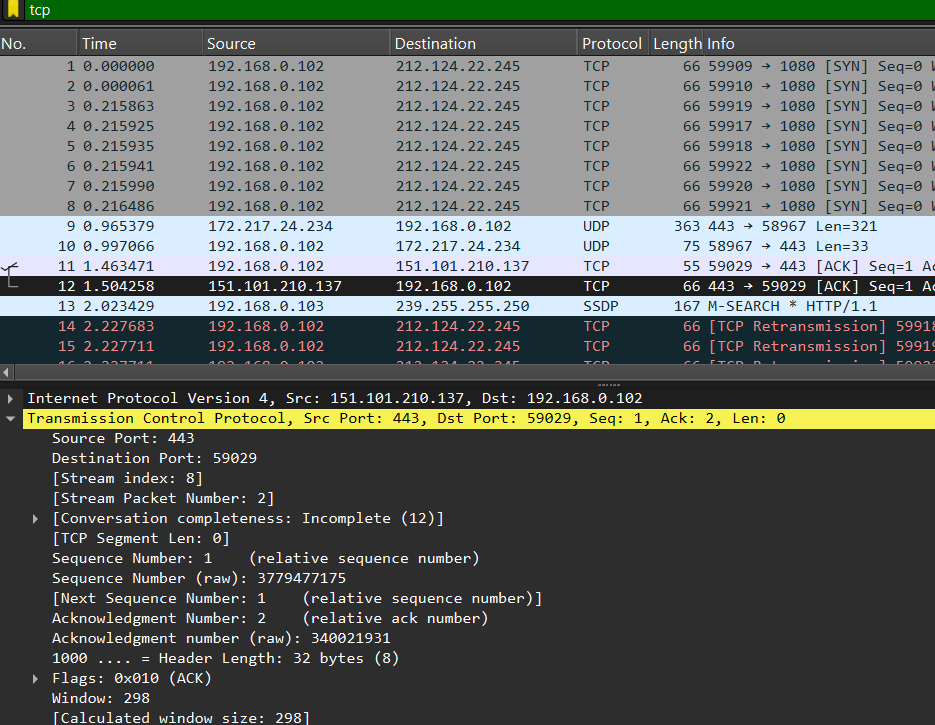
**Flags:** Control bits (e.g., SYN, ACK, FIN) used to manage the connection’s state.

Fig: TCP header analysis of selected packet

**TCP Header Analysis Result:**

From above figure of TCP header analysis, we can deduce the following details for the website youtube.com

|  |  |  |
| --- | --- | --- |
| **S.N.** | **Parameters** | **Details** |
| 1 | Source Port | 443 |
| 2 | Destination Port | 59029 |
| 3 | Sequence Number | 1 |
| 4 | Acknowledgement Number | 2 |
| 5 | Flags | ACK |

Fig: TCP header analysis details table

1. **UDP Header Analysis**

Select a UDP packet and analyze its header:

**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.

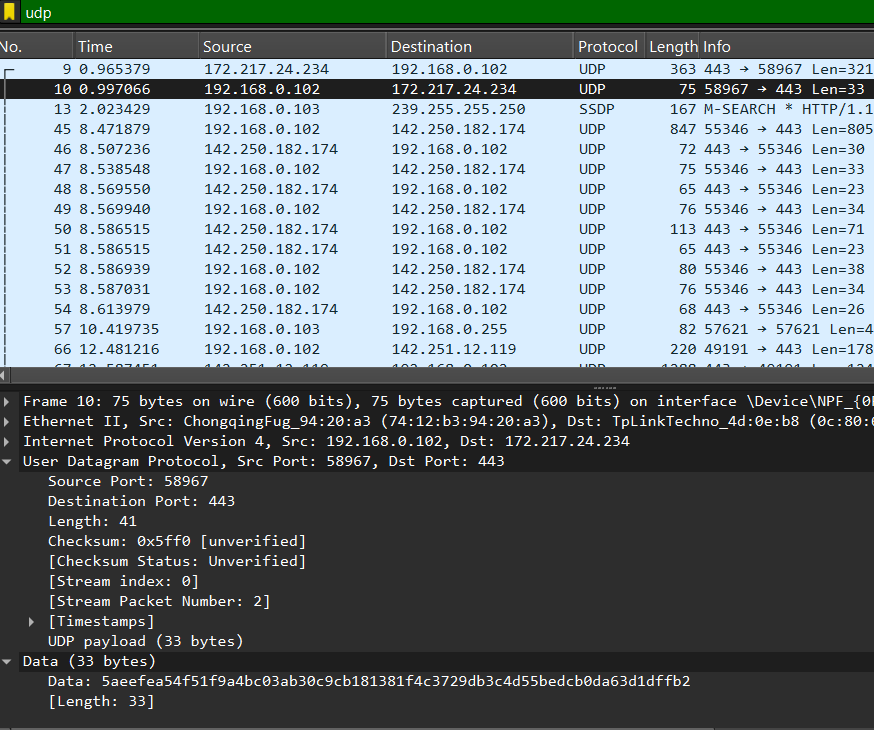
**Checksum:** A verification field for ensuring data integer.

Fig: UDP header analysis of selected packet

**UDP Header Analysis Result:**

From above figure of UDP header analysis, we can deduce the following details of the website youtube.com

|  |  |  |
| --- | --- | --- |
| **SN** | Parameters | Details |
| 1 | Source Port | 58967 |
| 2 | Destination Port | 443 |
| 3 | Length | 41 |
| 4 | Checksum | 0x5ff0 |
| 5 | Stream Index | 0 |
| 6 | Stream Packet Number | 2 |

Fig; UDP header analysis details table

1. **IP Header Analysis**

**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.

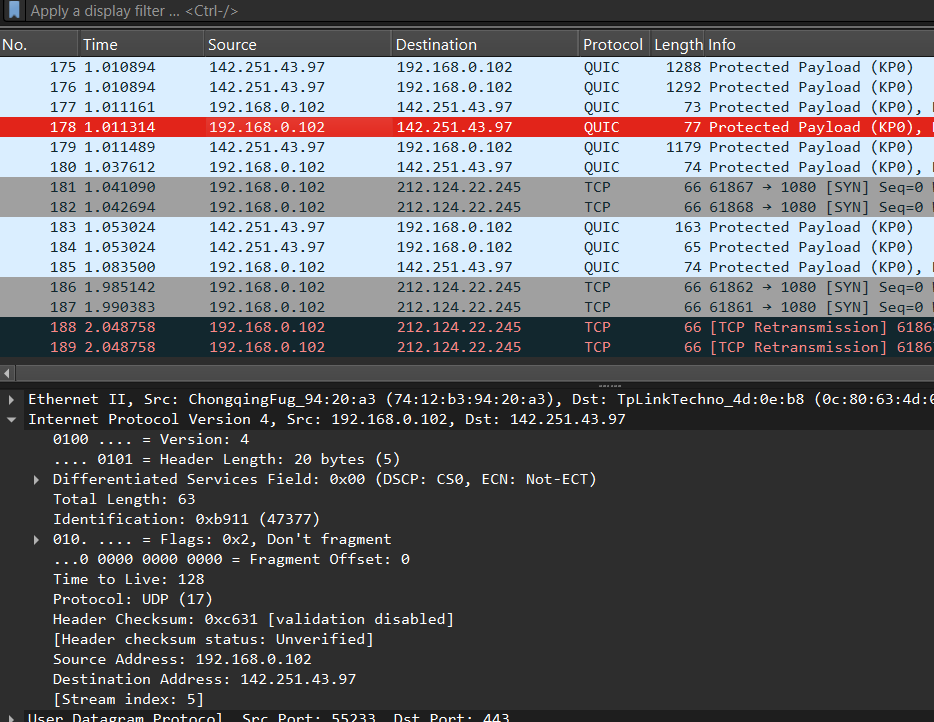
**Protocol:** Specifies whether TCP, UDP, or another protocol is being used.

Fig: IP header analysis of selected packet

**IP Header Analysis Result:**

From above figure of IP header analysis, we can deduce the following details for the website youtube.com

|  |  |  |
| --- | --- | --- |
| **SN** | **Parameters** | **Details** |
| 1 | Source IP | 192.168.0.102 |
| 2 | Destination IP | 142.251.43.97 |
| 3 | Header Length | 20 bytes |
| 4 | TTL (Time to live) | 128 |
| 5 | Protocol | UDP |

Fig: IP header analysis details table

**Conclusion**

In conclusion, this lab provided an in-depth exploration of Packet Capture and Header Analysis using Wireshark, concentrating on essential protocols such as TCP, UDP, and IP. By capturing live network traffic and analyzing packet headers, we gained valuable insights into the mechanisms of data transmission and management across networks. This hands-on experience is crucial for understanding the functionality of different protocols, allowing us to effectively troubleshoot network issues, and enhance security through careful examination of packet-level details, including ports, IP addresses, and control flags.