TCP Packet Analysis Report Using Wireshark

Introduction

In this experiment, Wireshark was used to capture and analyze live network traffic over a Wi-Fi connection. The focus of the analysis was on understanding TCP behavior by observing TCP flags such as SYN, ACK, RST, and retransmissions. These flags play an important role in connection establishment, termination, and error handling. The captured packets were analyzed using display filters to clearly observe TCP connection attempts, resets, and retransmission events.

Objective

The objective of this analysis was to study TCP connection behavior by:

- Observing TCP three-way handshake packets
- Identifying TCP reset (RST) packets
- Analyzing TCP retransmissions
- Understanding why these events occur during normal network communication

---

Methodology

Live network traffic was captured using Wireshark on the active Wi-Fi interface. Display filters were applied to isolate specific TCP flags and TCP analysis events. The packet list, packet details, and packet bytes sections were used together to understand the flow and behavior of TCP connections. Screenshots were taken during analysis to record observations.

Analysis and Observations

1. TCP SYN Packets (Connection Initiation)

Using the display filter:

tcp.flags.syn == 1

TCP SYN packets were observed in large numbers. These packets indicate that the client system was attempting to initiate TCP connections with different destination servers, mostly on port **443 (HTTPS)** and **53 (DNS over TCP)**.

In several cases, SYN packets were followed by SYN-ACK responses from the destination server, confirming that the connection request was received. This

represents the **first and second steps of the TCP three-way handshake**, showing normal connection establishment behavior.

---

2. TCP SYN Retransmissions

In the packet list, multiple entries were marked as:

[TCP Retransmission] [SYN]

This indicates that the client retransmitted SYN packets because a response was not received within the expected time. These retransmissions usually occur due to:

- Network latency
- Temporary packet loss
- Server delay or congestion
- Firewall or filtering behavior

This behavior is **normal in real-world networks** and shows TCP's reliability mechanism, where packets are resent to ensure delivery.

---

3. TCP Reset (RST) Packets

Using the display filter:

tcp.flags.reset == 1

Multiple TCP RST packets were observed. These packets indicate that an existing or attempted TCP connection was **forcefully terminated**.

From the analysis, RST packets were commonly seen when:

- The server closed the connection abruptly
- The connection was no longer required
- The destination port or service was unavailable
- The connection was rejected for security reasons

RST packets were often sent after unsuccessful connection attempts or when encrypted HTTPS sessions were terminated quickly.

---

4. TCP Analysis Flags

Using the filter:

tcp.analysis.flags

Wireshark highlighted several TCP issues such as:

- TCP retransmissions
- Keep-alive packets
- Duplicate acknowledgments

TCP keep-alive packets were observed, which are used to check whether an idle connection is still active. This confirms that long-lived connections were maintained during the capture.

Retransmission warnings further confirmed packet loss or delayed acknowledgments, which is common in wireless networks.

---

5. Encrypted HTTPS Traffic

Most TCP connections were established on **port 443**, indicating HTTPS traffic. The packet payload appeared unreadable, confirming that the communication was encrypted using TLS. Only TCP metadata such as sequence numbers, acknowledgment numbers, and flags were visible.

This confirms that secure communication was taking place and sensitive data was protected.