# HTTP and TCP Basics

Ali Bahja

15th november 2024

# 1 Objective

The purpose of this lab is to analyze the **HTTP protocol**, its encapsulation within **TCP/IP**, and the behavior of TCP mechanisms such as the three-way handshake, acknowledgments, retransmissions, and congestion control.

# 2 Topology and Setup

The topology consists of a client communicating with a web server using HTTP over TCP. Wireshark is used to capture and analyze packets exchanged during :

— Correct HTTP requests.
— Incorrect requests (wrong URL, wrong port).
— HTTPS requests to a server not configured for TLS.

# 3 Procedure and Results

## 3.1 HTTP Communication

A standard HTTP GET request is sent to the server. The response contains headers such as :

— Status line : `HTTP/1.1 200 OK`
— Server date and type
— Content length

The packet structure follows Ethernet → IP → TCP → HTTP layers.

## 3.2 Incorrect Requests

— Wrong URL : no valid response is returned.
— Wrong port : a SYN is sent but no SYN-ACK is received.
— HTTPS : the server refuses the connection since it only listens on port 80.

## 3.3 TCP Mechanisms

The three-way handshake is observed (SYN, SYN-ACK, ACK) before data transmission. Wireshark displays relative sequence and acknowledgment numbers. **Congestion control** and **slow start** appear in the TCP flow graph.

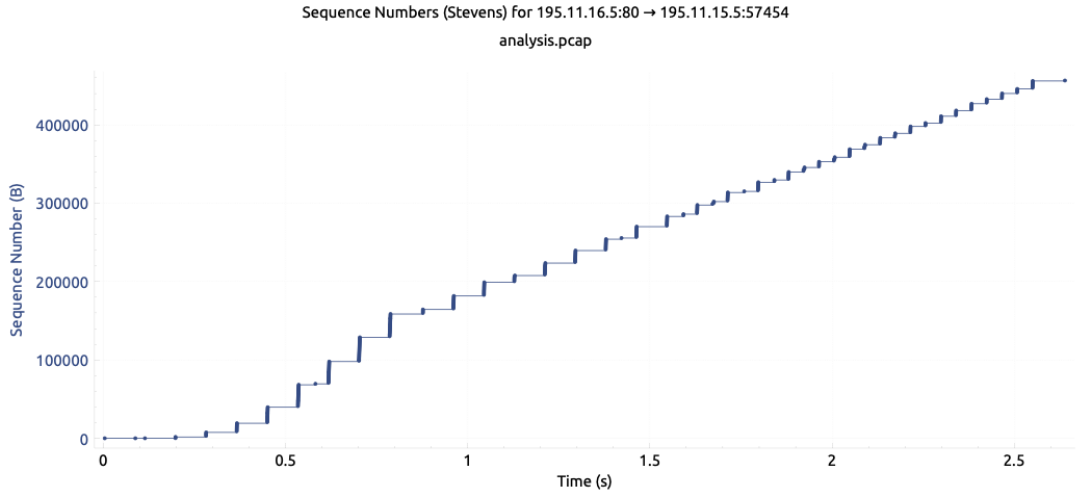


Figure 1 – Wireshark timesequence

# 4 Analysis

HTTP is shown to be a plaintext protocol encapsulated in TCP/IP. TCP ensures reliable delivery via acknowledgments, retransmissions, and congestion management. Incorrect requests highlight the importance of correct addressing and port usage.

# 5 Conclusion

This lab demonstrates how HTTP operates on top of TCP, how TCP ensures reliability through the handshake and acknowledgments, and how congestion control mechanisms function. It also shows the limitations of plaintext HTTP and its dependency on correct addressing and ports.