CS471 Lab1

Mohammed Saad

412112496

Table of Contents

[Part 1: Capturing HTTP Traffic 3](#_Toc189512478)

[*Task 1: Start Wireshark and Capture Packets* 3](#_Toc189512479)

[*Task 2: Filter HTTP Packets and Analyze Them* 3](#_Toc189512480)

[Part 2: Analyzing TCP/IP Traffic 3](#_Toc189512481)

[*Task 1: Filter TCP Packets* 3](#_Toc189512482)

[*Task 2: Analyze TCP Handshake and Investigate Data Transfer and Termination* 3](#_Toc189512483)

[Part 4: Comparing TCP and UDP 4](#_Toc189512484)

[*Task 1: Fill in the table and provide reasons* 4](#_Toc189512485)

[Task 2: Identify Use Cases and Performance of TCP and UDP 4](#_Toc189512486)

# **Part 1: Capturing HTTP Traffic**

## **Task 1: Start Wireshark and Capture Packets**

☑ **Step 1:** Open Wireshark.  
☑ **Step 2:** Select the network interface connected to the internet (e.g., Ethernet or Wi-Fi).  
☑ **Step 3:** Click the "Start Capturing Packets" button (the shark fin icon).  
☑ **Step 4:** Open your favorite web browser and navigate to <http://neverssl.com/>.  
☑ **Step 5:** After the website has fully loaded, stop capturing packets by clicking the red stop button in Wireshark.

## **Task 2: Filter HTTP Packets and Analyze Them**

☑ **Step 1:** In the filter bar, type http and press Enter to show only HTTP packets.  
☑ **Step 2:** Select any HTTP packet to view its details.  
☑ **Step 3:** Observe the HTTP request and response messages. Note the method (GET, POST), URL, response codes (200 OK, 404 Not Found), etc.

# **Part 2: Analyzing TCP/IP Traffic**

## **Task 1: Filter TCP Packets**

☑ **Step 1:** Clear the previous filter and type TCP to focus on TCP packets.  
☑ **Step 2:** Select a TCP packet related to your HTTP request/response.  
☑ **Step 3:** Right-click on the packet and select **"Follow" → "TCP Stream"**.  
☑ **Step 4:** This displays the entire conversation between the client and server.

## **Task 2: Analyze TCP Handshake and Investigate Data Transfer and Termination**

☑ **Step 1:** Find and select packets related to the TCP **three-way handshake**:

**SYN**: Initiates a connection.

**SYN-ACK**: Acknowledges and responds to the SYN.

**ACK**: Acknowledges the SYN-ACK and establishes the connection.

☑ **Step 2:** Note the **sequence and acknowledgment numbers**. Screenshot and upload the image to your [online Git repository](https://github.com/MAlharbi-dev/CS471_Lab/blob/main/Task2Step2.png).  
☑ **Step 3:** Observe the **data packets exchanged** between the client and server. Take a screenshot and upload it to your [online Git repo.](https://github.com/MAlharbi-dev/CS471_Lab/blob/main/Task2Step3.png)  
☑ **Step 4:** Look at the **TCP termination process** (FIN, ACK packets).

# Part 4: Comparing TCP and UDP

## **Task 1: Fill in the table and provide reasons**

| **TCP or UDP** | **Reasons** |
| --- | --- |
| **Reliability and Connection Establishment** | **TCP** because it uses a three-way handshake (SYN, SYN-ACK, ACK) to establish a reliable connection before data transfer. UDP is connectionless and does not guarantee delivery. |
| **Data Integrity and Ordering** | **TCP** because it ensures data arrives in order and without errors using sequence numbers and acknowledgments. UDP does not provide built-in ordering or error correction, making it faster but less reliable. |

## **Task 2: Identify Use Cases and Performance of TCP and UDP**

| **TCP** | **UDP** |
| --- | --- |
| **Use Cases** | **Use Cases** |
| - Web browsing (HTTP, HTTPS) | - Streaming video/audio (YouTube, Netflix) |
| - Email (SMTP, IMAP, POP3) | - Online gaming (real-time updates) |
| - File transfers (FTP, SFTP) | - VoIP calls (Zoom, Skype, WhatsApp calls) |
| - Secure connections (SSH, VPN) | - DNS queries (fast lookups) |
| **Performance** | **Performance** |
| - **Reliable but slower** due to error checking and retransmissions. | - **Faster but less reliable** since there is no retransmission mechanism. |
| - Ensures **data integrity and order**. | - Best for **real-time applications** where speed is more important than reliability. |