جامعة جدة
University of Jeddah

College of Science and Computer Engineering

Department of Computer and Network Engineering

# CCCN312: Computer Networks

## Computer Networks Lab Project:
## Network Monitoring

**Prepared by:**

Lina Bashawyah

Lama Al-Ghamdi

Fall 2024

جامعة جدة
University of Jeddah

# Project Outline

- Packet Capture and Parsing.
- Logging System.
- Throughput Calculation.
- Latency Measurement.
- Multi-Connection Management.
- Network Metrics Calculation.
- Real-Time Statistics Display and Analysis.
- Results Visualization.
- Graceful Termination and Final Statistics.

جامعة جدة
University of Jeddah

# Introduction

This is a project that monitors and analyzes network traffic using Python. This project displays network statistics such as the number of connections, the average size of the packet, and the number of unique IP/MAC addresses. Also, show graphs of network traffic every 30 seconds such as Throughput Over Time, Latency Average and Protocol Usage. By using Python libraries such as Scapy for packet capture and matplotlib for data visualization, the system provides insights into network performance, throughput, latency, and protocol usage. Additionally, implement a TCP server-client model to support multiple concurrent connections, ensuring robustness and concurrency. The system is designed to run indefinitely until manually terminated (e.g., via Ctrl+C).

## Functions Used

- **handle_client_connection(client_socket):**
  Handles communication with a connected client.

- **start_tcp_server():**
  Initializes a TCP socket, listens for connections, and spawns threads for each client.

- **log_event(protocol, src_addr, dest_addr, message_size, ports=None, flags=None):**
  Logging system to store network events to a log file.

- **update_event_data(protocol, src_addr, dest_addr, message_size, timestamp):**
  Updates data structures with packet metadata.

- **process_packet(packet):**
  Extracts protocol-specific information and logs events.

- **start_sniffing():**
  Captures packets using Scapy and processes them via the process_packet function.

- **cal_and_plot_latency():**
  Computes latency based on timestamps and plots latency average for all protocols.

- **cal_and_plot_throughput(interval=10):**
  Calculates throughput over a specified interval and plot it.

- **plot_protocol_usage():**
  Visualizes the usage of different protocols and the count of unique IP/MAC addresses.

- **display_statistics():**
  Summarizes key statistics like packet counts, average packet sizes, and unique address count.

- **main():**
  Organize all components, including starting threads, displaying statistics, and handling user termination.

# Results & Discussion

At the start of running the code the server start listening on local host and sniffing.

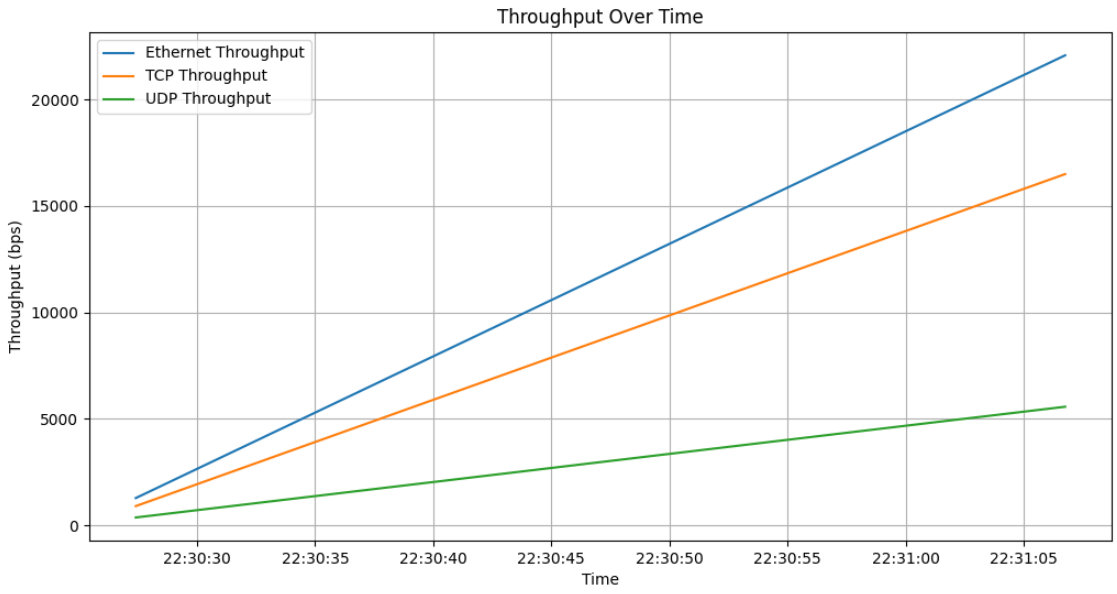
```
(venv) PS C:\Users\Lenovo\PycharmProjects\pythonProject2> python NetworkProject.py
Starting packet sniffing...

Server listening on 127.0.0.1:5000
```

Then the system produces several outputs:

**1. Throughput Over Time**

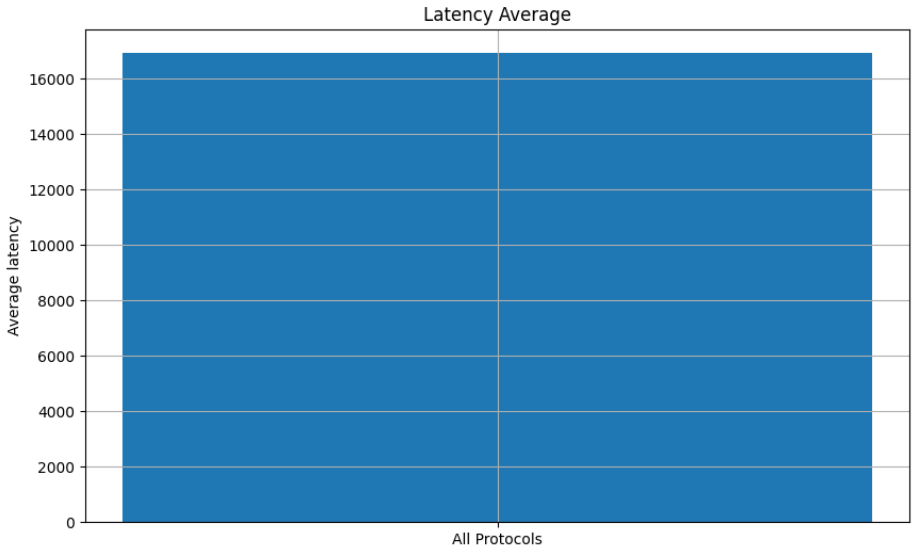
A line graph shows data throughput for Ethernet, TCP, and UDP protocols. For example: the throughput for ethernet is the largest then throughput for TCP and last is throughput for UDP.



**2. Latency Distribution**

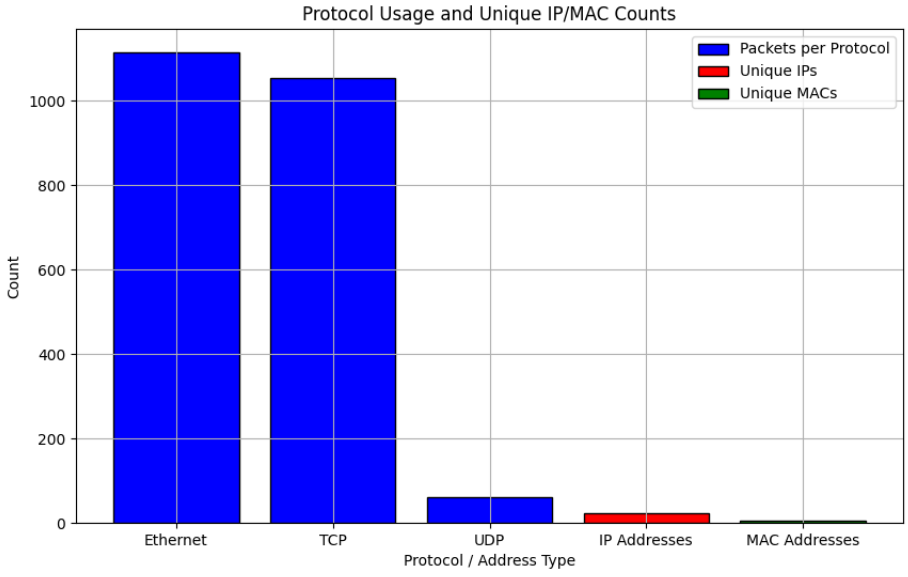
A graph that visualizes the latency of network connections:

We notice that the average latency of all protocols is more than 16000ms which is a large latency.




**3. Protocol Usage**

A bar chart highlights the number of packets for each protocol and unique IP/MAC addresses:



**4. Console Output**

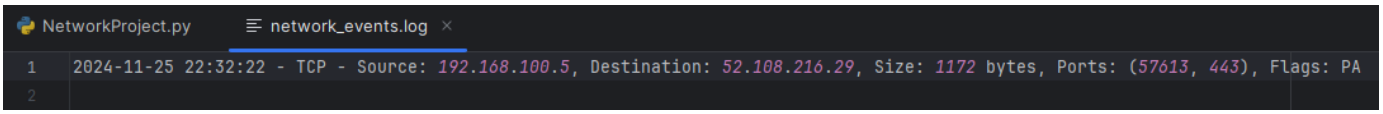
Statistics such as connection counts, average packet sizes, and rates of new connections are displayed every 30 seconds:

```
--- Network Statistics ---
Ethernet Connections: 87, Average Size: 335.55 bytes
TCP Connections: 66, Average Size: 329.62 bytes
UDP Connections: 21, Average Size: 354.19 bytes
Rate of New Connections (Ethernet): 52 connections
Rate of New Connections (TCP): 4 connections
Rate of New Connections (UDP): 3 connections
Unique IP Addresses: 10
Unique MAC Addresses: 5
---------------------------
```

Also, Throughput calculation for each protocol:

```
------------------------
Throughput (bps):
Ethernet: 22066.40 bps
TCP: 16492.80 bps
UDP: 5573.60 bps
------------------------
```

And the log file include:



Finally, when we press Ctrl+c the system shows the final statistics and terminates the program. This statistic is like the statistics that display every 30 seconds.

```
Ctrl+C detected!
 ****  This is the final statistics   ****

--- Network Statistics ---
Ethernet Connections: 1184, Average Size: 494.20 bytes
TCP Connections: 946, Average Size: 581.34 bytes
UDP Connections: 237, Average Size: 148.24 bytes
Rate of New Connections (Ethernet): 460 connections
Rate of New Connections (UDP): 1 connections
Rate of New Connections (TCP): 6 connections
Unique IP Addresses: 25
Unique MAC Addresses: 4
--------------------------


Shutting down...
Program terminated gracefully.
(venv) PS C:\Users\Lenovo\PycharmProjects\pythonProject2>
```

## Conclusion

This network monitoring and analysis system effectively captures and processes network traffic, calculates performance metrics, and visualizes key data trends. The results provide valuable insights into network behavior, aiding in diagnostics and optimization.