Lab 05: Wireless Signal Measurements

Release Date: 18th February, 2025 (Tuesday Section) & 20th February, 2025(Thursday Section)
Demo and Submission Date: 25th February, 2025(Tuesday Section) & 27th February, 2025(Thursday Section)

Deliverables: **Part 1:**Bar chart image and main.py, **Part 2**: 5 images for TCP, UDP throughputs at 5 different distances, iperf_plot.py(plotting script)

Introduction

In this lab, you'll measure the WiFi signal strength at different locations in your house (or whatever building you're working in) to see how signal strength changes as you move away from the router, around barriers, etc.

Code:

Go to the lab5-demo in Google Drive and download to your PC.

Part One - Wireless Signal Measurements

You will need to do this lab on your PC **(not your VM or Rpi)**. First, you'll need some python packages that don't come with the standard library. These requirements are listed in the requirements.txt file in the repository. To install them, you can run

```
pip install -r requirements.txt
```

Open main.py and read through the comments, documentation, and implemented code. Answer all the questions in the comments (put the answers in your README.md file, as usual) and implement the TODOs.

Experiment Setup

1. Identify at least **five different locations** where you will measure WiFi signal strength.
2. At each location, collect **ten samples** of the signal strength.
3. Compute the **mean and standard deviation** of the collected signal strengths.
4. Visualize the results in a bar chart with error bars.

Function: Get WiFi Signal Strength

- This function retrieves the WiFi signal strength in **dBm**.
- Uses OS-specific commands to extract signal strength.
- On **Windows**, the percentage quality is converted to dBm using `-100 + quality/2`.

Function: Main Execution

- Prompts the user to go to different locations and press enter to start sampling.
- Records **10 samples** of signal strength at each location.
- Computes **mean and standard deviation**.
- Uses **Plotly** to create a bar chart with error bars.

There are a total of 9 questions in the main.py file which you'd need to include in your Readme files.

Part 2: Measuring Network Performance Using iPerf3

Measuring TCP and UDP Performance with iPerf3

(Main shell script to generate csv for TCP UDP throughput tests)

Objective

In this lab, students will:

✅ Set up **network performance measurement** using iPerf3.
✅ Compare **TCP vs. UDP throughput** and **packet loss**. Throughput here refers to the Actual speed of **successful data transfer.**
✅ Understand how **distance affects Wi-Fi performance**.
✅ Analyze and visualize **logged data**.

Wi-Fi networks **adapt their transmission rates based on signal quality**. As devices move further from the router, the **signal strength decreases**, affecting **throughput, reliability, and packet loss**.

🔍 **What We Are Testing (Here we will use Laptop VM & RPi)**

1. **We keep the laptop VM stationary near the router.**
2. **We move the Raspberry Pi to 5 different distances from the router** (e.g., 2m, 4m, 6m, 7m, 8m) while sending dummy network packets to the server on VM.
    a. **What is Being Sent from Raspberry Pi (Sender) to VM (Receiver)?**
        i. When running **iPerf3**, the **Raspberry Pi (client/sender)** sends **dummy network traffic** (not actual files or meaningful data) to the **VM (server/receiver)**. The goal is to **measure network**

performance, not transfer real content.

3. **We measure how throughput changes** in TCP and UDP tests.

Expected Observations

Distance	Expected TCP Behavior	Expected UDP Behavior
2m, 5m (Near Router)	High throughput, minimal packet loss	High throughput, near-zero packet loss
10m (Moderate Distance)	Reduced throughput due to lower signal strength	Some packet loss starts appearing
15m+ (Weak Signal)	Unstable throughput, increased latency	Significant packet loss, potential disconnections

Key Learning Points

- **TCP adapts to weak signals by slowing down** and retransmitting lost packets.
- **UDP does not retransmit**, so packet loss increases significantly at larger distances.
- **Wi-Fi degrades with range**, and students will observe this using real experiments.

1.1 Required Equipment

- **VM (Ubuntu/Linux)** → Acts as the **iPerf3 server**.
- **Raspberry Pi (Linux)** → Acts as the **iPerf3 client**.
- **Wi-Fi Router** → Connect both devices to the **same network**.
- **Power Banks** for wirelessly powering the Rpi.(its cheap and also please share if needed)

1.2 Network Configuration

1. Connect both the **Raspberry Pi** and Laptop **VM** to the same **Wi-Fi network**.
2. Find their IP addresses using:

```
hostname -I
```

Ensure VM and Rpi can see each other using ping:

From VM: `ping -c 5 <RPi_ip>`

From RPi: `ping -c 5 <VM_ip>`

Expected output: 64 bytes from 192.168.1.100: time=2.5 ms

2. Install iPerf3

Run the following command **on both devices**:

```
sudo apt update
```

```
sudo apt install iperf3 -y
```

3. Start the iPerf3 Server (On Laptop VM)

On the **VM**, run:

```
iperf3 -s
```

This starts a **TCP & UDP server**, listening on **port 5201**.

To disable Firewalls from blocking TCP and UDP traffic, do:

```
sudo ufw disable
```

4. Running TCP & UDP Tests from Raspberry Pi

4.1 Running a TCP Test (One-Time)

On the **Raspberry Pi**, run:

```
iperf3 -c <VM IP(iperf3 server)> -t 10
```

Expected output:

```
[  5]   0.00-10.00  sec  112 MBytes  93.5 Mbits/sec  sender
[  5]   0.00-10.00  sec  111 MBytes  92.8 Mbits/sec  receiver
```

Key Observations:

- **Sender throughput** = How fast the Raspberry Pi sent data.
- **Receiver throughput** = How much data the VM received

4.2 Running a UDP Test (One-Time)

On the **Raspberry Pi**, run:

```
iperf3 -c <VM IP(iperf3 server) -u -b 10M -t 10
```

5. Automating TCP & UDP Tests with a Script(Script in Drive, implement in RPi with 5 distances)

Now, we will use a **Bash script to automate**:

✅ **Run 5 TCP tests, compute the average, and log the data**.
✅ **Run 5 UDP tests, compute the average, and log the data**.
✅ **Print results to the terminal** for verification.

5.1 Create & Edit `iperf_logger.sh`

On the **Raspberry Pi**, create the script:

```
nano iperf_logger.sh
```

Copy the code from the file in the drive or SCP to transfer the file from local to RPi (SCP covered before)

This script is already given in your drive folder such that you don't have to write the shell script from scratch. Just run it on RPi as is.

To run the bash file,

`chmod +x iperf_logger.sh` **(this command adds the execute (x) permission to a bash file)**

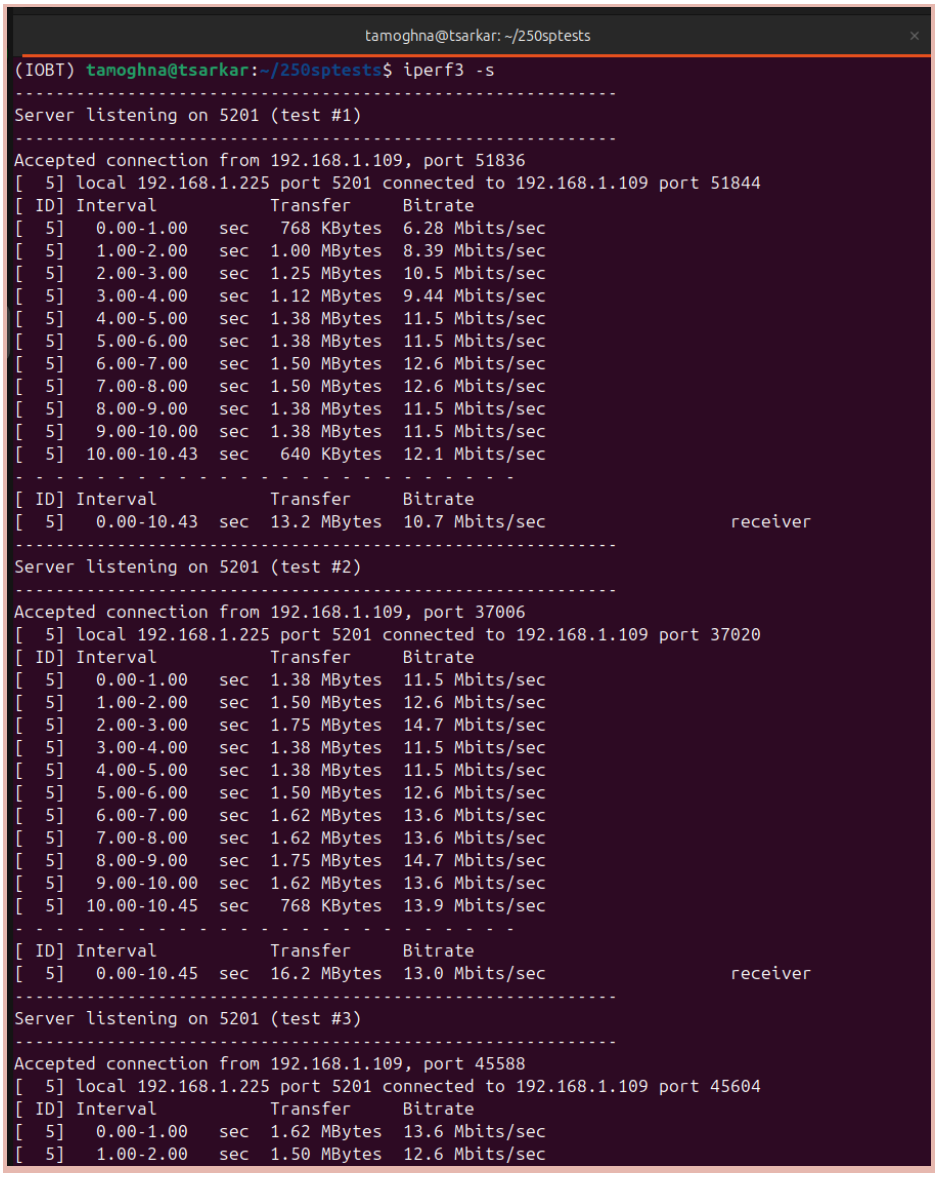
```
./iperf_logger.sh <distance_Args>
```

For example, the laptop VM is fixed near the router and rpi is at 2m distance,

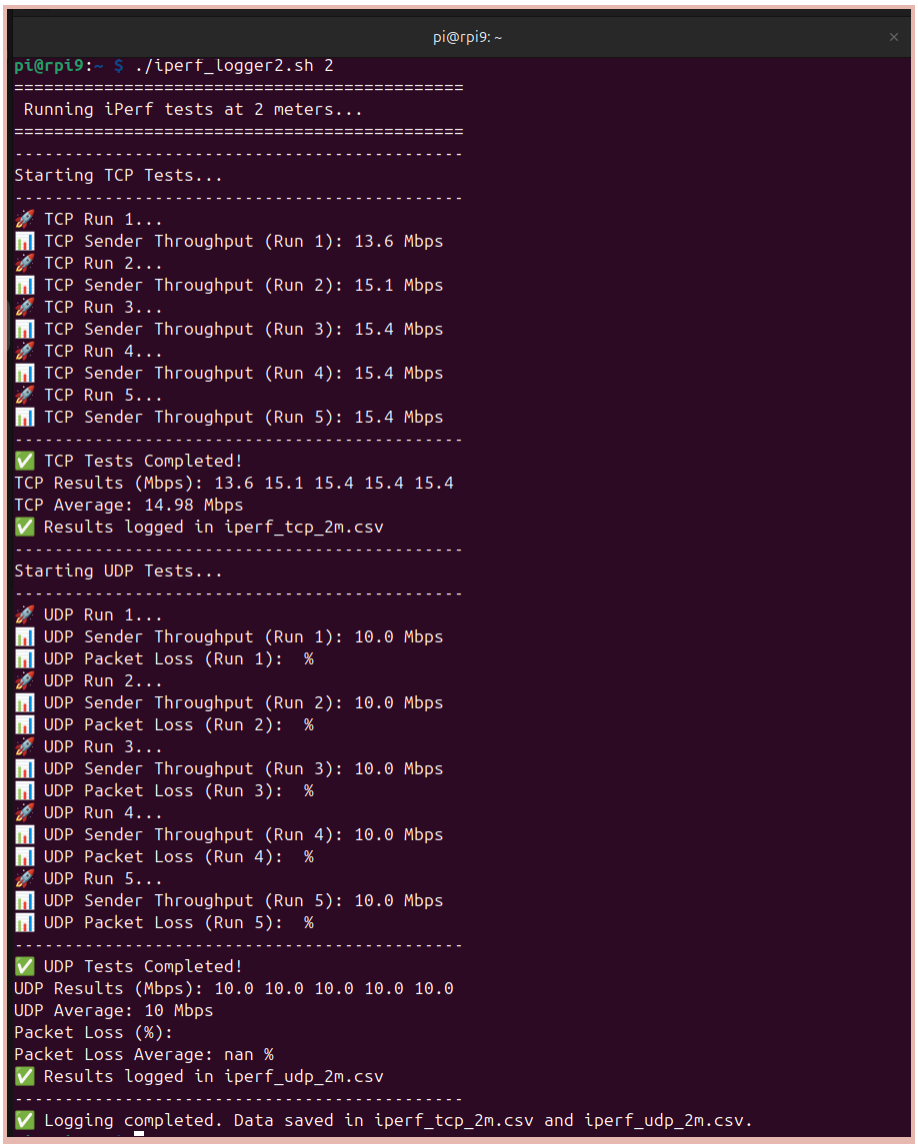
```
./iperf_logger.sh 2
```

```
./iperf_logger.sh 4
```

And so on as you move ahead. Both TCP and UDP throughput tests are done in the same script, the only change is the distances, logs are stored in the CSVs with respective names of protocols as well as distances for plotting.



On the Iperf3 server-side implementation (Laptop VM fixed near router)



On the RPi Client side (varying distances)

6. Logging and Analyzing Results

After successfully running the **iPerf tests** at different distances, the Raspberry Pi will have generated **CSV log files** containing the throughput measurements.

6.1 Locating the Logged Data

The test results will be stored in the following files:

- **TCP Results:** `iperf_tcp_<distance>m.csv`
  📄 Example: `iperf_tcp_2m.csv`, `iperf_tcp_5m.csv`
- **UDP Results:** `iperf_udp_<distance>m.csv`

📄 Example: `iperf_udp_2m.csv`, `iperf_udp_5m.csv`

6.2 Viewing Logged Data

Checking the Latest Entries

To view the last few lines of the logged CSV files:

- **For TCP results** (replace `2m` with the correct distance):

```
tail -n 5 iperf_tcp_2m.csv
```

For UDP results:

```
tail -n 5 iperf_udp_2m.csv
```

Displaying the Full CSV

To see the entire contents of the log file:

```
cat iperf_tcp_2m.csv
```

```
cat iperf_udp_2m.csv
```

6.3 Understanding the CSV File Format

The CSV logs store **multiple test runs** for each distance:

📂 **`iperf_tcp_2m.csv` (Example Data)**

```
Distance, Run1, Run2, Run3, Run4, Run5, Avg
2, 92.3, 90.5, 89.1, 91.2, 93.0, 91.2
5, 85.2, 84.0, 83.5, 82.9, 85.1, 84.1
```

Column 1: Distance from the router (in meters).

Run1 – Run5: Throughput (Mbps) from five test runs.

Avg: Average TCP throughput.

📂 `iperf_udp_2m.csv` (Example Data)

```
Distance, Run1, Run2, Run3, Run4, Run5, Avg, PacketLoss1, PacketLoss2,
PacketLoss3, PacketLoss4, PacketLoss5, Avg_PacketLoss
```

2, 9.8, 9.7, 9.5, 9.6, 9.9, 9.7, 0, 0.2, 0.5, 0.3, 0.1, 0.22

5, 8.4, 8.1, 7.9, 7.8, 8.0, 8.04, 1.2, 1.5, 2.1, 2.3, 1.8, 1.78

UDP throughput (Mbps) and packet loss (%) are logged separately.

6.4 Analyzing Data with Python

To **automate analysis** and generate plots, we will use a Python script.

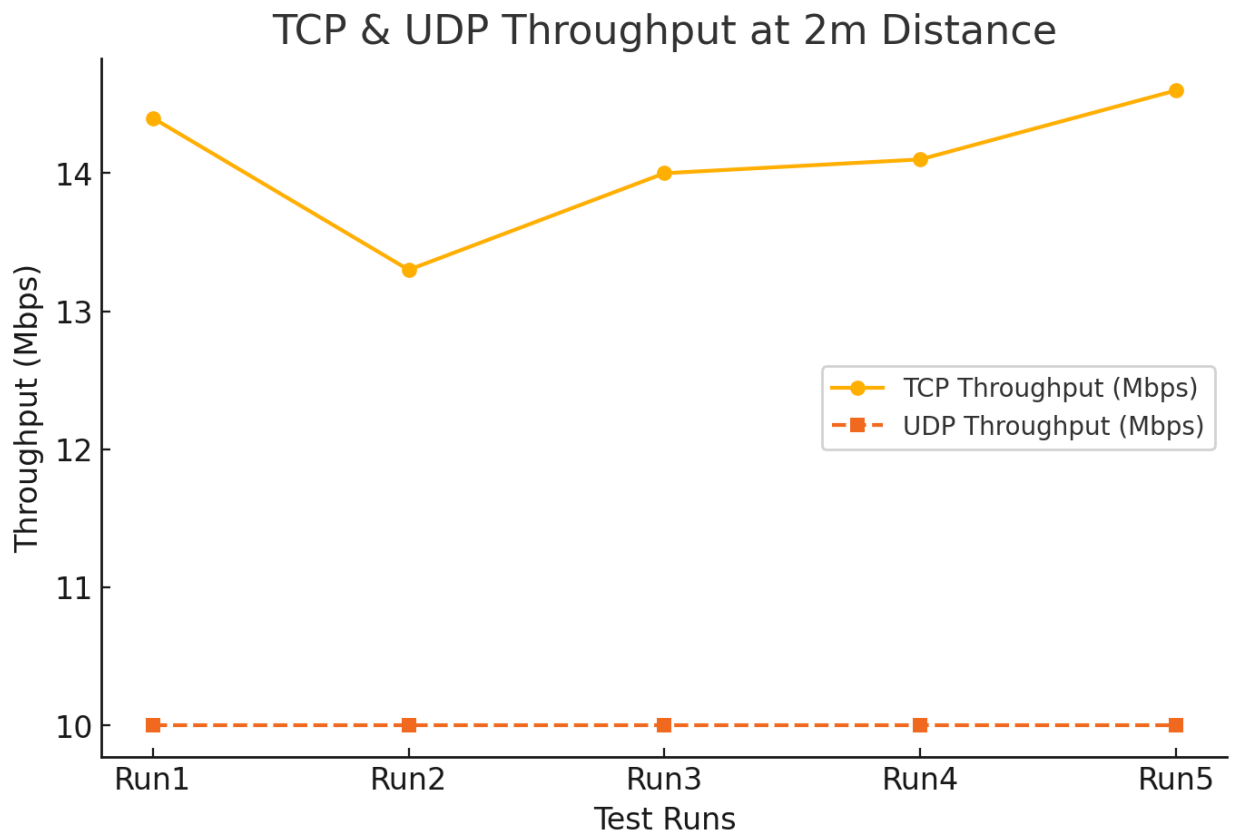
6.4.1 Create `iperf_plot.py`

On the Raspberry Pi, create a new script:

TCP & UDP Throughput vs Distance 📈

- TCP throughput should **gradually decrease** with distance.
- UDP throughput may **drop suddenly** with packet loss.

Example plot for Fixed Laptop VM near router, Rpi at 2metres from laptop(Distance bw Router and Rpi is 2m)



Observations from above plot:

- **TCP throughput fluctuates slightly between 13.3 Mbps and 14.6 Mbps, with an average of 14.08 Mbps.**
- **UDP throughput remains stable at 10 Mbps, showing no packet loss.**

This shows that at close range (2m), the network is stable, with minimal TCP variations and no UDP packet loss.

More Questions from Part 2

After analyzing the data, students should answer the following:

1. **How does distance affect TCP and UDP throughput?**
2. **At what distance does significant packet loss occur for UDP?**
3. **Why does UDP experience more packet loss than TCP?**
4. **What happens if we increase the UDP bandwidth (`-b 100M`)?**
5. **Would performance be different on 5 GHz Wi-Fi vs. 2.4 GHz?**

Total of 14 Questions(9 in Part 1 in script main.py, 5 in Part 2)

Demo and Code Grading Rubric

Points	Description
Demo	
3	Part 1 - Code runs and generates a correct bar graph.
3	Part 2 - Code runs and generates a correct graph for TCP and UDP(throughput vs distance)
Code	Only one submission to the Vocareum is need for each team.
2	Part 1 - Samples collected correctly (10 samples, 1 second delay between each)
2	Part 1 - Signal strength mean computed correctly
2	Part 2 - Iperf3 server implemented correctly

2	Part 2 - Iperf3 Plotting Script
2	Part 2 - Graphs(plot and save the graphs for TCP and UDP from Section 6)
Reflection	Answer Lab Questions in Readme.md
14	Questions (1 point per question)
	Total: 30