Lab WiFi Report

Andrea Botticella[*]
andrea.botticella@studenti.polito.it
Politecnico di Torino
Turin, Italy

Elia Innocenti[*]
elia.innocenti@studenti.polito.it
Politecnico di Torino
Turin, Italy

Renato Mignone[*]
renato.mignone@studenti.polito.it
Politecnico di Torino
Turin, Italy

Simone Romano[*]
simone.romano2@studenti.polito.it
Politecnico di Torino
Turin, Italy

ABSTRACT

This report investigates the performance of wireless and device-to-device communication by comparing theoretical predictions with experimental results in three distinct scenarios: both devices connected via WiFi, both via Ethernet, and a mixed configuration where one device uses Ethernet and the other WiFi. Using tools such as iperf3 and Wireshark, we measured the goodput and analyzed its variability under different network conditions. The experimental results are compared with theoretical calculations based on protocol efficiencies and network overheads. Our findings highlight the stability of Ethernet connections and the challenges posed by WiFi's shared medium and half-duplex constraints. This study provides useful insights for optimizing network configurations in mixed environments.

1 BACKGROUND AND OBJECTIVES

In this laboratory exercise, we aim to evaluate and compare the performance of wired and wireless communication within a local area network. The experiment involves setting up three distinct scenarios: both devices connected via WiFi, both devices connected via Ethernet, and a mixed configuration where one device uses Ethernet and the other uses WiFi.
The main objectives of the lab are to:

- Measure the goodput, i.e., the rate of useful data received at the application layer, using iperf3.
- Analyze the variability and stability of the connection in each scenario by collecting data over multiple test runs.
- Compare the experimental results against theoretical predictions based on protocol efficiencies and network overhead.
- Investigate potential sources of performance degradation in wireless communication, such as interference, half-duplex operation, and shared medium limitations.

Through these experiments, we will gain practical insights into the strengths and limitations of both Ethernet and WiFi communication methods, which are crucial for optimizing network performance in mixed connectivity environments.

2 METHODOLOGY AND CONCEPTS

This section outlines the experimental setup, the tools employed for the measurements, and the theoretical basis for estimating goodput.

[*]The authors collaborated closely in developing this project.

2.1 Selected Tools

To evaluate the performance of both Ethernet and WiFi connections, we utilized several specialized tools:

- **iperf3**: Used to generate traffic and measure goodput in both TCP and UDP modes. By executing repeated tests, iperf3 provides key metrics such as minimum, maximum, average, and standard deviation of the throughput.
- **Wireshark**: Employed to capture and analyze network traffic, Wireshark enabled us to inspect data flows, identify control and data frames, and validate experimental results.
- **Automation Script**: A Python script was developed to automate the entire measurement process. This script manages both server and client modes of iperf3, logs output in JSON, CSV, and plain text formats, and computes summary statistics. The script accepts several command-line flags:
  - **–server**: Launches the iperf3 server in JSON output mode.
  - **<SERVER_IP>**: Specifies the server's IP address when running in client mode.
  - **–udp**: Switches the test from the default TCP mode to UDP.
  - **–bitrate**: Sets the target bitrate for UDP tests (e.g., `10M` for 10 Mbps).
  - **–iter**: Determines the number of test iterations to perform.

  The output files (logs, JSON, CSV reports, and raw output) generated by this script are used to document the experimental results and facilitate further analysis.

2.2 Goodput Estimation

Goodput represents the rate at which useful data is delivered to the application layer, excluding protocol overheads and retransmitted packets. The theoretical estimation of goodput is based on the efficiency of the protocol and the capacity of the network link:

$$G \leq \eta_{\text{protocol}} \times C,$$

where C is the capacity of the bottleneck link and η_{protocol} is the protocol efficiency.
For Ethernet connections, the efficiency for TCP is computed as:

$$\eta_{TCP}^{Eth} = \frac{MSS}{MSS + \text{TCP headers} + \text{IP headers} + \text{Ethernet overhead}},$$

with the Maximum Segment Size (MSS) defined as the MTU minus the headers. For a standard MTU of 1500 bytes, we obtain:

- MSS ≈ 1460 bytes (after subtracting 20 bytes for the IP header and 20 bytes for the TCP header),
- An additional Ethernet overhead of approximately 38 bytes.

Thus, the efficiency for TCP over Ethernet is approximately:

$$\eta_{TCP}^{Eth} \approx \frac{1460}{1460+20+20+38} \approx 94.9\%.$$

Similarly, the efficiency for UDP is computed as follows. Since UDP has an 8-byte header, its MSS is given by:

- MSS $\approx$ 1472 bytes (after subtracting 20 bytes for the IP header and 8 bytes for the UDP header).

Thus, the efficiency for UDP over Ethernet is given by:

$$\eta_{UDP}^{Eth} \approx \frac{1472}{1472+20+8+38} \approx 95.7\%.$$

For WiFi, additional factors must be considered due to its half-duplex nature and the inherent overhead of the 802.11 protocol (e.g., control frames, retransmissions, and channel contention). Consequently, the effective efficiency is reduced by a WiFi-specific factor (η_{WiFi}). The adjusted efficiency for TCP over WiFi can be expressed as:

$$\eta_{TCP}^{WiFi} = \eta_{TCP}^{Eth} \times \eta_{WiFi},$$

with η_{WiFi} typically around 80% in optimal conditions. A similar adjustment applies for UDP.

These theoretical estimates set an upper bound on the achievable goodput, against which our experimental results are compared. Discrepancies between theory and practice are primarily due to dynamic environmental factors, such as interference, channel variability, and the inherent limitations of wireless communication.

3 EXPERIMENTAL SETUP AND TEST CASES

3.1 Equipment and Configuration

In this section, we describe the hardware and software configuration used to perform our network performance measurements. Table 1 summarizes the main devices, their interfaces, and relevant specifications.

Device	Key Specifications
PC1	**Victus 16-s1005nl Notebook** *Operating System:* Ubuntu 24.04.2 LTS *Ethernet Interface:* Realtek RTL8111/8168/8211/8411 *Wireless Interface:* Realtek RTL8852BE (802.11ax)
PC2	**Microsoft Surface Laptop Go 3** *Operating System:* Ubuntu 24.10 *Ethernet Interface:* via Anker PowerExpand+ USB-C Hub *Wireless Interface:* Intel Alder Lake-P CNVi (802.11ax)
Router	**Vodafone Power Station Wi-Fi 6** *Ethernet Ports:* 4 × 1 GbE ports *Wi-Fi:* Dual-band 802.11ax (2.4 GHz / 5 GHz)
Cables	CAT.5E (up to 1 Gbps)

Table 1: Summary of Hardware and Network Configuration

This hardware setup allows us to compare Ethernet versus Wi-Fi performance under a consistent router and cabling environment. In the next section, we detail the evaluation scenarios and the measurement methodology.

3.2 Evaluation Scenarios

We considered three distinct network configurations to assess the performance differences between wired and wireless communications. For each scenario, the theoretical goodput is computed based on the nominal link capacity and protocol efficiency.

1. **Both Ethernet:**
   In this scenario, both PC1 and PC2 are connected to the router via CAT.5E cables, providing a nominal link capacity of 1 Gbps. The efficiency for Ethernet is calculated as follows:

   $$\eta_{TCP}^{Eth} \approx \frac{1460}{1460+20+20+38} \approx 94.9\%,$$

   $$\eta_{UDP}^{Eth} \approx \frac{1472}{1472+20+8+38} \approx 95.7\%.$$

   Thus, the expected goodput is:

   $$G_{TCP}^{Eth} \le 0.949 \times 1000\,\text{Mbps} \approx 949\,\text{Mbps},$$

   $$G_{UDP}^{Eth} \le 0.957 \times 1000\,\text{Mbps} \approx 957\,\text{Mbps}.$$

2. **Both Wi-Fi:**
   For this configuration, both devices use their wireless interfaces (802.11ax) to connect to the router. Although the nominal Wi-Fi link speed is assumed to be approximately 867 Mbps, the half-duplex nature of Wi-Fi effectively halves the throughput available for data transfer. Assuming a Wi-Fi efficiency factor of about 80%, the expected goodput for TCP is:

   $$G_{TCP}^{WiFi} \le 0.80 \times 867\,\text{Mbps} \times \frac{1}{2} \approx 347\,\text{Mbps},$$

   and similarly for UDP:

   $$G_{UDP}^{WiFi} \le 0.806 \times 867\,\text{Mbps} \times \frac{1}{2} \approx 350\,\text{Mbps}.$$

3. **Mixed Scenario:**
   In this configuration, one device (PC1) is connected via Ethernet while the other (PC2) uses its Wi-Fi interface. Here, the bottleneck is the Wi-Fi link; however, since only one device is utilizing Wi-Fi, the throughput is not halved. The expected goodput is then:

   $$G_{TCP}^{Mixed} \le 0.80 \times 867\,\text{Mbps} \approx 694\,\text{Mbps},$$

   $$G_{UDP}^{Mixed} \le 0.806 \times 867\,\text{Mbps} \approx 698\,\text{Mbps}.$$

These calculations provide the theoretical upper bounds for goodput in each scenario. The experimental results, obtained via automated measurements using the provided Python script, are compared against these predictions to evaluate real-world performance.

Note on Experimental Automation:
The experimental process was automated using a custom Python script. This script manages both the server and client modes of iperf3 and handles output logging in JSON, CSV, and plain text formats. Key command-line flags include:

- `-server`: Runs the iperf3 server in JSON output mode.
- `<SERVER_IP>`: Specifies the server's IP address for client mode.
- `-udp`: Switches the test from TCP to UDP.
- `-bitrate`: Sets the target bitrate for UDP tests (e.g., `10M`).
- `-iter`: Specifies the number of test iterations.

The output files generated by the script serve as the basis for our statistical analysis and subsequent comparison with the theoretical predictions.

4 ANALYSIS AND FINDINGS

…

4.1 TCP Performance

…

4.2 UDP Performance

…

5 CONCLUSION

…

A APPENDIX

…

```
# code here
```

Listing 1: Snipped name