

Performance Evaluation in Ethernet and WiFi Scenarios

Andrea Botticella^{*†}

andrea.botticella@studenti.polito.it

Renato Mignone^{*†}

renato.mignone@studenti.polito.it

Elia Innocenti^{*†}

elia.innocenti@studenti.polito.it

Simone Romano^{*†}

simone.romano2@studenti.polito.it

ABSTRACT

This report examines the performance of wireless and device-to-device communication by comparing theoretical predictions with experimental results in three scenarios: WiFi-only, Ethernet-only, and a mixed configuration. Using iperf3 and Wireshark, we measured goodput and analyzed its variability under different conditions. The experimental data were contrasted with theoretical estimates based on protocol efficiencies and network overheads. Our findings underscore Ethernet's stability and highlight the challenges of WiFi's shared medium and half-duplex constraints.

1 BACKGROUND AND OBJECTIVES

- 1 This laboratory evaluates and compares the performance of wired
2 and wireless communication in a local area network by setting
3 up three scenarios: both devices on WiFi, both on Ethernet, and a
4 mixed configuration with one on each. The main objectives of the
5 lab are to:
- 6 • Measure performances using iperf3 and Wireshark [7, 8].
 - 7 • Analyze the variability and stability of the connection in each
8 scenario by collecting data over multiple test runs.
 - 9 • Compare the experimental results against theoretical predictions
10 based on protocol efficiencies and network overhead.
 - 11 • Investigate potential sources of performance degradation in wire-
12 less communication, such as interference, half-duplex operation,
13 and shared medium limitations.
 - 14 These experiments provide practical insights into the strengths and
15 limitations of both Ethernet and WiFi, crucial for optimizing mixed
16 network performance.

2 METHODOLOGY AND CONCEPTS

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

19 2.1 Selected Tools

- 20 To evaluate the performance of both Ethernet and WiFi connections,
21 we utilized several specialized tools:
- 22 • **iperf3**: Used to generate traffic and measure goodput in both
23 TCP and UDP modes. By executing repeated tests, iperf3 provides
24 key metrics such as minimum, maximum, average, and standard
25 deviation of the throughput.
 - 26 • **Wireshark**: Used to capture and analyze network traffic, Wire-
27 shark helped inspect data flows, identify frames, and validate
28 results. It also generated useful charts for analyzing TCP streams.

^{*}The authors collaborated closely in developing this project.

[†]All the authors are students at Politecnico di Torino, Turin, Italy.

- **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, and computes summary statistics. The script accepts several command-line flags, as detailed in the Appendix A.

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.

1. For **Ethernet** [1], the efficiency for TCP is computed as:

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

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

- $MSS \approx 1460$ 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:

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

2. For **WiFi** [2], 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). In the case of **TCP over WiFi**, the effective efficiency is typically around 80% under optimal conditions. This lower efficiency arises from the extra overhead associated with TCP's connection-oriented features—such as congestion control, flow control, and the guarantee of in-order delivery—which require additional control packets and retransmissions. In contrast, **UDP over WiFi** generally attains an efficiency of approximately 85–90% by avoiding these mechanisms, leading to a simpler and faster data transmission process.

$$\eta_{TCP}^{WiFi} (\approx 80\%) \quad \text{and} \quad \eta_{UDP}^{WiFi} (\approx 85\%-90\%).$$

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 <i>Operating System:</i> Ubuntu 24.04.2 LTS [11] <i>Ethernet Interface:</i> Realtek RTL8111/8168/8211/8411 [3] <i>Wireless Interface:</i> Realtek RTL8852BE (802.11ax) 2x2 [4]
PC2	Microsoft Surface Laptop Go 3 <i>Operating System:</i> Ubuntu 24.10 [11] <i>Ethernet Interface:</i> via Anker PowerExpand+ USB-C Hub [10] <i>Wireless Interface:</i> Intel Alder Lake-P CNVi (802.11ax) 2x2 [5]
Router	Vodafone Power Station Wi-Fi 6 <i>Ethernet Ports:</i> 4 × 1 GbE ports <i>Wi-Fi:</i> Dual-band 802.11ax (2.4 GHz 2x2, 5 GHz 4x4) [12]
Cables	CAT.5E (up to 1 Gbps)

Table 1: Summary of Hardware and Network Configuration

Connection	Key Specifications	
Ethernet	<i>Cabling:</i> CAT.5E <i>Nominal Speed:</i> 1 Gbps	<i>Protocol:</i> Ethernet II
Wi-Fi	<i>Standard:</i> 802.11ax <i>Nominal Speed:</i> 1200 Mbps <i>Bandwidth:</i> 80 MHz	<i>Security Protocol:</i> WPA2-AES <i>Frequency:</i> 5 GHz <i>Channel:</i> 100

Table 2: Ethernet and Wi-Fi Connection Specifications

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:

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

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

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 1.2 GbEbps, 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:

$$G_{TCP}^{WiFi} \leq 0.80 \times 1.2 \text{ Gbps} \times \frac{1}{2} \approx 480 \text{ Mbps},$$

and similarly for UDP, with a different efficiency factor:

$$G_{UDP}^{WiFi} \leq 0.85 \times 1.2 \text{ Gbps} \times \frac{1}{2} \approx 510 \text{ Mbps}.$$

3. Mixed Scenario:

In this configuration, one device (PC1) is connected via Ethernet while the other (PC2) uses its Wi-Fi interface. Since only one side is on Wi-Fi, we average the Wi-Fi portion and the Ethernet portion rather than halving for two Wi-Fi paths. Hence, the expected goodput is:

$$G_{TCP}^{Mixed} \leq \frac{(0.80 \times 1.2 \text{ Gbps}) + (0.949 \times 1.0 \text{ Gbps})}{2} \approx 955 \text{ Mbps},$$

$$G_{UDP}^{Mixed} \leq \frac{(0.85 \times 1.2 \text{ Gbps}) + (0.957 \times 1.0 \text{ Gbps})}{2} \approx 989 \text{ Mbps}.$$

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.

4 ANALYSIS AND FINDINGS

4.1 TCP Performance

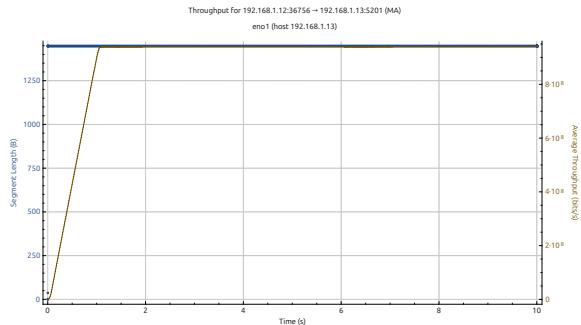
The performance tests using TCP reveal several noteworthy trends. The analysis for TCP performance in different scenarios is organized as follows:

Test	TCP: Goodput per flow (Mbps)				
	Prediction	Average	Min	Max	Std
Both WiFi	480	434.7	396.3	461.96	22.5
Both Ethernet	949	939.6	938.2	942.7	1.5
Mixed	955	663.7	619.2	698.86	26.6

Table 3: TCP Results (Client → Server)

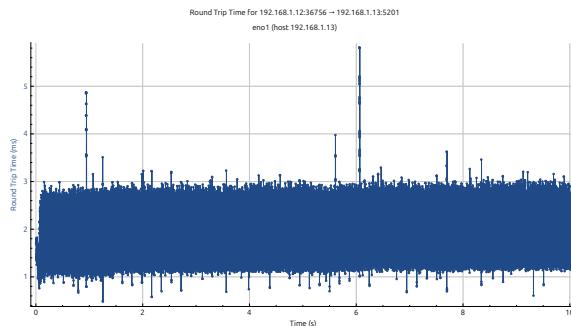
113 **1. Both Ethernet:**

114 Figure 1 shows the TCP throughput measured in the Ethernet scenario. The graph reveals a rapid ramp-up in throughput during
 115 the first few seconds, followed by a stable transmission rate that approaches the theoretical value. The **Maximum Segment**
 116 **Size (MSS)** reaches and remains stable at 1500 bytes, as defined by the TCP protocol. Additionally, the **bandwidth** is stable at
 117 **950 Mbps**, as indicated by the results and the low standard deviation.



121 **Figure 1:** TCP Throughput in the Ethernet Scenario.

122 Figure 2 illustrates the round-trip time (RTT), which remains
 123 very low (typically within 1-3 milliseconds), highlighting the
 124 minimal latency in wired connections.



125 **Figure 2:** TCP Round Trip Time in the Ethernet Scenario.

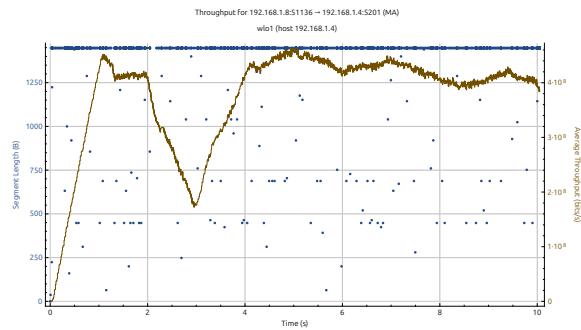
126 Overall, the Ethernet scenario demonstrates a near-ideal per-
 127 formance with high throughput and minimal latency, closely
 128 matching the theoretical predictions.

129 **2. Both WiFi:**

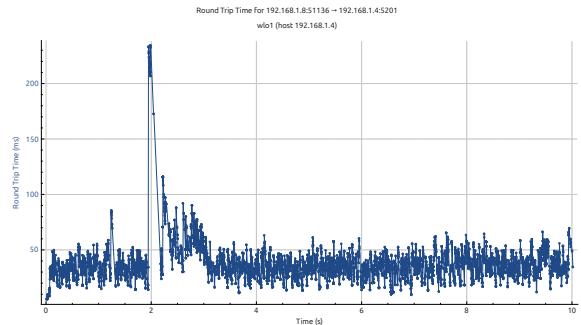
130 In the WiFi scenario, the throughput graph (Fig. 3) shows an
 131 initial ramp-up phase during the first 2 seconds, after which the
 132 throughput fluctuates around an average value of 434.7 Mbps.
 133 These fluctuations suggest that protocol overhead, wireless in-
 134 terference, and the half-duplex nature of WiFi adversely affect
 135 performance.

136 The round-trip time (RTT) measurements (Fig. 4) reveal RTT val-
 137 ues ranging from about 20-50 ms, indicating intermittent delays
 138 likely due to congestion and contention in the wireless medium.

139 Overall, while the theoretical capacity for TCP over WiFi is
 140 estimated to be around 480 Mbps, the experimental data indicate



141 **Figure 3:** TCP Throughput in the WiFi Scenario.

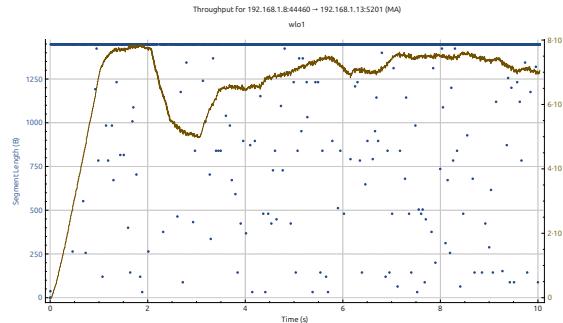


142 **Figure 4:** TCP Round Trip Time in the WiFi Scenario.

143 that real-world factors does not reduce that much the effective
 144 throughput of the WiFi network. This due to the fact that the
 145 network was in an ideal condition, where only the client and
 146 the server were connected to the access point, and no other
 147 devices were generating traffic.

148 **3. Mixed:**

149 Figure 5 displays the TCP throughput for the mixed configura-
 150 tion. The graph shows that the throughput reaches a stable level
 151 after an initial ramp-up phase, although it remains below the
 152 expected reduction due to the reliance on the wireless link.



153 **Figure 5:** TCP Throughput in the Mixed Ethernet/WiFi Scenario.

154 The round-trip time (RTT) measurements, presented in Figure 6,
 155 indicate moderate latency, with RTT values generally remain-
 156 ing within a lower range compared to the pure WiFi scenario.
 157 This suggests that the wired segment helps in reducing overall
 158 latency.

142 that real-world factors does not reduce that much the effective
 143 throughput of the WiFi network. This due to the fact that the
 144 network was in an ideal condition, where only the client and
 145 the server were connected to the access point, and no other
 146 devices were generating traffic.

147 **3. Mixed:**

148 Figure 5 displays the TCP throughput for the mixed configura-
 149 tion. The graph shows that the throughput reaches a stable level
 150 after an initial ramp-up phase, although it remains below the
 151 expected reduction due to the reliance on the wireless link.

152

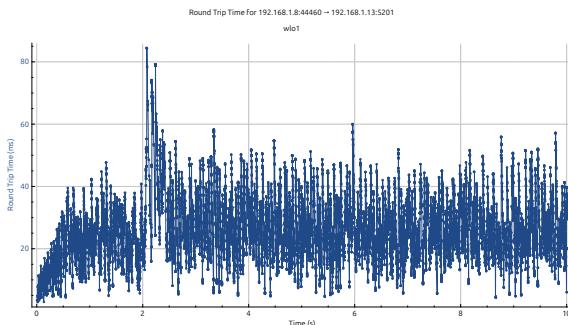


Figure 6: TCP Round Trip Time in the Mixed Ethernet/WiFi Scenario.

161 3a. Shared Capacity:

162 In this scenario, a third host connected to the same access point
 163 was concurrently downloading the film "Natale a Rio" (directed
 164 by Neri Parenti), which introduced significant interference during
 165 the tests. This additional traffic compromised the available
 166 network capacity, leading to degraded performance. As we can
 167 see in (Fig. 7), the other host starts the download from the sec-
 168 ond test (~25 seconds). In fact the value of the throughput, goes
 169 from the one of the standard Mixed Scenario (~670Mbps), to a
 170 lower one (~500Mbps). This behavior is due to the fact that the
 171 bandwidth is shared between the two hosts, and the download
 of the movie is consuming useful bandwidth.

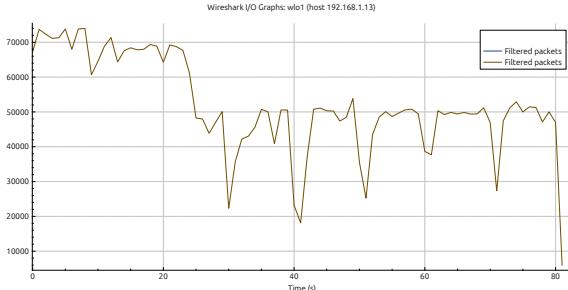


Figure 7: Wireshark I-O Graph for TCP in the Shared Capacity Scenario.

172 4.2 UDP Performance

173 The UDP tests offer an insightful comparison to the TCP results
 174 by eliminating congestion control and acknowledgment overhead.
 175 The analysis for UDP performance across different scenarios is
 176 structured as follows:

Test	UDP: Goodput per flow (Mbps)				
	Prediction	Average	Min	Max	Std
Both WiFi	510	487.8	453.1	499.9	15.8
Both Ethernet	957	952.8	948.3	954.6	1.73
Mixed	989	674.9	636.6	717.8	28

Table 4: UDP Results (Client → Server)

177 1. Both Ethernet:

178 The throughput achieved is very close to the theoretical pre-
 179 diction, confirming that the wired setup reliably supports high-
 180 speed data transfer. The absence of retransmission or congestion

182 control overhead in UDP further contributes to this consistent
 183 performance. In summary, the wired (Ethernet) tests demon-
 184 strate that both TCP and UDP protocols achieve performance
 185 levels very close to their theoretical capaci This confirms that,
 186 in a controlled wired environment, network performance is
 187 minimally impacted by protocol overhead or environmental
 188 factors.

189 2. Both WiFi:

191 A direct comparison between TCP and UDP in the WiFi scenario
 192 shows that UDP can achieve marginally higher throughput due
 193 to its reduced overhead. This discrepancy between capacity and
 194 theoretical goodput emphasizes the impact of wireless inter-
 195 ference, channel contention, and protocol-specific overhead on
 196 performance.

197 3. Mixed:

199 In summary, the mixed scenario demonstrates that while the
 200 presence of a wired connection on one end improves overall
 201 latency and stability compared to a full WiFi configuration,
 202 the performance remains primarily constrained by the wireless
 203 link. Both TCP and UDP protocols achieve throughput values
 204 that are consistent with theoretical predictions for a mixed
 205 Ethernet/WiFi environment.

206 3a. Shared Capacity:

207 The UDP tests under the shared capacity scenario also reveal the
 208 negative impact of the additional download traffic. Although
 209 UDP is less affected by protocol overhead, the increased con-
 210 tention for the wireless medium leads to degraded performance.
 211 Despite the inherent resilience of UDP to retransmission de-
 212 lays, the interference from the third host causes a noticeable
 213 reduction in performance. Overall, the shared capacity scenario
 214 clearly demonstrates that when a third host generates signifi-
 215 cant traffic (as in the case of streaming a movie), the available
 216 network resources are further divided, leading to performance
 217 degradation for both TCP and UDP protocols. This scenario
 218 highlights the importance of considering real-world usage pat-
 219 terns and interference when designing and evaluating network
 220 performance.

5 CONCLUSION

221 This project compared wired (Ethernet) and wireless (WiFi) network
 222 performance using iperf3 measurements and Wireshark analysis.
 223 Ethernet tests approached theoretical goodput with minimal la-
 224 tency, confirming the efficiency of controlled wired environments.
 225 In contrast, WiFi exhibited lower throughput and higher latency
 226 due to interference, contention, and half-duplex constraints. The
 227 mixed scenario showed that while a wired link can reduce latency,
 228 the wireless segment remains the bottleneck. Finally, adding a third
 229 host performing heavy traffic (movie streaming) further degraded
 230 both TCP and UDP results, highlighting how real-world congestion
 231 significantly impacts performance.

232 These observations underscore the need to consider environmen-
 233 tal and traffic factors in network design. While theoretical models
 234 provide upper bounds, practical limitations such as interference

235 and shared medium contention ultimately determine real-world
236 performance.

A APPENDIX

237 Server Mode Initialization

```
238 def run_server():
239     """
240     Run iperf3 server with clean output handling.
241     """
242     server_logger.info("Starting iperf3 server...")
243
244     proc = subprocess.Popen(
245         ["iperf3", "-s", "-J"],
246         stdout=subprocess.PIPE,
247         stderr=subprocess.PIPE,
248         text=True,
249         bufsize=1,
250     )
251     # Handle server output and errors in a separate
252     # thread...
```

Listing 1: Excerpt for server mode initialization.

253 Client Mode Execution and Reporting

```
254 def run_client(server_ip, udp=False, bitrate="1M",
255     iterations=10):
256     """
257     Run iperf3 client tests and generate reports.
258     """
259     for i in range(iterations):
260         cmd = ["iperf3", "-c", server_ip, "-J", "-t"
261             , "10", "-i", "1"]
262         if udp:
263             cmd.extend(["-u", "-b", bitrate])
264
265         result = subprocess.run(
266             cmd,
267             capture_output=True,
268             text=True,
269             check=True
270         )
271
272         data = json.loads(result.stdout)
273         # Extract test data, compute statistics, and
274         log results...
```

Listing 2: Excerpt for client mode execution.

275 Logging and Output Management

```
276 def setup_logger(log_file, name):
277     logger = logging.getLogger(name)
278     logger.setLevel(logging.DEBUG)
279     handler = logging.FileHandler(log_file)
280     formatter = logging.Formatter(
281         '%(asctime)s - %(levelname)s - %(message)s'
282     )
283     handler.setFormatter(formatter)
284     logger.addHandler(handler)
285     return logger
```

Listing 3: Excerpt for logging setup.

286 These snippets illustrate how the script handles server mode ini-
287 tialization, client tests (both TCP and UDP), and logging. Error
288 handling, multithreaded stderr management, and CSV report gen-
289 eration are also included in the complete script.

All the code for the report and lab material is available on the repository [13]: <https://github.com/WDCSecure/LabWiFi.git> 290
291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324

REFERENCES

- [1] 2018. IEEE Standard for Ethernet. (2018). https://standards.ieee.org/standard/802_3-2018.html 292
293
- [2] 2021. IEEE Standard for Information technology—Telecommunications and information exchange between systems Local and metropolitan area networks—Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for High Efficiency WLAN. (2021). https://standards.ieee.org/standard/802_11ax-2021.html 294
295
- [3] Realtek Semiconductor Corp. 2023. *Realtek RTL8111/8168/8211/8411: PCI Express Gigabit Ethernet controllers*. <https://www.realtek.com> 299
300
- [4] Realtek Semiconductor Corp. 2023. *Realtek RTL8852BE: 802.11ax Wi-Fi 6 PCIe network interface card*. <https://www.realtek.com> 301
302
- [5] Intel Corporation. 2023. *Intel Alder Lake-P CNVi: Integrated Wi-Fi 6E solution*. <https://www.intel.com/> 303
304
- [6] Python Software Foundation. 2023. *Python: A programming language that lets you work quickly and integrate systems more effectively*. <https://www.python.org/> 305
306
- [7] The Wireshark Foundation. 2023. *Wireshark: The world's foremost network protocol analyzer*. <https://www.wireshark.org/> 307
308
- [8] The iperf3 Development Team. 2023. *iperf3: A TCP, UDP, and SCTP network bandwidth measurement tool*. <https://iperf.fr/> 309
310
- [9] Jangeun Jun, P. Peddabachagari, and M. Sichitiu. 2003. Theoretical maximum throughput of IEEE 802.11 and its applications. In *Second IEEE International Symposium on Network Computing and Applications, 2003. NCA 2003*. 249–256. <https://doi.org/10.1109/NCA.2003.1201163> 311
312
- [10] Anker Innovations Limited. 2023. *Anker PowerExpand+ USB-C Hub: A USB-C hub with Ethernet and other interfaces*. <https://www.anker.com/> 313
314
- [11] Canonical Ltd. 2023. *Ubuntu: An open-source software platform that runs everywhere from the PC to the server and the cloud*. <https://ubuntu.com/> 315
316
- [12] Vodafone Group Plc. 2023. *Vodafone Power Station Wi-Fi 6: A dual-band Wi-Fi 6 router*. <https://www.vodafone.it/privati/area-sopporti/assistenza-dispositivi/vodafone-station/vodafone-power-station-wifi6.html> 317
318
- [13] WDCSecure. 2025. LabWiFi: repository containing the material and the documentation for the WiFi Lab. (2025). <https://github.com/WDCSecure/LabWiFi.git> 319
320
- [14] Wikipedia. 2008. Natale a Rio. (2008). https://en.wikipedia.org/wiki/Natale_a_Rio 321
322