

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
 115 during the first few seconds, followed by a stable transmission rate that approaches the theoretical value.
 116
 117

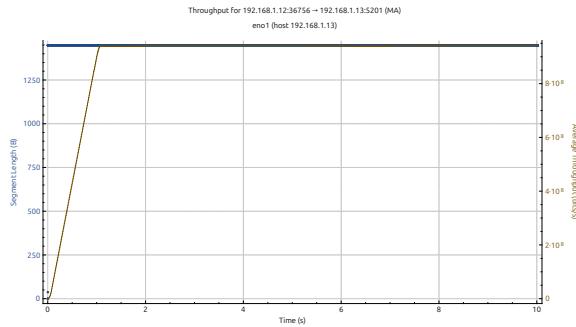


Figure 1: TCP Throughput in the Ethernet Scenario.

118 Figure 2 illustrates the round-trip time (RTT), which remains
 119 very low (typically within a few milliseconds), highlighting the
 120 minimal latency in wired connections.

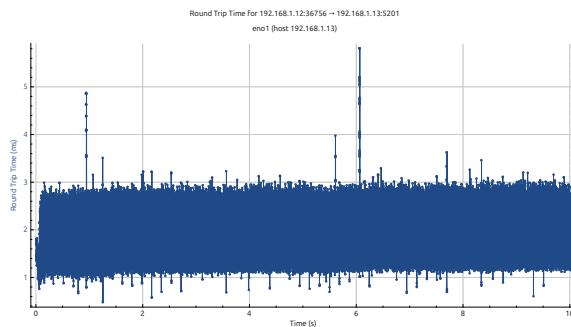


Figure 2: TCP Round Trip Time in the Ethernet Scenario.

121 Furthermore, the I-O graph (Fig. ??) confirms a consistent packet
 122 flow with little variation, indicating that the Ethernet setup
 123 effectively utilizes the available capacity.

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

127 **2. Both WiFi:**

128 In the WiFi scenario, the throughput graph (Fig. 3) shows an
 129 initial ramp-up phase during the first 2 seconds, after which the
 130 throughput fluctuates around an average value that is signifi-
 131 cantly lower than the theoretical maximum of approximately
 132 347 Mbps. These fluctuations suggest that protocol overhead,
 133 wireless interference, and the half-duplex nature of WiFi ad-
 134 versely affect performance.

135
 136 The round-trip time (RTT) measurements (Fig. 4) reveal RTT
 137 values ranging from about 50 to 200 ms, indicating intermittent
 138 delays likely due to congestion and contention in the wireless
 139 medium.

140 Furthermore, the I-O graph (Fig. ??) illustrates a variable num-
 141 ber of transmitted packets per interval, reflecting the dynamic

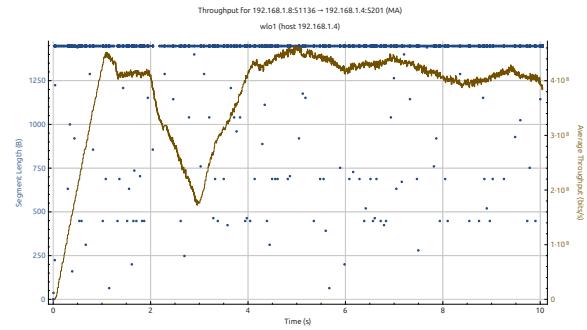


Figure 3: TCP Throughput in the WiFi Scenario.

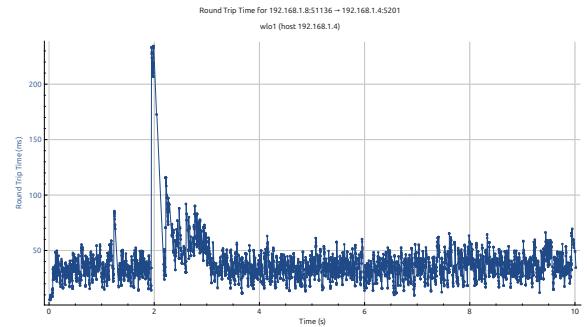


Figure 4: TCP Round Trip Time in the WiFi Scenario.

nature of WiFi communication where channel conditions and
 collision avoidance mechanisms influence performance.
 Overall, while the theoretical capacity for TCP over WiFi is
 estimated to be around 347 Mbps, the experimental data indicate
 that real-world factors substantially reduce the effective
 throughput.

3. Mixed:

Figure 5 displays the TCP throughput for the mixed configuration. The graph shows that the throughput reaches a stable level after an initial ramp-up phase, although it remains below the Ethernet scenario and is consistent with the expected reduction due to the reliance on the wireless link.

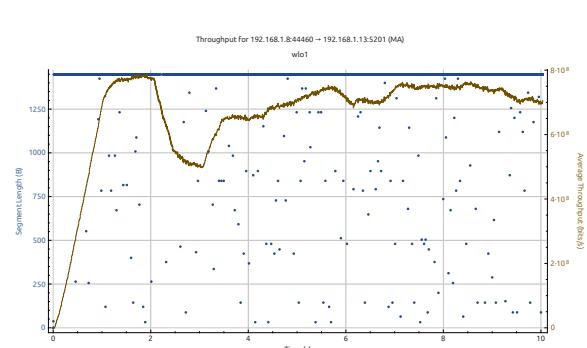


Figure 5: TCP Throughput in the Mixed Ethernet/WiFi Scenario.

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

160 latency.
161

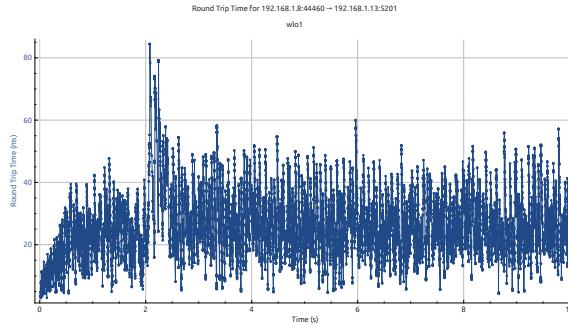


Figure 6: TCP Round Trip Time in the Mixed Ethernet/WiFi Scenario.

162 The I-O graph for TCP (Fig. ??) shows a relatively steady packet
163 flow over the test intervals, confirming that the mixed config-
164 uration maintains a stable performance despite the inherent
165 variability of the wireless link.

3a. Shared Capacity:

166 In this scenario, a third host connected to the same access point
167 was concurrently downloading the film "Natale a Rio" (directed
168 by Neri Parenti), which introduced significant interference dur-
169 ing the tests. This additional traffic compromised the available
170 network capacity, leading to degraded performance. Figure ??
171 shows the TCP throughput under this shared capacity condi-
172 tion. Compared to the mixed scenario without interference, the
173 throughput exhibits a notable decrease. The average throughput
174 is lower, reflecting the reduced available bandwidth caused by
175 the competing download traffic.

176 The round-trip time measurements (Fig. ??) indicate increased
177 variability and slightly elevated latency. Although the RTT
178 values remain relatively moderate, the fluctuations suggest that
179 the network experiences occasional congestion and delays as a
180 result of the third host's activity.

181 The I-O graph for TCP (Fig. 7) further confirms the impact of
182 the interference. The graph displays irregular intervals and a
183 lower packet transmission rate compared to the mixed scenario
184 without the additional load, demonstrating how the extra traffic
185 disrupts the steady flow of data.

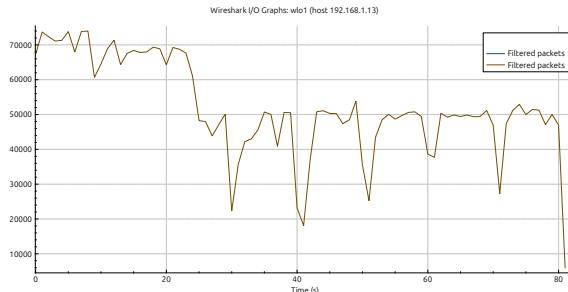


Figure 7: Wireshark I-O Graph for TCP in the Shared Capacity Scenario.

4.2 UDP Performance

187 The UDP tests offer an insightful comparison to the TCP results
188 by eliminating congestion control and acknowledgment overhead.

The analysis for UDP performance across different scenarios is
190 structured as follows:
191

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)

1. Both Ethernet:

In the Ethernet configuration, the I-O graph for UDP indicates
193 a steady packet flow, with minimal fluctuations compared to
194 the WiFi scenario.
195

The throughput achieved is very close to the theoretical pre-
196 diction, confirming that the wired setup reliably supports high-
197 speed data transfer. The absence of retransmission or congestion
198 control overhead in UDP further contributes to this consistent
199 performance. In summary, the wired (Ethernet) tests demon-
200 strate that both TCP and UDP protocols achieve performance
201 levels very close to their theoretical capaci This confirms that,
202 in a controlled wired environment, network performance is
203 minimally impacted by protocol overhead or environmental
204 factors.
205

2. Both WiFi:

In the WiFi scenario, the I-O graph for UDP demonstrates a
207 more consistent packet flow compared to TCP. However, despite
208 the smoother transmission, the overall throughput remains
209 below the theoretical upper bound. The lack of retransmission
210 mechanisms in UDP allows for slightly higher instantaneous
211 throughput; nonetheless, factors like interference and channel
212 contention continue to impact performance.
213 A direct comparison between TCP and UDP in the WiFi scenario
214 shows that UDP can achieve marginally higher throughput due
215 to its reduced overhead. This discrepancy between capacity and
216 theoretical goodput emphasizes the impact of wireless inter-
217 ference, channel contention, and protocol-specific overhead on
218 performance.
219

3. Mixed:

In the mixed scenario, the UDP I-O graph indicates a consistent
221 flow of packets, similar to the TCP case but with slightly less
222 variability due to the absence of congestion control.
223

In summary, the mixed scenario demonstrates that while the
224 presence of a wired connection on one end improves overall
225 latency and stability compared to a full WiFi configuration,
226 the performance remains primarily constrained by the wireless
227 link. Both TCP and UDP protocols achieve throughput values
228 that are consistent with theoretical predictions for a mixed
229 Ethernet/WiFi environment.
230

3a. Shared Capacity:

The UDP tests under the shared capacity scenario also reveal the
232 negative impact of the additional download traffic. Although
233 UDP is less affected by protocol overhead, the increased con-
234 tention for the wireless medium leads to degraded performance.
235 Despite the inherent resilience of UDP to retransmission de-
236 lays, the interference from the third host causes a noticeable
237

238 reduction in performance. Overall, the shared capacity scenario
239 clearly demonstrates that when a third host generates signifi-
240 cant traffic (as in the case of streaming a movie), the available
241 network resources are further divided, leading to performance
242 degradation for both TCP and UDP protocols. This scenario
243 highlights the importance of considering real-world usage pat-
244 terns and interference when designing and evaluating network
245 performance.

5 CONCLUSION

246 This project compared wired (Ethernet) and wireless (WiFi) network
247 performance using iperf3 measurements and Wireshark analysis.
248 Ethernet tests approached theoretical goodput with minimal la-
249 tency, confirming the efficiency of controlled wired environments.
250 In contrast, WiFi exhibited lower throughput and higher latency
251 due to interference, contention, and half-duplex constraints. The
252 mixed scenario showed that while a wired link can reduce latency,
253 the wireless segment remains the bottleneck. Finally, adding a third
254 host performing heavy traffic (movie streaming) further degraded
255 both TCP and UDP results, highlighting how real-world congestion
256 significantly impacts performance.

257 These observations underscore the need to consider environmen-
258 tal and traffic factors in network design. While theoretical models
259 provide upper bounds, practical limitations such as interference
260 and shared medium contention ultimately determine real-world
261 performance.

A APPENDIX

262 Server Mode Initialization

```
263 def run_server():
264     """
265     Run iperf3 server with clean output handling.
266     """
267     server_logger.info("Starting iperf3 server...")
268
269     proc = subprocess.Popen(
270         ["iperf3", "-s", "-J"],
271         stdout=subprocess.PIPE,
272         stderr=subprocess.PIPE,
273         text=True,
274         bufsize=1,
275     )
276     # Handle server output and errors in a separate
277     # thread...
```

Listing 1: Excerpt for server mode initialization.

278 Client Mode Execution and Reporting

```
279 def run_client(server_ip, udp=False, bitrate="1M",
280                 iterations=10):
281     """
282     Run iperf3 client tests and generate reports.
283     """
284     for i in range(iterations):
285         cmd = ["iperf3", "-c", server_ip, "-J", "-t"
286               , "10", "-i", "1"]
287         if udp:
288             cmd.extend(["-u", "-b", bitrate])
289
290         result = subprocess.run(
291             cmd,
292             capture_output=True,
293             text=True,
294             check=True
295         )
296
297         data = json.loads(result.stdout)
298         # Extract test data, compute statistics, and
299         log results...
```

Listing 2: Excerpt for client mode execution.

300 Logging and Output Management

```
301 def setup_logger(log_file, name):
302     logger = logging.getLogger(name)
303     logger.setLevel(logging.DEBUG)
304     handler = logging.FileHandler(log_file)
305     formatter = logging.Formatter(
306         '%(asctime)s - %(levelname)s - %(message)s'
307     )
308     handler.setFormatter(formatter)
309     logger.addHandler(handler)
310     return logger
```

Listing 3: Excerpt for logging setup.

311 These snippets illustrate how the script handles server mode ini-
312 tialization, client tests (both TCP and UDP), and logging. Error
313 handling, multithreaded stderr management, and CSV report gen-
314 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> 315
316

REFERENCES

- [1] 2018. IEEE Standard for Ethernet. (2018). https://standards.ieee.org/standard/802_3-2018.html 317
318
- [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 319
320
- [3] Realtek Semiconductor Corp. 2023. *Realtek RTL8111/8168/8211/8411: PCI Express Gigabit Ethernet controllers*. <https://www.realtek.com> 324
325
- [4] Realtek Semiconductor Corp. 2023. *Realtek RTL8852BE: 802.11ax Wi-Fi 6 PCIe network interface card*. <https://www.realtek.com> 326
327
- [5] Intel Corporation. 2023. *Intel Alder Lake-P CNVi: Integrated Wi-Fi 6E solution*. <https://www.intel.com/> 328
329
- [6] Python Software Foundation. 2023. *Python: A programming language that lets you work quickly and integrate systems more effectively*. <https://www.python.org/> 330
331
- [7] The Wireshark Foundation. 2023. *Wireshark: The world's foremost network protocol analyzer*. <https://www.wireshark.org/> 332
333
- [8] The iperf3 Development Team. 2023. *iperf3: A TCP, UDP, and SCTP network bandwidth measurement tool*. <https://iperf.fr/> 334
335
- [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> 336
337
- [10] Anker Innovations Limited. 2023. *Anker PowerExpand+ USB-C Hub: A USB-C hub with Ethernet and other interfaces*. <https://www.anker.com/> 340
341
- [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/> 342
343
- [12] Vodafone Group Plc. 2023. *Vodafone Power Station Wi-Fi 6: A dual-band Wi-Fi 6 router*. <https://www.vodafone.it/privati/area-supporto/assistenza-dispositivi/vodafone-station/vodafone-power-station-wifi6.html> 344
345
- [13] WDCSecure. 2025. LabWiFi: repository containing the material and the documentation for the WiFi Lab. (2025). <https://github.com/WDCSecure/LabWiFi.git> 346
347
- [14] Wikipedia. 2008. Natale a Rio. (2008). https://en.wikipedia.org/wiki/Natale_a_Rio 348
349