

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.
5 The main objectives of the lab are to:
6 • Measure goodput using iperf3.
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
15 and limitations of both Ethernet and WiFi, crucial for optimizing
16 mixed 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.

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.

- **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, as detailed in the Appendix.

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**, 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**, 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.

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

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

65 and
 $\eta_{TCP}^{WiFi} = \eta_{TCP}^{Eth} \times \eta_{WiFi,TCP}$ ($\approx 80\%$),
 $\eta_{UDP}^{WiFi} = \eta_{UDP}^{Eth} \times \eta_{WiFi,UDP}$ ($\approx 85\text{--}90\%$).
66 These theoretical estimates set an upper bound on the achievable
67 goodput, against which our experimental results are compared.
68 Discrepancies between theory and practice are primarily due to
69 dynamic environmental factors, such as interference, channel vari-
70 ability, and the inherent limitations of wireless communication.

3 EXPERIMENTAL SETUP AND TEST CASES

3.1 Equipment and Configuration

72 In this section, we describe the hardware and software configura-
73 ration used to perform our network performance measurements.
74 Table 1 summarizes the main devices, their interfaces, and relevant
75 specifications.

Device	Key Specifications
PC1	Victus 16-s1005nl Notebook <i>Operating System:</i> Ubuntu 24.04.2 LTS <i>Ethernet Interface:</i> Realtek RTL8111/8168/8211/8411 <i>Wireless Interface:</i> Realtek RTL8852BE (802.11ax) 2x2
PC2	Microsoft Surface Laptop Go 3 <i>Operating System:</i> Ubuntu 24.10 <i>Ethernet Interface:</i> via Anker PowerExpand+ USB-C Hub <i>Wireless Interface:</i> Intel Alder Lake-P CNVi (802.11ax) 2x2
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)
Cables	CAT.5E (up to 1 Gbps)

Table 1: Summary of Hardware and Network Configuration

Connections	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

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

3.2 Evaluation Scenarios

81 We considered three distinct network configurations to assess the
82 performance differences between wired and wireless communica-
83 tions. For each scenario, the theoretical goodput is computed based
84 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. 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:

$$G_{TCP}^{Mixed} \leq 0.80 \times 1.2 \text{ Gbps} \approx 960 \text{ Mbps},$$

$$G_{UDP}^{Mixed} \leq 0.85 \times 1.2 \text{ Mbps} \approx 1020 \text{ Mbps}.$$

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	?	663.7	619.2	698.86	26.6
Shared Capacity	?	536.8	440.5	722.2	112

Table 3: TCP Results (Client → Server)

115 **1. Both Ethernet:**

116 Figure 1 shows the TCP throughput measured in the Ethernet scenario. The graph reveals a rapid ramp-up in throughput
 117 during the first few seconds, followed by a stable transmission rate that approaches the theoretical value.
 118
 119

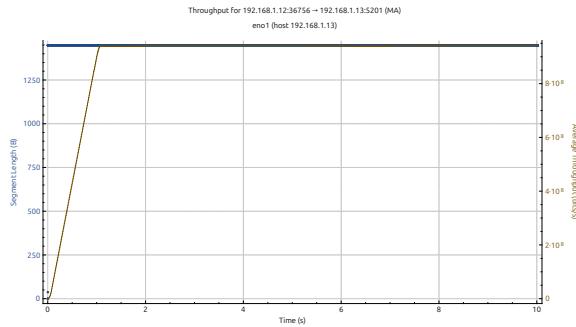


Figure 1: TCP Throughput in the Ethernet Scenario.

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

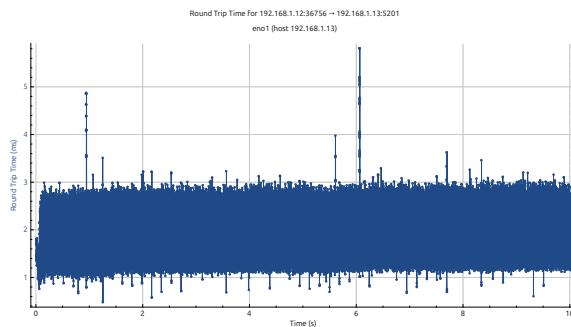


Figure 2: TCP Round Trip Time in the Ethernet Scenario.

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

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 that is signifi-
 133 cantly lower than the theoretical maximum of approximately
 134 347 Mbps. These fluctuations suggest that protocol overhead,
 135 wireless interference, and the half-duplex nature of WiFi ad-
 136 versely affect performance.

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

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

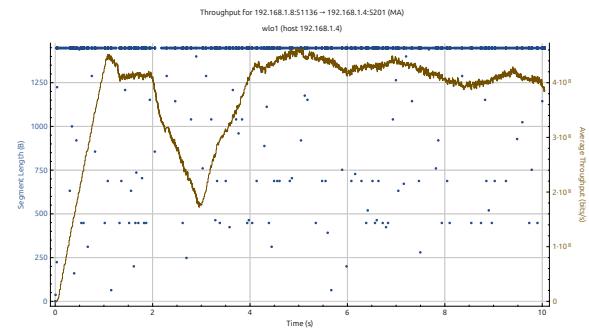


Figure 3: TCP Throughput in the WiFi Scenario.

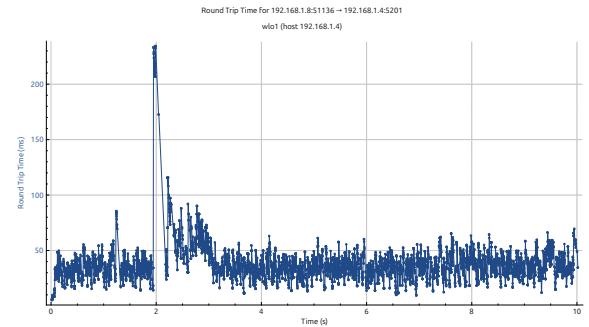


Figure 4: TCP Round Trip Time in the WiFi Scenario.

nature of WiFi communication where channel conditions and
 145 collision avoidance mechanisms influence performance.
 146 Overall, while the theoretical capacity for TCP over WiFi is
 147 estimated to be around 347 Mbps, the experimental data indi-
 148 cate that real-world factors substantially reduce the effective
 149 throughput.
 150

3. Mixed:

151 Figure 5 displays the TCP throughput for the mixed configura-
 152 tion. The graph shows that the throughput reaches a stable level
 153 after an initial ramp-up phase, although it remains below the
 154 Ethernet scenario and is consistent with the expected reduction
 155 due to the reliance on the wireless link.
 156

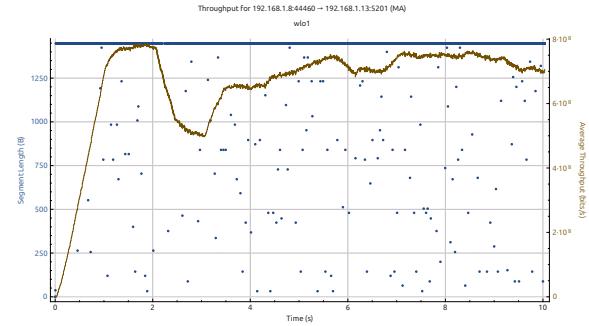


Figure 5: TCP Throughput in the Mixed Ethernet/WiFi Scenario.

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

162 latency.
163

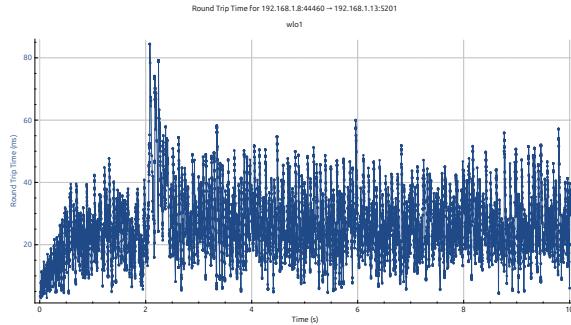


Figure 6: TCP Round Trip Time in the Mixed Ethernet/WiFi Scenario.

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

168 3a. Shared Capacity:

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

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

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

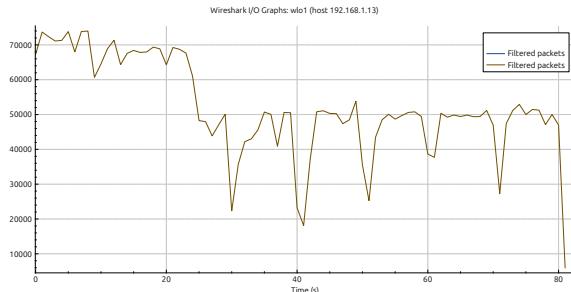


Figure 7: Wireshark I-O Graph for TCP in the Shared Capacity Scenario.

189 4.2 UDP Performance

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

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

Test	UDP: Goodput per flow (Mbps)				
	Prediction	Average	Min	Max	Std
Both WiFi	480	487.8	453.1	499.9	15.8
Both Ethernet	957	952.8	948.3	954.6	1.73
Mixed	?	674.9	636.6	717.8	28
Shared Capacity	?	472.1	355.9	699.1	121.1

Table 4: UDP Results (Client → Server)

192 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.

The throughput achieved is very close to the theoretical pre-
195 diction, confirming that the wired setup reliably supports high-
196 speed data transfer. The absence of retransmission or congestion
197 control overhead in UDP further contributes to this consistent
198 performance.

In summary, the wired (Ethernet) tests demonstrate that both
199 TCP and UDP protocols achieve performance levels very close
200 to their theoretical capacities, with TCP showing stable through-
201 put and low latency, and UDP exhibiting a consistent packet flow
202 and high throughput. This confirms that, in a controlled wired
203 environment, network performance is minimally impacted by
204 protocol overhead or environmental factors.

205 2. Both WiFi:

In the WiFi scenario, the I-O graph for UDP demonstrates a
206 more consistent packet flow compared to TCP. However, despite
207 the smoother transmission, the overall throughput remains
208 below the theoretical upper bound. The lack of retransmission
209 mechanisms in UDP allows for slightly higher instantaneous
210 throughput; nonetheless, factors like interference and channel
211 contention continue to impact performance.

A direct comparison between TCP and UDP in the WiFi scenario
212 shows that UDP can achieve marginally higher throughput due
213 to its reduced overhead. Nonetheless, both protocols suffer from
214 real-world limitations that prevent them from reaching their
215 theoretical capacities. This discrepancy between capacity and
216 throughput 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
220 flow of packets, similar to the TCP case but with slightly less
221 variability due to the absence of congestion control.

The throughput observed is in line with expectations given that
222 only the wireless link acts as the bottleneck.

In summary, the mixed scenario demonstrates that while the
223 presence of a wired connection on one end improves overall
224 latency and stability compared to a full WiFi configuration,
225 the performance remains primarily constrained by the wireless
226 link. Both TCP and UDP protocols achieve throughput values

238 that are consistent with theoretical predictions for a mixed
239 Ethernet/WiFi environment.

240 3a. **Shared Capacity:**

241 The UDP tests under the shared capacity scenario also reveal the
242 negative impact of the additional download traffic. Although
243 UDP is less affected by protocol overhead, the increased con-
244 tention for the wireless medium leads to degraded performance.
245 The UDP performance in this scenario is indirectly reflected
246 in the I-O graph. The steady flow of packets observed in a
247 non-interfered environment is disrupted, resulting in a lower
248 effective throughput.

249 Despite the inherent resilience of UDP to retransmission de-
250 lays, the interference from the third host causes a noticeable
251 reduction in performance. The graph shows that the packet flow
252 is not as consistent, further underlining the effects of shared
253 capacity when additional traffic is present.

254 Overall, the shared capacity scenario clearly demonstrates that
255 when a third host generates significant traffic (as in the case
256 of streaming a movie), the available network resources are fur-
257 ther divided, leading to performance degradation for both TCP
258 and UDP protocols. This scenario highlights the importance of
259 considering real-world usage patterns and interference when
260 designing and evaluating network performance.

5 CONCLUSION

261 In this project, we evaluated the performance of network commu-
262 nication under various scenarios using both wired (Ethernet) and
263 wireless (WiFi) connections. Our experimental results, obtained
264 through automated measurements with iperf3 and detailed packet
265 analysis with Wireshark, were compared against theoretical pre-
266 dictions of goodput for both TCP and UDP protocols.

267 Overall, the Ethernet scenario demonstrated near-ideal perfor-
268 mance, with throughput and latency closely matching the theoreti-
269 cal values. This confirms that a controlled wired environment can
270 efficiently utilize available bandwidth with minimal interference. In
271 contrast, the WiFi scenario showed a significant performance drop,
272 with fluctuations in throughput and increased latency due to the in-
273 herent limitations of wireless communication such as interference,
274 contention, and the half-duplex nature of WiFi.

275 The mixed scenario, where one device is connected via Ethernet
276 and the other via WiFi, presented an intermediate case. Here, while
277 the wired segment helped in reducing latency and stabilizing per-
278 formance, the overall throughput remained limited by the wireless
279 link. Finally, the shared capacity scenario—where an additional host
280 engaged in heavy traffic (streaming a movie)—further degraded per-
281 formance for both TCP and UDP tests. This clearly highlights the
282 impact of network congestion and shared medium contention on
283 real-world performance.

284 These findings emphasize the importance of considering environ-
285 mental and traffic-related factors when designing and optimizing
286 network infrastructures. While theoretical models provide useful
287 upper bounds, actual network performance is influenced by a range
288 of practical factors that must be taken into account for effective
289 network planning and troubleshooting.

A APPENDIX

290 Server Mode Initialization

```
291     def run_server():
292         """Run_iperf3_server_with_clean_output_handling.
293             """
294         server_logger.info("Starting_iperf3_server...")
295
296         proc = subprocess.Popen(
297             ["iperf3", "-s", "-J"],
298             stdout=subprocess.PIPE,
299             stderr=subprocess.PIPE,
300             text=True,
301             bufsize=1,
302         )
303         # Handle server output and errors in a separate
304         # thread...
```

Listing 1: Excerpt for server mode initialization.

305 Client Mode Execution and Reporting

```
306     def run_client(server_ip, udp=False, bitrate="1M",
307                     iterations=10):
308         """Run_iperf3_client_tests_and_generate_reports.
309             """
310         for i in range(iterations):
311             cmd = ["iperf3", "-c", server_ip, "-J", "-t"
312                   , "10", "-i", "1"]
313             if udp:
314                 cmd.extend(["-u", "-b", bitrate])
315
316             result = subprocess.run(
317                 cmd,
318                 capture_output=True,
319                 text=True,
320                 check=True
321             )
322
323             data = json.loads(result.stdout)
324             # Extract test data, compute statistics, and
325             log results...
```

Listing 2: Excerpt for client mode execution.

326 Logging and Output Management

```
327     def setup_logger(log_file, name):
328         logger = logging.getLogger(name)
329         logger.setLevel(logging.DEBUG)
330         handler = logging.FileHandler(log_file)
331         formatter = logging.Formatter(
332             '%(asctime)s - %(levelname)s - %(message)s'
333         )
334         handler.setFormatter(formatter)
335         logger.addHandler(handler)
336         return logger
```

Listing 3: Excerpt for logging setup.

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

REFERENCES

- | | |
|---|-----|
| [1] 2018. IEEE Standard for Ethernet. (2018). https://standards.ieee.org/standard/802_3-2018.html | 341 |
| [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 | 343 |
| [3] Realtek Semiconductor Corp. 2023. <i>Realtek RTL8111/8168/8211/8411: PCI Express Gigabit Ethernet controllers</i> . https://www.realtek.com/en/products/communications-network-ics/item/rtl8111h | 344 |
| [4] Realtek Semiconductor Corp. 2023. <i>Realtek RTL8852BE: 802.11ax Wi-Fi 6 PCIe network interface card</i> . https://www.realtek.com/en/products/communications-network-ics/item/rtl8852be | 345 |
| [5] Intel Corporation. 2023. <i>Intel Alder Lake-P CNVi: Integrated Wi-Fi 6E solution</i> . https://www.intel.com/content/www/us/en/products/docs/wireless-products/wi-fi-6e.html | 346 |
| [6] Python Software Foundation. 2023. <i>Python: A programming language that lets you work quickly and integrate systems more effectively</i> . https://www.python.org/ | 347 |
| [7] The Wireshark Foundation. 2023. <i>Wireshark: The world's foremost network protocol analyzer</i> . https://www.wireshark.org/ | 348 |
| [8] The iperf3 Development Team. 2023. <i>iperf3: A TCP, UDP, and SCTP network bandwidth measurement tool</i> . https://iperf.fr/ | 349 |
| [9] Anker Innovations Limited. 2023. <i>Anker PowerExpand+ USB-C Hub: A USB-C hub with Ethernet and other interfaces</i> . https://www.anker.com/products/variant/powerexpand-8in1-usbc-pd-media-hub/A83810A1 | 350 |
| [10] Canonical Ltd. 2023. <i>Ubuntu: An open-source software platform that runs everywhere from the PC to the server and the cloud</i> . https://ubuntu.com/ | 351 |
| [11] Vodafone Group Plc. 2023. <i>Vodafone Power Station Wi-Fi 6: A dual-band Wi-Fi 6 router</i> . https://www.vodafone.co.uk/broadband/wifi-hub | 352 |
| [12] Wikipedia. 2008. Natale a Rio. (2008). https://en.wikipedia.org/wiki/Natale_a_Rio | 353 |

342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370