WiFi performance lab report

Giorgia Moscato, Angelo Barbera, Alessandro Genova

ABSTRACT

This report explores the performance of Ethernet and Wi-Fi connections within a Local Area Network (LAN) environment using tools like iperf3 and Wireshark. By conducting a series of experiments, the report compares the goodput (speed at which useful data arrives at destination) over different protocols (TCP and UDP) in different network scenarios: direct Ethernet connections, Wi-Fi connections, and a hybrid setup combining both.

1 INTRODUCTION

The motivation behind this report is to assess and compare the performance of Ethernet and Wi-Fi connections within a LAN. Understanding the capabilities and limitations of these technologies is very important for optimizing network setups in practical scenarios. The goals of this lab include evaluating the goodput of TCP and UDP transmissions over Ethernet and Wi-Fi, analyzing the impact of protocol overhead and wireless constraints. This investigation aims to provide network performance considerations that can be useful to guide decisions related to a LAN network design.

2 TOOLS AND THEORY

2.1 Tools

The tests have been performed using iperf3 (version 3.9) [4], a tool for network performance measurement that can calculate either TCP or UDP goodput. To conduct an iperf3 test, the user needs to set up both a server and a client.
The main used options are:

- **Server Mode** `-s`: Run iperf in server mode.
- **Client Mode** `-c <IP_server>`: Run iperf in client mode, connecting to an iperf server with the specified IP address.
- **UDP Testing** `-u`: Use UDP rather than TCP.
- **Bandwidth** `-b`: Set target bandwidth to n bits/sec.
- **Buffer length** `-l`: The length of buffers to read or write.

For more details about the options visit the iPerf3 docs.
In the test performed with UDP, the following options were used:

```
iperf3 -c {ip_address} -u -l 1472 -b 1G
```

In the case of UDP, iperf3 tries to dynamically determine an appropriate sending size. If it cannot be determined, it defaults to 1460. However, to optimize goodput, it should be set to $MTU - 20 - 8$, which equals 1472, given MTU equals to 1500. In addition, the default bandwidth is 1 Mbps (in UDP case), however to not limit the throughput is better select 1 Gbps. Finally, Wireshark has been used to generate different plots in the different scenarios. For instance, the Round Trip Time plot can be useful to detect if congestion is arising, while the throughput plot shows the total number of bytes received within the Moving Average Window over the window duration, which shows the throughput consistency over time.

2.2 Goodput estimation

Given the capacity C at the physical layer, the goodput G is defined as the speed at which useful data is received at the application layer.

$$G = \frac{\text{useful data}}{\text{transfer time}} = \eta_{Protocol} * C$$

In order to calculate the goodput is needed the protocol efficiency that can be calculated in the following way (considering that MSS is the transport layer payload size, and $MSS = MTU - Headers(L3, L4)$ where MTU is 1500B and it's the commonly maximum IP datagram size):

$$\eta_{Protocol} = \frac{\text{MSS}}{MSS + Header_{TCP/UDP} + Header_{IP} + Header_{Eth}}$$

In the case of Ethernet the efficiencies for TCP and UDP protocols, respectively, are the following:

$$\eta_{TCP_{Eth}} = \frac{1460}{1460+20+20+38} = 94.9\%$$

$$\eta_{UDP_{Eth}} = \frac{1472}{1472+20+8+38} = 95.7\%$$

When dealing with Wi-Fi, the situation becomes more complex: it is not possible to replace $Header_{Eth}$ size with $Header_{WiFi}$ size in formula above. It necessary to consider the 802.11 protocol particular overhead and the fact that Wi-Fi operates as a shared medium. Wi-Fi communications are half duplex, meaning data can only be transmitted in one direction at a time, and certain frames (control frames) must be sent at slower speeds (1 Mbps) to ensure protocol correctness, which introduces latency. The half duplex nature effectively splits the maximum throughput in half because the Wireless Access Point can either receive or transmit data but cannot do both simultaneously. However, newer Wireless Access Points (like the one used for this report) use protocols like 802.11ac or 802.11ax, which incorporate Multi-User Multiple Input Multiple Output (MU-MIMO) technology, which allow to reduce this unwanted overhead effect, allowing to reach in optimal conditions an efficiency of 80% using TCP [6]. MU-MIMO technology enhances Wi-Fi performance by enabling an AP to receive data from multiple devices simultaneously or transmit data to multiple devices simultaneously [3, 5]. However, it's important to note that MU-MIMO is not full duplex; rather, it leverages spatial direction antennas to increase bandwidth. To describe a wireless device, abbreviation AxB:C is often used, where A represents the number of Tx radio chains available to the device, B represents the number of Rx radio chains, and C represents the number of spatial streams. You need at least one radio chain to make use of one spatial stream, so C is usually omitted because in most cases it has the same number of A and B. Spatial streams exploit the difference in signal paths caused by the space between antennas. This multipath phenomenon, once problematic for wireless communication, now allows multiple streams of data to be transmitted simultaneously from different

WiFi performance lab report, Torino, 2024
2024. ACM ISBN .
https://doi.org/

antennas. MU-MIMO in 802.11ac works only in the downstream direction, so only allows multiple simultaneous communications from AP to client stations. Upstream MU-MIMO was introduced later in 802.11ax. However, it's important to note that MU-MIMO signals are broadcast directionally. If two client devices are very close to each other, they would not benefit much from MU-MIMO. So, given that

$$\eta_{TCP_{WiFi}} = \eta_{TCP+IP} * \eta_{WiFi} \quad \text{and} \quad \eta_{TCP_{WiFi}} = 80\% \text{ [6]}$$

follows that

$$\eta_{WiFi} = 0.80 * \frac{1460+20+20}{1460} = 82.2\%$$

therefore the efficiency over Wi-Fi for UDP is the following:

$$\eta_{UDP_{WiFi}} = \eta_{UDP+IP} * \eta_{WiFi} = \frac{1472}{1472+20+8} * 0.822 =$$

$$= 0.981 * 0.822 = 80.6\%$$

3 LAB SETUP AND SCENARIOS

In this section, the setup of the experiments is described, including the network configuration, the hardware and the software utilized.

3.1 Hardware

The hardware used for the experiments includes:

- **Host A** :
  - Product name: Acer Nitro 5 AN515-55
  - Ethernet interface: Intel® Killer™ Gigabit Ethernet E2600
  - Wireless interface: Intel® Wi-Fi 6 AX201 160MHz (802.11ax) 2x2
  - Operating system: Ubuntu 22.04.4 LTS
- **Host B** :
  - Product name: Apple MacBook Pro 13" 2019
  - Ethernet interface: Rankie USB3.0 to Gigabit Ethernet Adapter
  - Wireless interface: Broadcom BCM4377b Wi-Fi 5 80MHz (802.11ac) 2x2
  - Operating system: MacOS Sonoma 14.0
- **Router**:
  - Product name: iliadbox with Wi-Fi 5 technology
  - Ethernet interface: 1 Gbps LAN port
  - Wireless interface: WiFi 11n 2x2 2.4GHz, 11ac 4x4 5GHz
- **Ethernet** cable: UTP Cat6, 1 Gbps

The network experiments configurations are the following:

- **Ethernet Link**:
  - Protocol: 802.3ab (1000BASE-T Full duplex)
  - Nominal link speed: 1 Gbps
- **WiFi Link**:
  - Protocol: 802.11ac
  - Frequency Band: 5 GHz
  - Bandwidth: 80 MHz
  - Channel: 44
  - Nominal link speed: 867 Mbps
  - Number of connected stations: 6
  - Security protocol: WPA2-AES

In the figure 1 it is possible to notice that the Wi-Fi channel used by the access point during the tests (channel 44) does not have an high noise level and it has a low occupation rate.

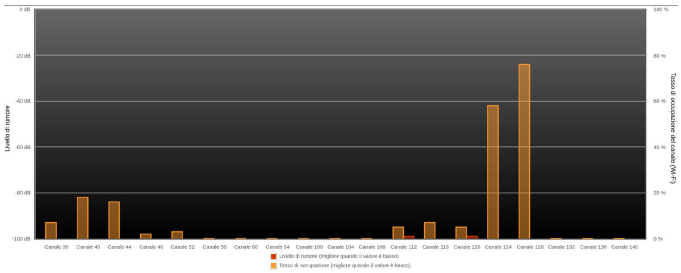

Figure 1: 5GHz band occupation and noise

3.2 Scenarios

The tests scenario are the following:

- **Both Ethernet**: Host A and Host B are directly connected trough the Ethernet cable
- **Both Wi-Fi**: Host A and Host B are connected to the same Wi-Fi network
- **Mixed**: The Host A is connected to the Router using the Ethernet cable, the Host B is connected to the Router Wi-Fi network

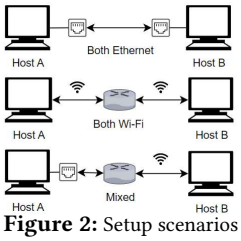

Figure 2: Setup scenarios

The expected goodputs, in the different scenario are (C is the capacity of the bottleneck link):

- Both Ethernet
  - TCP: $G \leq \eta_{TCP_{Eth}} * C_{Eth} = 94.9\% * 1000Mbps = 949Mbps$
  - UDP: $G \leq \eta_{UDP_{Eth}} * C_{Eth} = 95.7\% * 1000Mbps = 957Mbps$
- Both WiFi: since both Host A and Host B have 2x2 wireless cards (so 2 spatial streams), then the maximum bitrate (for a 80MHz channel) is for both hosts 867 Mbps [1]. Furthermore, since the medium is shared and half duplex, and since Host A and Host B Wi-Fi speed are the same and it is needed to transmit twice the same message, the throughput is cut in half, so it is necessary to take this into account dividing by 2.
  - TCP: $G \leq \eta_{TCP_{WiFi}} * C_{WiFi} * \frac{1}{2} = 80\% * 867Mbps * \frac{1}{2} = 346.8Mbps$
  - UDP: $G \leq \eta_{UDP_{WiFi}} * C_{WiFi} * \frac{1}{2} = 80.6\% * 867Mbps * \frac{1}{2} = 372.8Mbps$
- Mixed: in this case to calculate the maximum goodput it is needed to consider the bottleneck link, so the WiFi. The throughput is not divided by 2 since the message is sent only once in the Wi-Fi link.
  - TCP: $G \leq \eta_{TCP_{WiFi}} * C_{WiFi} = 80\% * 867Mbps = 693.6Mbps$

– UDP: $G \leq \eta_{UDP_{WiFi}} * C_{WiFi} = 80.6\% * 867Mbps = 698.8Mbps$

4 RESULTS

In this section the experiments results are compared with the predictions. Additionally some relevant plots are explained to show useful insights. In all cases the tests results are coherent with the predictions (which is an upper bound). In the Ethernet scenario the values are almost equal to predictions while in the Both Wi-Fi scenario and in the Mixed scenario the test results are lower then the predictions mostly due to throughput variations over a wireless link.

4.1 TCP

The predictions are compared with the actual results for each scenario in Table 1 for TCP, and in Table 2 for UDP. Those results were obtained using the script inserted in 6. Results in the table are goodput values, while plots are throughput values arrived at receiver (throughput and goodput plots have the same trend, they are almost equivalent graphically). These plots are taken from the receiver side.

Test		TCP: Goodput per flow (Mbps)				
		Prediction	Average	Min	Max	Std
Both Ethernet	A → B	949	940.9	940	941	0.3
	B → A	949	940.4	938	941	0.92
Both Wi-Fi	A → B	346.8	239.2	210	268	15.37
	B → A	346.8	201.5	158	240	25.71
Mixed	A → B	693.6	530.8	470	581	33.9
	B → A	693.6	595.8	530	639	31.6

Table 1: TCP results

- **Both Ethernet Scenario**: The average throughput (3) for both directions (A → B and B → A) remains relatively stable after an initial increase in the first second, oscillating between 800-950 Mbps. This stability is indicated by low standard deviations (0.3 Mbps for A → B and 0.92 Mbps for B → A), suggesting a consistent performance over time. The RTT (4) remains low, varying only slightly (within approximately 2.5 ms) throughout the duration of the test, indicating a reliable and stable connection, with only a few outliers (at t=1 and t=5.5).

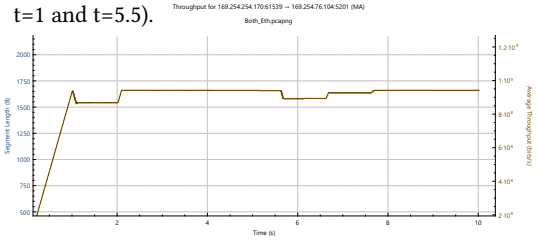

Figure 3: Both Ethernet Scenario Throughput

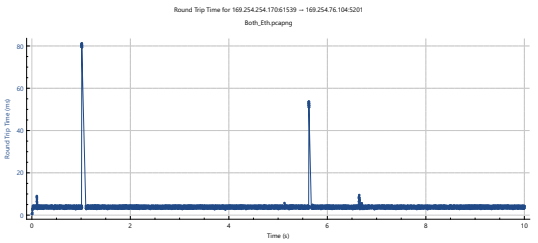

Figure 4: Both Ethernet Scenario RTT

- **Both Wi-Fi Scenario**: Unlike the Ethernet scenario, the Wi-Fi throughput (5) exhibits more variability. It initially increases but then oscillates between approximately 225-275 Mbps after 2 seconds. This oscillation is reflected in the higher standard deviations (15.37 Mbps for A → B and 25.71 Mbps for B → A), indicating fluctuations in the connection quality caused by interference or wireless traffic. The Wi-Fi scenario exhibits the most significant variation in RTT (6), with values ranging from as low as 20 ms to as high as 155 ms, however the RTT is mainly between 20 ms and 75 ms.

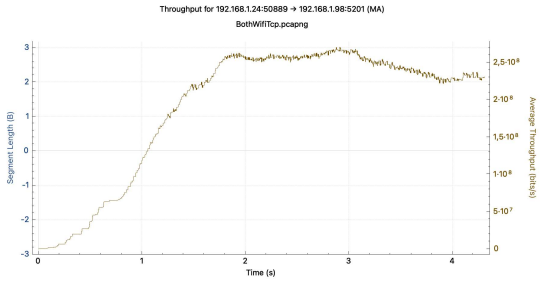

Figure 5: Both Wi-Fi Scenario Throughput

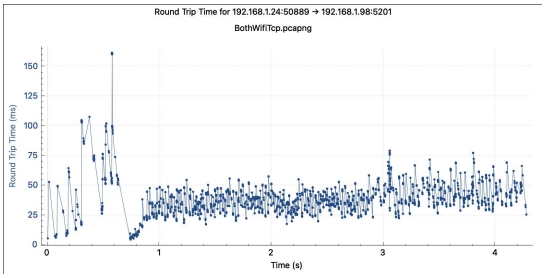

Figure 6: Both Wi-Fi Scenario RTT

- **Mixed Scenario**: In the mixed scenario, the throughput (7) uniformly increases until reaching stability around 600-640 Mbps after the first second. The variation in throughput is relatively minor compared to the Wi-Fi scenario, and this is evident from the standard deviations (around 30 Mbps). The RTT (8) oscillates less compared to the Wi-Fi scenario, as expected. The variation remains mainly within a range between 7 ms and 50 ms. This tests shows that combining Ethernet and Wi-Fi, the connection results in lower RTT values compared to the Wi-Fi to Wi-Fi setup.

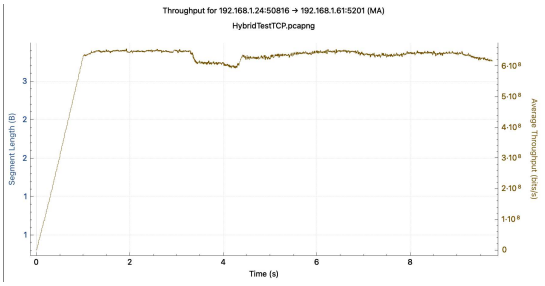

Figure 7: Mixed Scenario Throughput

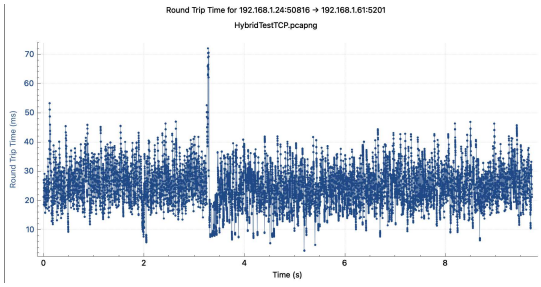

Figure 8: Mixed Scenario RTT

4.2 UDP

Test		UDP: Goodput per flow (Mbps)				
		Prediction	Average	Min	Max	Std
Both Ethernet	A → B	957	955.4	953	956	1.2
	B → A	957	955.2	952	957	1.47
Both Wi-Fi	A → B	372.8	240.9	234	246	3.3
	B → A	372.8	172.9	108	209	31.23
Mixed	A → B	698.8	636	470	581	33.9
	B → A	698.8	637.7	606	671	22.31

Table 2: UDP results

- **Both Ethernet**: The average goodput for UDP in both directions over Ethernet is approximately 955 Mbps, close to the maximum value. With UDP, there is no congestion control or packet acknowledgment, so the transmitter sends packets at the maximum rate supported by the interface without guaranteeing delivery. With a both Ethernet scenario, it is possible to achieve near-maximum bandwidth utilization and basically zero packet loss.
- **Both Wi-Fi**: When transmitting over Wi-Fi, the goodput drops to an average of 200 Mbps, significantly lower than Ethernet. Wi-Fi interfaces cannot immediately accept new inputs due to managing collision avoidance and Wi-Fi overhead, leading to reduced goodput compared to Ethernet. packet loss is more significant in this scenario, especially for transmissions from Host B to Host A. Have been observed superior performance in Wi-Fi-to-Wi-Fi and hybrid transmission scenarios when Host A sends to Host B, with approximately 0% packet loss. In contrast, packet loss occurs in the reverse direction, as previously mentioned. This disparity could be attributed to the superior wireless interface of Host A.
- **Mixed**: In the mixed scenario the average goodput is lower then the both Ethernet scenario (35% lower) but better than both Wi-Fi scenario (almost 3 times higher). When the transmitter is connected via Ethernet, it experiences peak bandwidth utilization (around 955 Mbps) because the AP handles Wi-Fi transmission, freeing the transmitter's network interface from this responsibility. This allows the transmitter to send data over its interface without limitations, as it transmits over Ethernet to the AP, shifting Wi-Fi overhead management to the AP. Therefore, the bottleneck occurs at the final Wi-Fi link between the AP and the receiver. However, packet loss is still evident, particularly when transmitting from Host B to Host A, though it's lower compared to the both Wi-Fi scenario.

packet loss and iperf behavior: iperf3 reports packets loss at the receiver, with varying percentages across tests and scenarios. The percentage of lost packets appears almost random, ranging from 0% to as high as 85%. This variability suggests potential issues with iperf3's packets numbering [2] or reporting. Ethernet transmission generally exhibits lower or negligible packet loss compared to Wi-Fi, where loss percentages are more significant. In general, there's noticeable packet loss when transmitting from Host B to Host A, possibly due to differences in the wireless interface capabilities of the two hosts, as already mentioned.

5 CONCLUSION

In this report was investigated the performance of Ethernet and Wi-Fi connections within Local Area Network (LAN) environments, considering variables such as link capacities and protocol distinctions. The experiments concerned various configurations:

- **Both Ethernet**: This scenario gave the most promising results, with near-maximum goodput achieved in both directions (around 940 Mbps for TCP and 955 Mbps for UDP). The Ethernet connection is stable and consistent as can be seen from the standard deviations (between 0.3 Mbps to 1 Mbps for TCP and between 1.2 Mbps and 1.4 Mbps for UDP) and low packet loss (approximately 1-1.5%).
- **Both Wi-Fi**: In contrast to Ethernet, Wi-Fi performance exhibited fluctuations and dropped to an average of 220 Mbps in TCP and 200 Mbps in UDP due to factors like Wi-Fi overhead (control frames are transmitted in lowest speed), half duplex, collision avoidance, retransmissions, electromagnetic noise. Standard deviations were significantly higher (around 20 Mbps for TCP and around 17 Mbps for UDP), indicating less consistent throughput. Packet loss was also more pronounced, reaching up high values (up to 90%) in some cases.
- **Mixed**: This scenario has performance in the middle between the previous scenarios. Goodput is around 565 Mbps for TCP and 637 Mbps for UDP, more than half of both-ethernet scenario, but packet loss persisted.

In summary, the results show that Ethernet connection allow high performance and stability compared to Wi-Fi. The mixed scenario offers a practical compromise, particularly when some devices require Wi-Fi connectivity due to constraints or location limitations. These findings highlight the importance to adopt new technologies, such as MU-MIMO, to maximize performance. Future investigations could focus on exploring more complex network configurations with a larger number of devices or different access point placements.

6 APPENDIX

The following are the python scripts used to automate the tests

```python
import subprocess, time
import numpy as np

n_times = 10
ip_address = "192.168.1.98" # May vary
bitrates=[]
print("\n")

for i in range(n_times):
        bitrate=subprocess.check_output(f"(
            iperf3 -c {ip_address} | grep
            receiver | tr -s ' ' | cut -d ' '
            -f 7)",shell=True).decode().strip(
            "\n")
        print(f"#{i+1} bitrate: {bitrate} Mbps
            ")
        bitrates.append(float(bitrate))
        time.sleep(1)

print("\n")

min = round(np.min(bitrates),2)
print(f"min: {min} Mbps")
max = round(np.max(bitrates),2)
print(f"max: {max} Mbps")
avg = round(np.mean(bitrates),2)
print(f"avg: {avg} Mbps")
std = round(np.std(bitrates),2)
print(f"std: {std} Mbps")

print("\n")
```

Listing 1: Script for TCP tests

For UDP, the script is the same, with the difference in the iperf3 command, in which has been used:

```
iperf3 -c {indirizzo_IP} -u -l 1472 -b 1G
```

REFERENCES

[1] 2013. 802.11ac data rates and speed. (2013). www.en.wikipedia.org/wiki/IEEE_802.11ac-2013#Data_rates_and_speed

[2] 2021. out-of-order packet reception corrupts sequence numbers. (2021). www.github.com/esnet/iperf/issues/1123

[3] Cisco. 2024. What Is MU-MIMO? (2024). www.cisco.com/c/en/us/products/wireless/what-is-mu-mimo.html

[4] Jon Dugan, Seth Elliott, Bruce A. Mah, Jeff Poskanzer, and Kaustubh Prabhu. 2023. iPerf3. (2023). www.iperf.fr

[5] Huawei. 2022. What Is MU-MIMO? (2022). www.info.support.huawei.com/info-finder/encyclopedia/en/MU-MIMO.html

[6] Jerry Jongerius. 2024. Understanding Wi-Fi 4/5/6/6E/7 (802.11 n/ac/ax/be). (2024). www.wiisfi.com/#wifioverhead