Computer Networks II

Report
Project 1: WiFi Doctor

Due Date: April 1st, 2025

Submitted by:
Gravalos Georgios - Angelos, 2021030001
Kerimi Rafaela - Aikaterina, 2021030007
Stamou Spyridon, 2021030090
Apostolopoulos - Papidas Dimitris, 2021030176

Technical University of Crete
Department of Electrical and Computer Engineering

Introduction

The goal of our project is to design and implement a Wi-Fi network performance monitoring / troubleshooting system, "Wi-Fi doctor". Our focus is the "diagnosis" aspect of a problem in a WiFi environment.

The system consists of:

- **WiFi Sniffer**: We used Wireshark to monitor packets transmitted over WiFi. In the first part of the project, our WiFi cards operated in monitor mode. For the second part, the card would be in Managed Mode to capture the traffic between the AP and the device, but we used the precaptured file that was provided.
- **.pcap Parser**: Using Pyshark, we extracted the necessary fields from each file. We use this information in the Perfomance Monitor.
- **Perfomance Monitor** : The Performance Monitor leverages the metrics extracted by the .pcap Parser. Its role is to calculate key performance indicators such as theoretical throughput, frame loss rate, and RateGap. Additionally, it implements scoring functions to identify whether the Wi-Fi performance is acceptable or degraded. These results are passed to the Performance Analyzer.
- **Perfomance Analyzer** : It makes the final "diagnosis" of the network condition, taking the Performance Monitor's calculations and uses them to rule whether the connection is good or bad. It focuses on identifying potential bottlenecks by analyzing the performance in three key dimensions:
  - *Device Configuration*: e.g., PHY Type, Bandwidth, Short GI, Spatial Streams
  - *Link Performance*: e.g., Signal Strength, Signal-to-Noise Ratio (SNR)
  - *Interference / Network Load*: e.g., Channel congestion, retry rates, and MCS Index behavior
- **Visualizer** : The visualizer takes the outputs of Perfomance Monitor and Perfomance Analyzer and produces bar plots and time series graphs (along with min/mean/median/75th/95th/max statistics) to present the results clearly. It also correlates metrics to help reach a clearer diagnosis.

The project consists of 2 parts:

- **Wi-Fi Network Density** : In this part we did two captures, one for the campus network and one for the home network, across both 2.4 Ghz and 5GHz bands by channel hopping during the capture. By determining a score function we calculate the density. Our observations showed that the home network appeared to be denser.
- **Wi-Fi Network Performance** : In this part we use the precapture file to observe the data frames. By monitoring and analyzing metrics that help us to determine to answer at how good is the perfomance. We found that network connection was relatively good, based on frame loss, rate gap, RSSIs and other factors.

Design

1.1 - WiFi Network Density

To evaluate Wi-Fi network density, we collected two packet capture files using Wireshark in monitor mode: `TUC.pcapng` (campus network) and `MyHome.pcapng` – (home network).

In `WifiNetDensity.py`, we implemented:

- `analyze_ap_signal_strength(pcap_file)`: Parses the pcap file with PyShark, filters beacon frames (type/subtype 0x08), computes Wi-Fi channel characteristics, and calculates a density score using penalties for overlapping APs, RSSI, and PHY type. Prints a per-channel summary with score.
- `visualizer(data1, data2)`: Generates plots showing: Density score per channel, Average RSSI per channel, Number of overlapping APs per channel, Mean density score per capture.

The analysis is applied to all channels in the files, but in this report we will talk about one 2.4GHz and one 5GHz channel, for each network .

1.2: Wi-Fi Network Performance

We used `HowIWiFi_PCAP.pcap` to estimate theoretical downlink throughput,
and `ht_mcs_full_correct.json` to reference MCS data.

The MCS .json file is parsed in `utils.py` into a dictionary:
Keys: (MCS index, bandwidth, spatial streams, short GI)
Values: (data rate, required RSSI)

In `WifiNetPerformance.py`, we implemented:

- `pcap_parser(pcap_file)`: Parses 802.11 data frames between a given AP and device using PyShark and returns a list of parsed frames.
- `performance_monitor(parsed_frames)`: Computes frame loss ratio using retry flags, mean data rate, throughput using: $Throughput = DataRate \cdot (1 - FrameLossRate)$ and Rate Gap using Equation (1) from [5].
- `performance_analyzer(parsed_frames)`: Analyzes frame loss, retry distribution, short GI usage, RSSI distribution. Adds comments per frame covering RSSI quality, retry causes, short GI effects, Rate Gap interpretation, MCS optimization, and potential bottlenecks.
- `visualizer(parsed_frames)`: Generates plots for: Throughput and RateGap (time series), Data rate per RSSI, Throughput per PHY type, bandwidth, GI setting, and MCS index, item Retry count, spatial stream distribution, RSSI range distribution.
  Also prints frame loss percentage and average normalized RateGap.

1.1 - WiFi Network Density

(File `WifiNetDensity.py`)
Our first task was to capture packets in the university network, as well as in our home network, using Wireshark. Having completed this step, we proceed with analyzing their performance.
In both the university and home WiFi networks, we analyzed transmissions from all APs in our vicinity, by accessing their beacon frames. For each packet, we extracted the following attributes: **SSID, BSSID, PHY type, channel, signal strength (RSSI)**.

SNR and noise floor were not available in our packets and therefore were not included in the metrics. Instead, we used signal strength (RSSI), given that it $RSSI = SNR - NoiseFloor$.
Going through all the beacon frames, we grouped the data **by channel**. For each channel, we collected data from the frames transmitted in it: their signal strength, which 802.11 protocol was used and which AP was.
We calculated the density of each channel, by introducing a density equation that incorporated the parameters we introduced above.

$$Density = w_{overlap} \cdot \left(\frac{\#OverlappingAPs}{Overlap_{norm}} \right) + \frac{phy_{score}}{phy_{norm}} + w_{rssi} \cdot \left| \frac{avgRSSI}{RSSI_{norm}} \right|$$

where:
$\#overlappingAPsInChannel$ the number of APs broadcasting in the channel,
phy_{score} the penalty each 802.11 version applies to the function (we will elaborate on this very soon),
$avgRSSI$ the mean RSSI of all transmissions made in each channel.

All these parameters are divided by their correspondent normalization factors, to bring them to a range [0,1]. For example, the worst value of RSSI can be -100 and the best 0, so we set the normalization factor for RSSI to 100.

The higher the density score returned by the density function, the denser the channel. This is why we will now discuss the weights attributed to the parameters.
The weights are always 1 for the 2.4GHz band, so we assign values to the weights of the 5GHz band, in comparison to 2.4GHz.

- $\boldsymbol{w_{overlap}}$: Always 1 for 2.4GHz. For 5GHz, since channel bonding is an option decided by APs, channels bonded together can form 40, 80, even 160MHz channels. The 5GHz band also offers many more non-overlapping channels than 2.4GHz. Apart from this, some 5GHz channels are DFS-enabled [1], so APs can dynamically switch channels to avoid interference. Lastly, protocols that operate primarily in 5GHz employ mechanisms that help multiple devices share the channel efficiently. Given these, we assign the weight for 5GHz to 0.30.

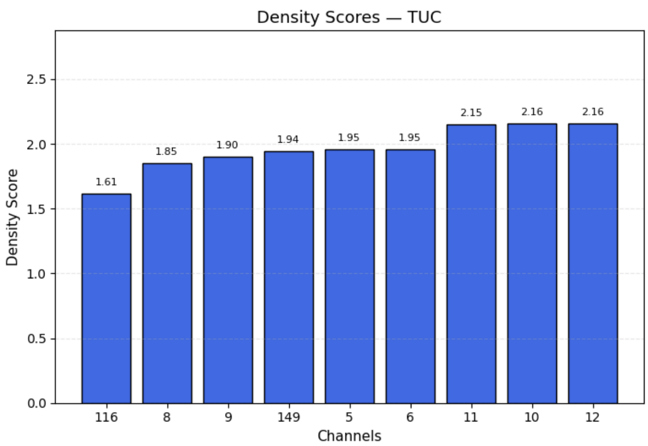
- PHY type has no weight, since it suffices to simply assign a penalty for each protocol used. The penalty assignment is as follows:
  802.11b (4) : 10 (worst penalty), 802.11a (5) : 8, 802.11g (6) : 7, 802.11n (7) : 4, 802.11ac (8) : 2, 802.11ax (9) : 1.

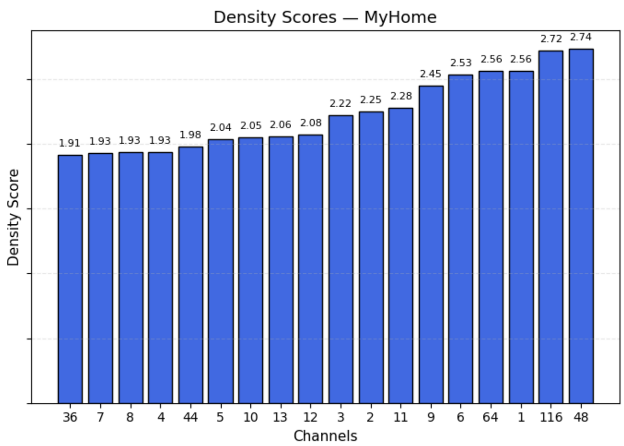
- $\boldsymbol{w_{rssi}}$: Always 1 for 2.4GHz. This is the only weight whose value varies for 5GHz. [2]
  $\mathbf{w_{rssi} = 0.5}$ **for avgRSSI** $\mathbf{> -55}$ **dBm**: 5GHz performs much better than 2.4GHz at the same RSSI values, offering better rates, so we give a bonus to 5GHz.
  $\mathbf{w_{rssi} = 1}$ **for** $\mathbf{-67 <}$ **avgRSSI** $\mathbf{\leq -55}$ **dBm**: The signal is moderate, we make the weight 1 (same as 2.4GHz).
  $\mathbf{w_{rssi} = 1.5}$ **for** $\mathbf{-75 <}$ **avgRSSI** $\mathbf{\leq -67}$ **dBm**: Signal quality begins to drop significantly, especially for 5GHz which offers less coverage due to shorter wavelength. [3]
  $\mathbf{w_{rssi} = 2}$ **for avgRSSI** $\mathbf{\leq -75}$ **dBm**: Signal quality is already severely degraded. Especially for 5GHz, signals of this RSSI indicate severe congestion (or large distance between AP and receiver), and are of no practical use.

We set our cards to monitor mode, and we automated them to hop channels during the capture, so that we can have one .pcap file for the university network (with both 2.4 and 5GHz) and one .pcap file for the home network (again, with both 2.4 and 5GHz).

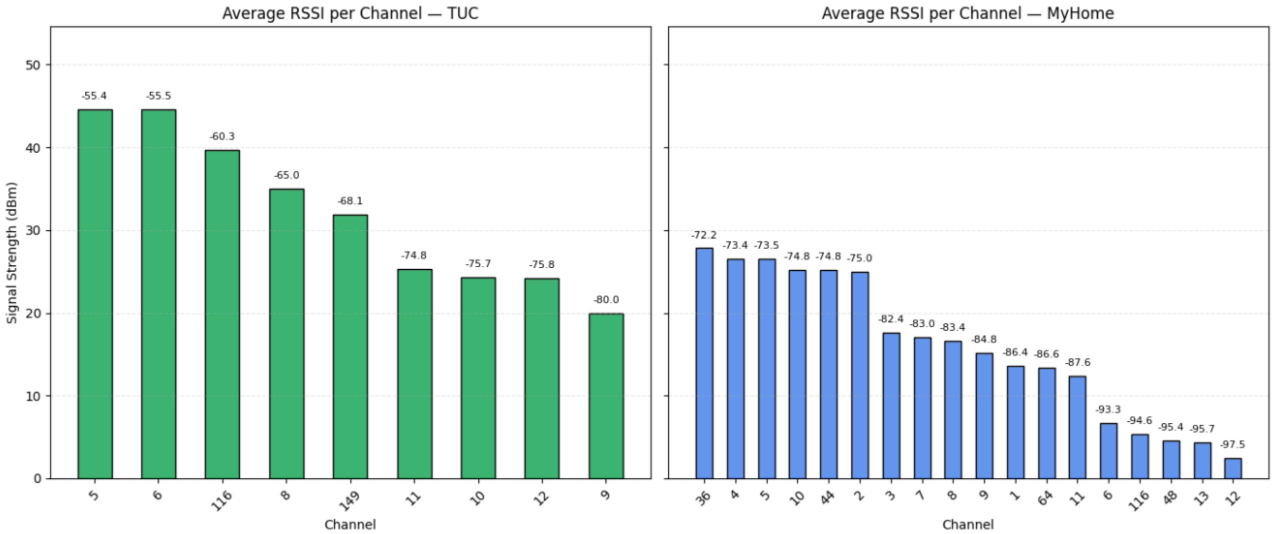
Channel Analysis

We now begin the analysis for the channels in our captured files.
We will analyze Channel 116 (5GHz) and Channel 10 (2.4GHz) for both networks.
We chose specifically these channels, because as we can see below, Channel 116 has the least density score in the university network, whereas in the home network, it is the second most dense. Channel 10, on the other hand, is the most dense in the university network, but in the home network its density score is relatively low.





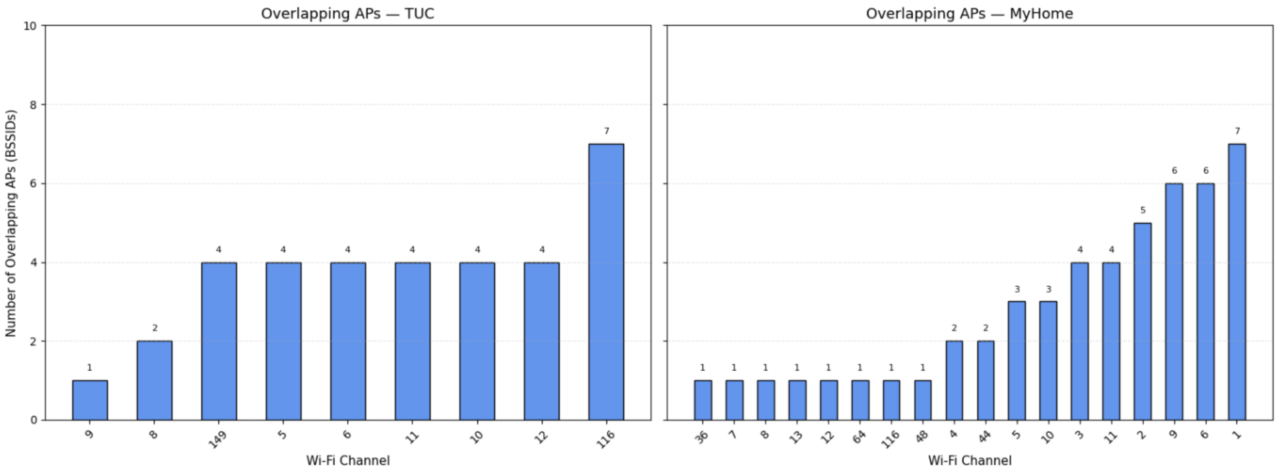
We will dive into the causes for these rankings, which consist of 3 factors : Average RSSI of all transmissions made in these channels, PHY types used in the frames transmitted in the channels, as well as the number of overlapping APs in each of the channels.



Channel 116 : In the university network, there are many points of access to the network, to allow better coverage in all areas of the campus, especially inside buildings.
On the other hand, in the home network, we also capture packets from neighboring apartments (far away from our capture device, with many walls/doors in between) and each apartment usually has a single access point to the home network (the router).
Factoring in larger signal attenuation for 5GHz compared to 2.4GHz, it makes absolute sense that **channel 116 averages -60.3dBm in the university network and -94.6dBm in the home network.**

Channel 10 : In the university network, we're observing weaker RSSI in Channel 10 transmissions (than Channel 116). This difference can be attributed to perhaps 2.4GHz APs being further from our capture device, or less densely deployed than 5GHz APs throughout campus. Apart from that, Channel 10 overlaps with Channels 7,8,9,11 (resulting in reduced overall performance due to congestion), whereas Channel 116 does not overlap with any other 5GHz channels (if using 20MHz width). 2.4GHz is already very crowded and uses legacy 802.11 protocols (such as 802.11b), which do not handle multiple devices in the channel as efficiently as 5GHz.
In the home network, where APs are often the home router, 2.4GHz achieves better RSSI than 5GHz, since the signal does not attenuate as easily as 5GHz due to distance or obstacles between the AP and the device.
All things considered, it is logical that **channel 10 averages -75.7dBm in the university network and -74.8dBm in the home network.**

As mentioned above, RSSI is a key factor in calculating the density score for each channel. 2.4GHz channels (Channel 10, in our case) always have $w_{ssid} = 1$.
In the university network, Channel 116 has an average RSSI of -60.3dBm, so its $w_{ssid} = 1$. In the home network, it has an average RSSI of -94.6dBm, thus is heavily penalized with a $w_{ssid} = 2$.
That said, it makes absolute sense for Channel 116 to score very poorly compared to Channel 10 in the home network. In the university network, however, with both channels having $w_{ssid} = 1$, Channel 116 achieves much less density score than Channel 10. The reason for that lies in different 802.11 protocols being used by the 2 channels, as well as the number of overlapping APs in each channel. We will be discussing these right below.



Channel 116 : 7 overlapping APs in the university network, only 1 AP in the home network.

Channel 10 : 4 overlapping APs in the university network, 3 overlapping APs in the home network.

As explained when we defined $w_{overlap}$ above, it is always 1 for 2.4GHz channels, and 0.30 for the 5GHz channels.
Despite the fact that in the university network, Channel 116 has more overlapping APs than Channel 10, the weight assigned to Channel 116 (due to being 5GHz) penalizes it 3 times less than Channel 10, overall contributing to its smaller density score than Channel 10.
This paradox also appears in the home network, in reverse. Despite Channel 116 having less overlapping APs than Channel 10, it has been so heavily penalized due to its very weak RSSI, that it ends up having a much higher density score than Channel 10.

Channel	TUC	MyHome
1	–	802.11b
2	–	802.11b
3	–	802.11b
4	–	802.11b
5	802.11b	802.11b
6	802.11b	802.11b
7	–	802.11b
8	802.11b	802.11b
9	802.11b	802.11b
10	802.11b	802.11b
11	802.11b	802.11b
12	802.11b	802.11b
13	–	802.11b
36	–	802.11a (OFDM)
44	–	802.11a (OFDM)
48	–	802.11a (OFDM)
64	–	802.11a (OFDM)
116	802.11a (OFDM)	802.11a (OFDM)
149	802.11a (OFDM)	–

Finally, we will take a look at the last factor that contributes to the density score: PHY types used in transmissions in each channel (as printed in the console after analyzing the files).

Wireshark shows all 5GHz frames as 802.11a (OFDM), bcause it labels frames based on legacy 5GHz support, unless it has precise PHY info (MCS index or any flags specifying more details on the PHY protocol, which our captured frames don't provide for Part 1.1).
However, based on the actual throughput capabilities of the networks we analyzed:

- The university network, with large coverage and enterprise-level access points, certainly supports 802.11ac, and very likely 802.11ax.
- Our home network has downlink speeds of up to 300 Mbps, which is impossible with 802.11a, thus confirms 802.11n or higher is in use.

The protocols used in the 5GHz captured frames are presumably 802.11n/ac/ax, which is what we will consider to continue with our analysis, despite the fact that our code assigns weight to 5GHz channels as if they were 802.11a.
[4] Modern PHY protocols (n/ac/ax) introduce Multi-User MIMO and OFDMA (in 802.11ac/ax) to enable simultaneous transmissions to multiple devices, shorter airtime per frame that reduces channel occupancy, improved spectrum efficiency and support up to 8 spatial streams.
These capabilities outperform 802.11b, which occupies more airtime per frame and increases contention and channel congestion.

In our capture, Channel 10 only includes transmissions with 802.11b and Channel 116 only includes transmissions with 802.11a (which we consider to be 802.11n/ac/ax).
The penalty for PHY type is factored in, contributing to the final density score alongside the penalties for average RSSI and overlapping APs.

1.2 - WiFi Network Performance

(File `WifiNetDensity.py`)

In the second part of the project, our task was to estimate the **theoretical downlink throughput** (from the Wi-Fi Access Point (AP) to the device) for all the data frames, and also analyze the network's performance.

With the pcap parser fetching all the required metrics, we began by calculating the **effective throughput** using the *Performance Monitor*.

To do this, we first calculated the **Frame Loss Rate** using Equation (1), where `RetriedDataFrames` were identified by monitoring the **retry** flag and the denominator is the amount of Data frames that we have in the .pcap file.

$$\text{Frame Loss Rate} = \frac{\text{RetriedDataFrames}}{\text{DataFrames}} \tag{1}$$

Once the `FrameLossRate` was known, we used Equation (2) to compute the **throughput** for each frame. This represents the effective throughput that can be achieved without considering frame retries.

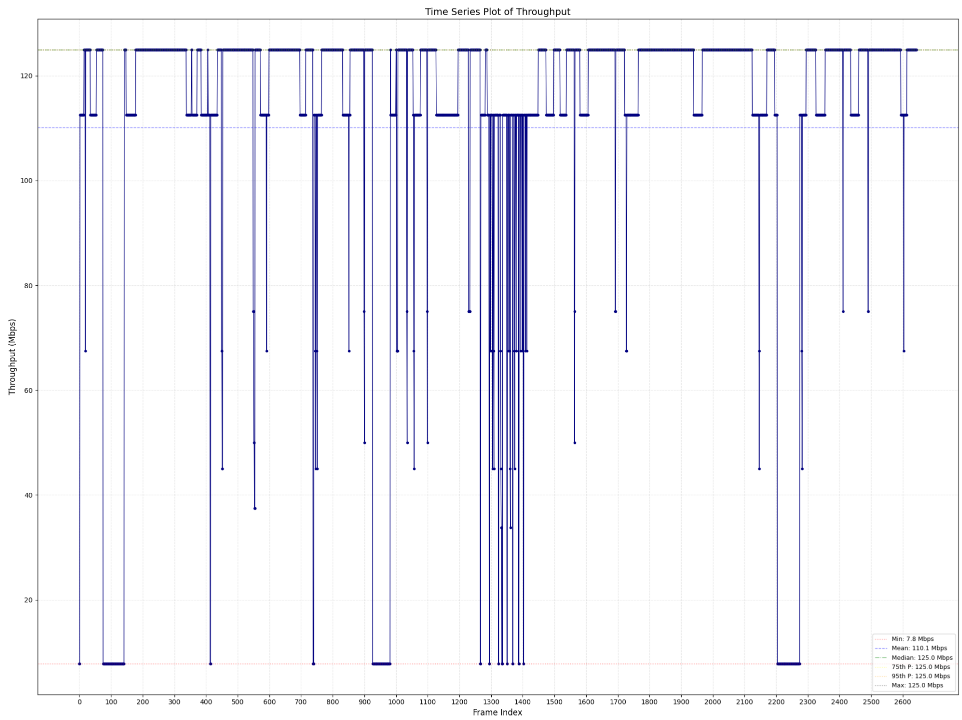
$$\text{Throughput} = \text{DataRate} \cdot (1 - \text{FrameLossRate}) \tag{2}$$

As part of our analysis of the effective throughput for each data frame, we computed various statistics, including minimum, maximum, mean, median, 75th percentile, and 95th percentile.

Having calculated the throughput, we went on to provide further insight on why the throughput is good or bad, by leveraging the metrics we got from the Parser to identify the performance bottlenecks. In particular we use the metrics PHY rate, Bandwidth, Short GI, Data Rate, MCS Index, Signal strength (dBm), and Rate Gap.

Throughput Performance Analysis

We can now represent the throughput we calculated in the performance monitor for our capture in a time series plot including its statistics.

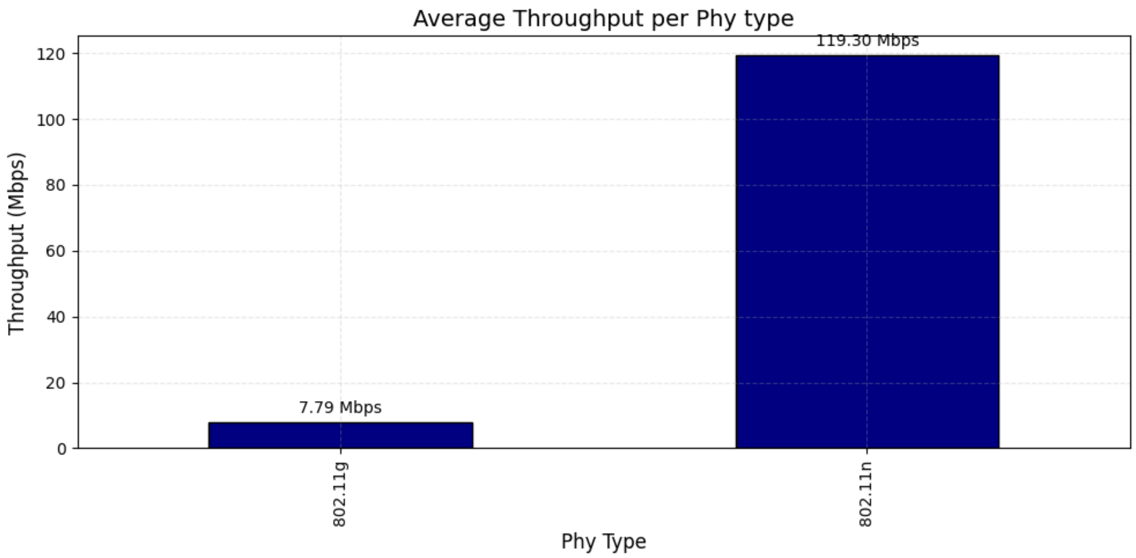


Statistic	Throughput (Mbps)
Minimum	7.79
Mean	110.10
Median	124.96
75th Percentile	124.96
95th Percentile	124.96
Maximum	124.96

Table 1: Throughput Statistics

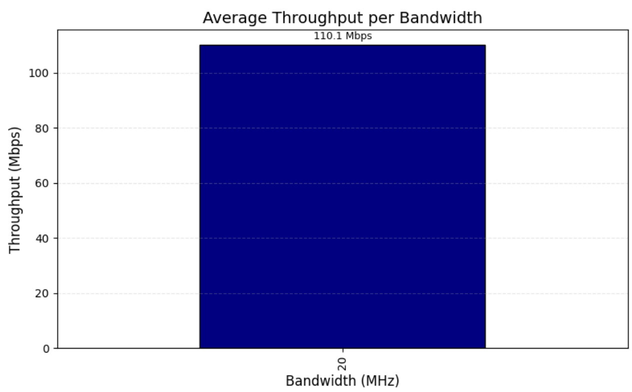
We now begin the analysis for the throughput and the identification for the performance bottlenecks for the pre-captured file.
We will focus on which the metrics that affect throughput are, and how they affect it.

PHY Type: For our purpose, the PHY type refers to the specific physical layer protocol used to transmit the 802.11 frames and it's strongly related to throughput. Essentially, it defines the upper limit for our performance.



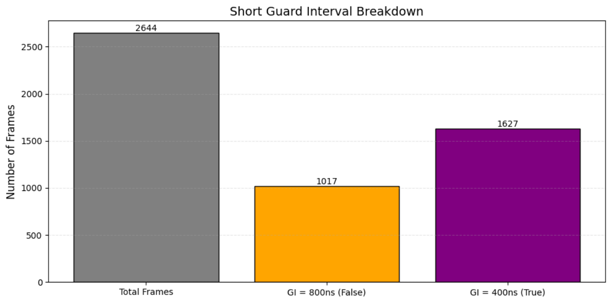
As we observe in the plot, 802.11n has much higher average throughput (119.30Mbps) compared to 802.11g (7.79Mbps) as expected, since they have a huge leap in theoretical speed (600Mbps vs 54Mbps). While there is a variety of different standards, for the data traffic between the specific AP and device, only the above two make an appearance.

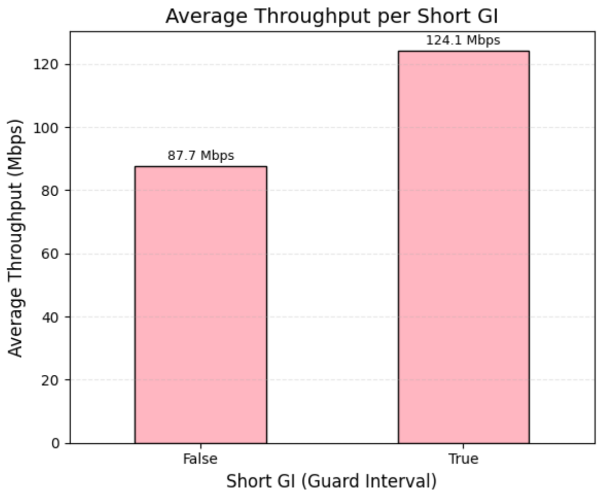
Bandwidth: The width of the frequency channel used to transmit data directly affects the maximum achievable throughput. Wider channels permit higher potential throughput but come with tradeoffs, like increased interference risk and reduced range, since higher frequency signals attenuate faster.



In the plot above, only a single bandwidth (20MHz) is present, with an average throughput of 110.1Mbps. This is not representative of a typical capture, where we would expect to see more bandwidths.

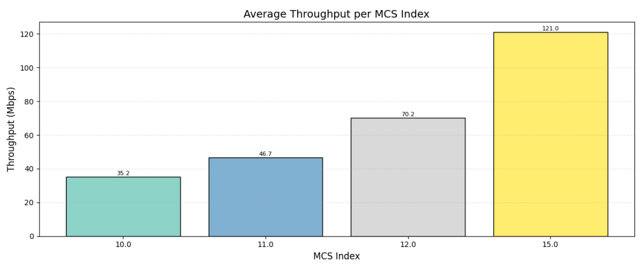
Short GI: Short GI has two values - True (400 ns) and False (800 ns). A lower guard improves the data rate (and therefore the throughput) and allows more information to be received as a result of that. However, it trades throughput for reliability. The guard interval is supposed to keep symbols from interfering with eachother, due to multipath propagation. A 400ns (short) interval between frames, might cause the receiver to misinterpret overlapping bits (previous symbols interfering with the next ones), and lead to higher BER and more retransmissions.





The left plot shows the distribution of Short GI (T/F) for the frames in the file. The right plot demonstrates that enabling Short GI (400 ns) leads to increased throughput (124.1 Mbps, compared to 87.7 Mbps when it is disabled). This confirms the theoretical expectation that a shorter guard interval can improve data rates.

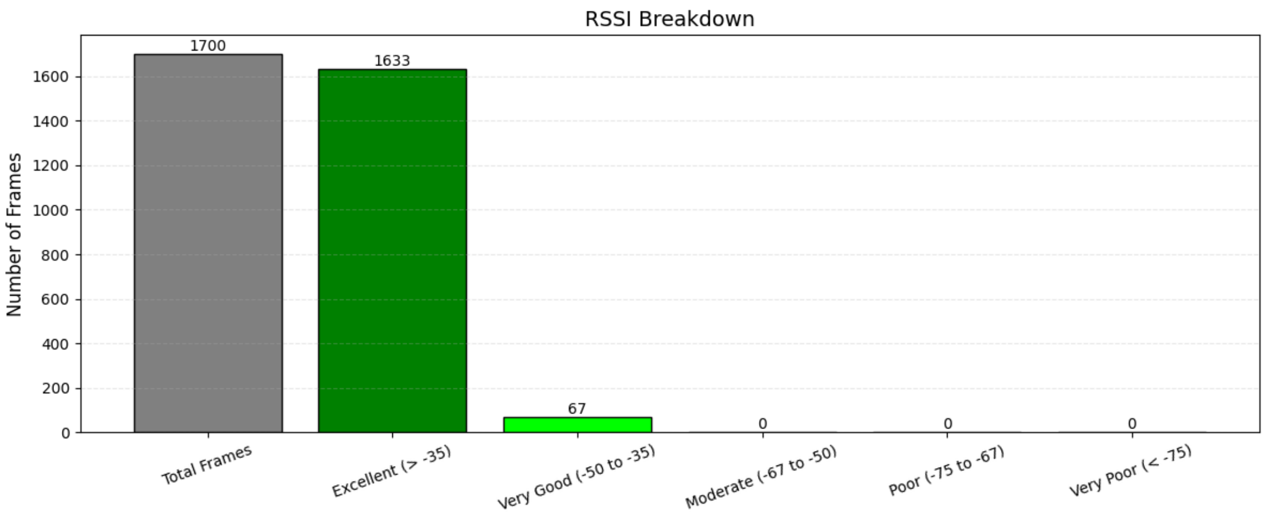
MCS Index: This is a value that corresponds to the data rate, modulation type, number of spatial streams, and coding rate used during transmission. Higher MCS indices correspond to better modulation methods and reduced error correction, resulting in increased throughput. The only requirement for using a higher MCS index is sufficient RSSI to support it.

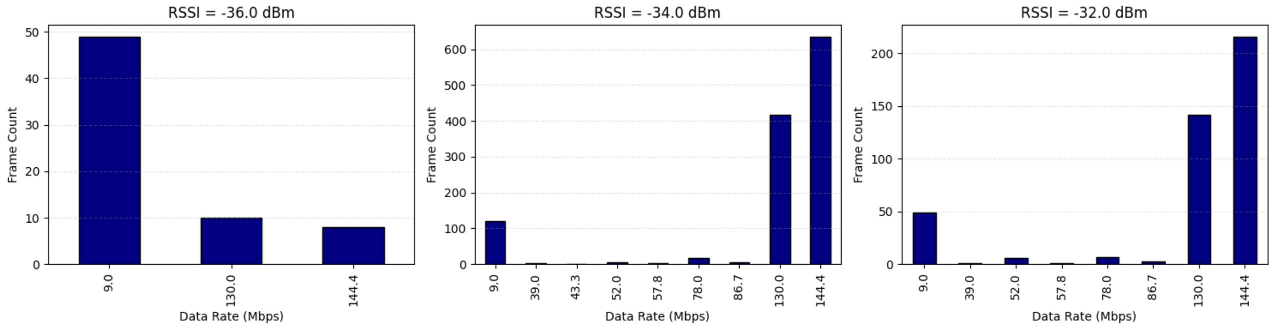


The figure gives us information on the average throughput per MCS index found in the frames. The higher the MCS index, the better the throughput, as expected.

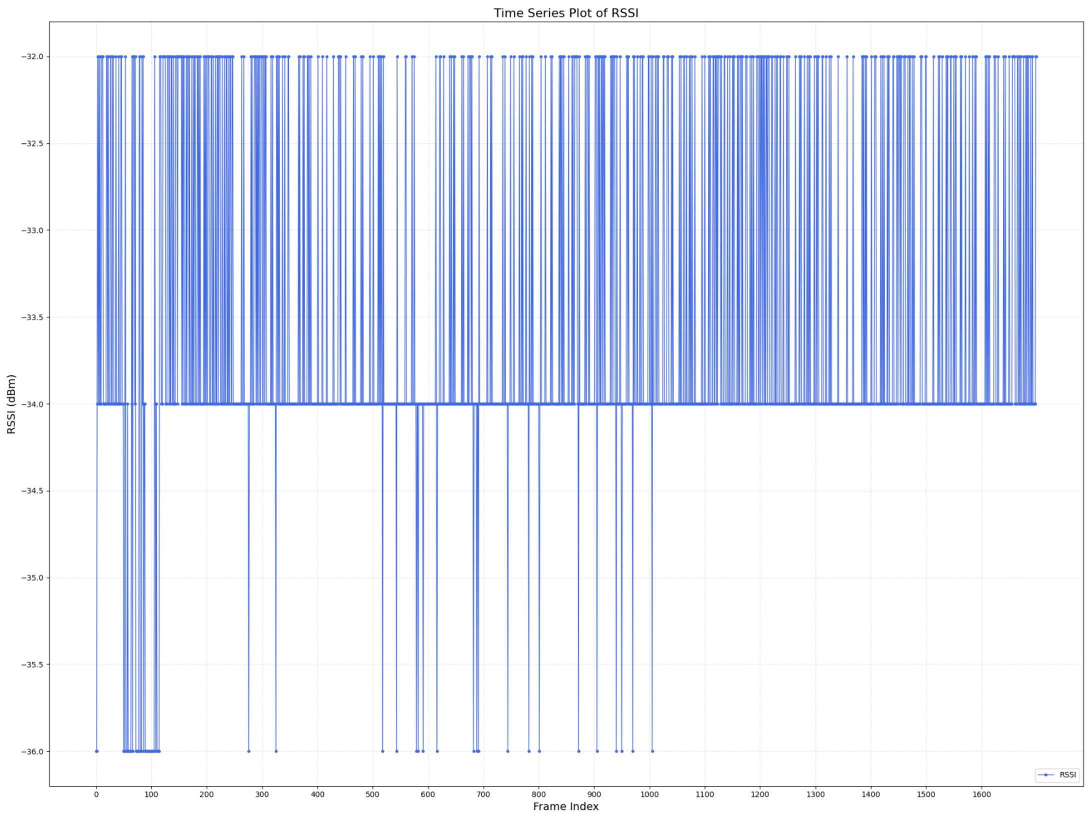
RSSI: RSSI directly influences network throughput, as it determines the quality of the wireless connection. Higher RSSI values-the ones closer to zero- allow for higher MCS indices, as mentioned above, and thus higher throughput.
Based on RSSI measurements, signal quality can be categorized into different ranges, each corresponding to varying levels of connectivity and data transmission efficiency. The following table outlines these ranges and their implications for network performance.

RSSI Values (dBm)	Signal Strength	Expected Performance
> -35	Excellent	Maximum throughput, highest MCS, ideal for high-speed data transfer and low latency.
$-35 \leq RSSI < -50$	Very Good	Reliable connection with high data rates. Minimal retries or packet loss may occur.
$-50 \leq RSSI < -67$	Moderate	Throughput may be affected; some speed reduction and occasional packet loss are likely.
$-67 \leq RSSI < -75$	Poor	Significant drop in throughput, increased retransmissions, and higher latency. Lower MCS rates may be used.
$RSSI < -75$	Very Poor / Unusable	Unstable or lost connection (closer to -100dBm). Severe packet loss and little to no data transfer.



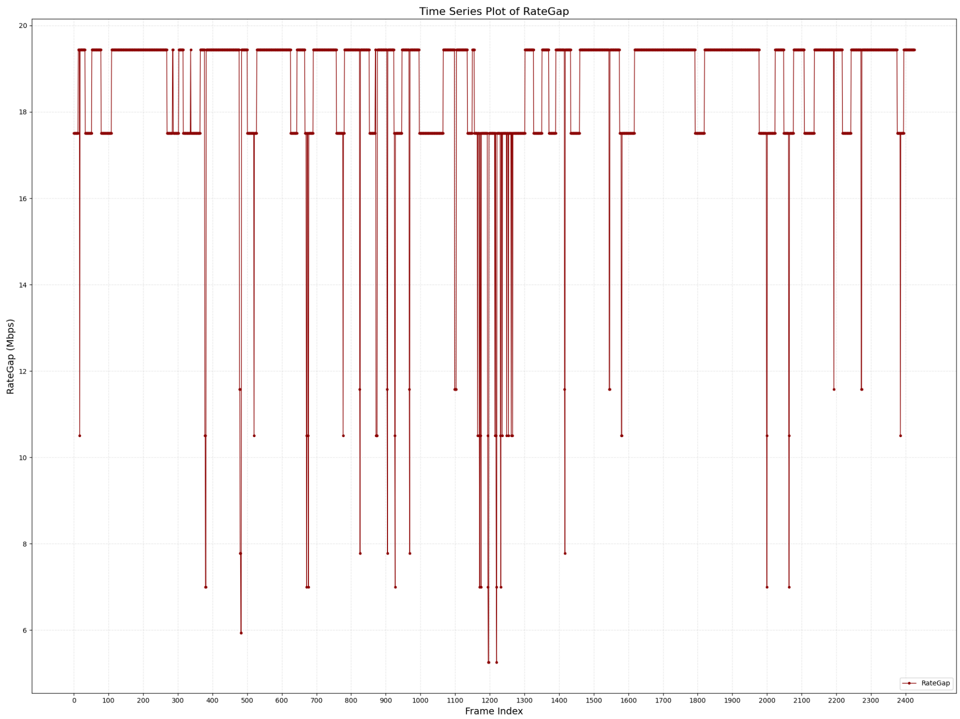


The first plot visualizes the distribution of RSSI values present in the pre-captured file. The second plot illustrates the data rate distribution corresponding to each RSSI value, providing insight into the correlation between signal strength and achievable throughput. As expected, the most frames with increased data rate are observed in the less negative RSSI values.



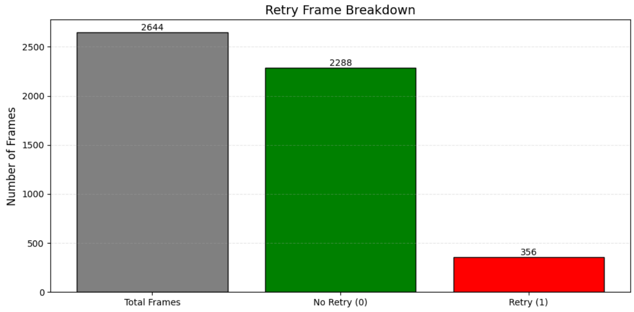
This is a complementary timeseries plot of the RSSI values to further demonstrate the distribution of RSSI values through the file.
As we can observe, the only values that exist are -32, -34 and -36dBm, therefore the signal strength is more than just acceptable. As we will explain in the conclusion, these values are way below the threshold of what we can consider "bottleneck" to throughput.

Rate Gap: The rate gap is calculated as the difference between the expected data rate (based on the table of [6]) and the throughput that was calculated in Performance Monitor. The rate gap is expressed in Mbps, and was later normalized to a percentage to facilitate comparisons. A larger rate gap indicates a lower data transmission rate. To visualize the output of the calculated rate gap, we plot a time series.



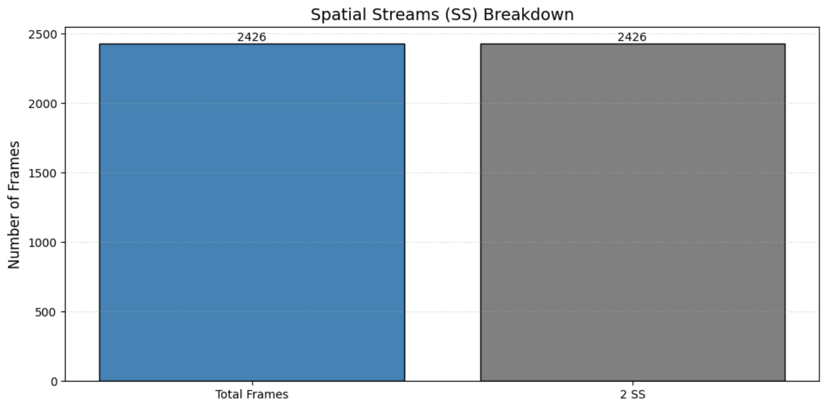
Rate Gap values range from 6 to 19.5 Mbps. The upper limit values may occur due to several factors, such as network congestion, retransmissions etc.

Retry Rate: The percentage of packets that had to be retransmitted because the original transmission failed (e.g., due to interference, collisions, or weak signal). The retry rate and throughput are inversely related, meaning that as the retry rate increases, the throughput tends to decrease. This is because retries consume airtime that could otherwise be used for transmitting new data, thereby reducing the overall throughput.



The above plot is a visual representation of the distribution of frames between retransmissions and successful transmissions. The actual value, calculated in the performance monitor, is 13.46%.

Spatial Streams : The number of spatial streams can range from 1 to 8 in newer Wi-Fi standards. A higher number of spatial streams enhances throughput by enabling parallel transmission across multiple antennas.



In the plot, we observe that in this capture only 2 spatial streams are utilized, which suggests that the system supports MIMO (Multiple Input Multiple Output).

In a typical capture, there may be more spatial streams, further improving throughput capabilities.

Conclusion of Performance Analysis
Since all of the metrics mentioned above affect throughput, they can all be potential bottleneck factors. We reach the following conclusions:

- **PHY type:** Older standards set an upper limit to throughput, no matter how much throughput the other metrics might offer.
- **Bandwidth:** Even with multiple spatial streams (MIMO), a 20MHz channel severely limits the maximum achievable throughput because the total available bandwidth is the main bottleneck.
- **Short GI:** It can become a bottleneck when set to false (800ns), in the sense that it reduces performance for the reasons mentioned above.
- **MCS Index:** Lower MCS index indicates to worse Modulation and Coding Scheme that can be cause of throughput bottleneck as mentioned above.
- **RSSI:** $-67dBm$ is the minimum value for applications that require reliable, timely delivery of data packets. Values below this threshold begin to cause increased packet loss, frequent retransmissions, lower modulation and coding rates (lower MCS), and of course introduce limits to throughput. These issues become more severe the closer we get to -100dBm.
- **Rate Gap:** Since the rate gap reflects the difference between the expected transmission rate and the actual observed rate, a rate gap greater than 30% strongly suggests that the link is underperforming. Therefore, we consider values over 30% to be a clear sign of a throughput bottleneck.
- **Retry Rate:** A high retry rate is often a strong indicator of interference, poor signal quality, or congestion on the channel, as mentioned in the analysis. This is why we believe that 30% is the maximum retry cate we can accept, before we consider it a throughput bottleneck factor.
- **Spatial Streams:** The fewer spatial streams, the worse the throughput. We can consider the bottleneck to be 1 SS (no MIMO), where there is no parallel data transmission.

References & Citations

References

[1] ASUS community FAQ thread
https://www.asus.com/support/faq/1045936/

[2] Understanding RSSI, MetaGeek
https://www.metageek.com/training/resources/understanding-rssi/

[3] 2.4 GHz vs. 5 GHz vs. 6 GHz: What's the Difference?, Intel
https://www.intel.com/content/www/us/en/products/docs/wireless/2-4-vs-5ghz.html

[4] Wi-Fi: Overview of the 802.11 Physical Layer and Transmitter Measurements
https://www.tek.com/en/documents/primer/wi-fi-overview-80211-physical-layer-and-tran
What are 802.11 Standards?
https://www.cbtnuggets.com/blog/technology/networking/when-to-use-802-11-a-b-g-b-nc-wifi-standards

[5] Pefkianakis et al. "Characterizing Home Wireless Performance: The Gateway View", IEEE INFOCOM 2015

[6] 802.11n and 802.11ac / MCS, SNR and RSSI
https://www.watchguard.com/help/docs/help-center/en-US/Content/en-US/WG-Cloud/Devices/access_point/deployment_guide/Checklist_Charts/mcs_80211n_ac.pdf