ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Μελέτη και ανάπτυξη μηχανισμών βελτιστοποίησης του ρυθμού μετάδοσης σε συστήματα επικοινωνίας, με τη χρήση μη επανδρωμένων εναέριων οχημάτων

Μεταπτυχιακή Διπλωματική Εργασία

Εμμανουήλ Μαρούλης

Επιβλέπων:Καθ. Αθανάσιος Κοράκης

Βόλος, 2023

UNIVERSITY OF THESSALY
SCHOOL OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Study and development of rate optimization mechanisms in communication systems, through the use of Unnamed Aerial Vehicles (UAVs)

Master's Thesis

Emmanouil Maroulis

Supervisor: Prof. Athanasios Korakis

Volos, 2023

Approved by the examination Committee:

Supervisor	**Korakis Athanasios** Professor, Department of Electrical and Computer Engineering, University of Thessaly
Member	**Argyriou Antonios** Associate Professor, Department of Electrical and Computer Engineering, University of Thessaly
Member	**Bargiotas Dimitrios** Professor, Department of Electrical and Computer Engineering, University of Thessaly

Date of approval: 28-02-2023

Acknowledgements

I would like to express my gratitude to my supervisor Prof. Athanasios Korakis for giving me the opportunity to work in such an amazing project, and for his invaluable advice throughout this journey. I am also grateful to Dr. Konstantinos Chounos, for his extremely useful help and guidance, his ideas and feedback on this project. I would also like to thank PhD candidate Panagiotis Tzimotoudis for his technical assistance and advice that helped me overcome many obstacles on this project. Last but not least, I would like to thank all the people at NITLab for the collaboration and fun moments in the office during this time.

Abstract

The growth of large scale aerial systems that transmit large amount of data (video streaming, images etc.) indicates the need of stable long range network connectivity between UAVs and Ground Stations (GS). This thesis explores the performance and limits of wireless network coverage between a UAV and a Ground Station (GS) using Wi-Fi technology. We analyze and evaluate the performance and energy footprint of a system prototype that consists of a commercial UAV and a GS with a IEEE802.11ah module and a Raspberry Pi board, also we present and discuss a partially developed concept of a directional antenna along with a steering mechanism on the UAV. We show that the system can outperform other conventional Wi-Fi technologies at distances above $400m$, achieving a throughput of $\sim 3Mbits/sec$, while consuming small amounts of energy and keeping a consistent connection when the UAV is moving.

Περίληψη

Η αύξηση των εναέριων συστημάτων μεγάλης κλίμακας, τα οποία μεταδίδουν μεγάλο όγκο δεδομένων υποδεικνύει την ανάγκη ανάπτυξης σταθερών μηχανισμών δικτύωσης μεγάλης εμβέλειας μεταξύ εναέριων μέσων και σταθμών εδάφους. Σε αυτή την εργασία μελετάμε την απόδοση και τα όρια ενός δικτύου Wi-Fi που χρησιμοποιείται για την επικοινωνία ενός μη επανδρωμένου εναέριου οχήματος (drone) και ενός σταθμού εδάφους. Αναλύουμε και αξιολογούμε την απόδοση του δικτύου αλλά και την ενεργειακή κατανάλωση του πρωτοτύπου συστήματος που αναπτύξαμε. Το σύστημα αποτελείται από ένα drone και ένα σταθμό εδάφους (laptop), στα οποία έχουμε ενσωματώσει υπολογιστές Raspberry Pi και κάρτες Wi-Fi, οι οποίες υποστηρίζουν το πρωτόκολλο IEE802.11ah, επίσης παρουσιάζουμε ένα μερικώς αναπτυγμένο μηχανισμό που περιλαμβάνει μια κατευθυντική κεραίας μαζί με ένα μηχανισμό πηδαλιούχησης στο drone. Δείχνουμε ότι το σύστημα έχει καλύτερες επιδόσεις σε σχέση με άλλες συμβατικές τεχνολογίες Wi-Fi σε αποστάσεις μεγαλύτερες των $400m$, πετυχαίνοντας ταχύτητες μετάδοσης της τάξης των $\sim 3Mbits/sec$ έχοντας χαμηλή καταναλώση ενέργειας, κρατώντας σταθερή σύνδεση μεταξύ των δυο μέσων ενώ το drone βρίσκεται σε κίνηση.

Contents

1 Introduction

Nowadays Unmanned Aerial Vehicles (UAVs) have become a very interesting field of study for scientist and engineers, people use them to tackle many kinds of problems in indoor and outdoor environments. Some use cases for UAVs are computer vision applications [1], security and surveillance [2] and search and rescue systems [3]. There is a variety of public UAV software and off the shelf hardware to choose and experiment with. Most of these systems provide telecommunication modules, for basic communication with a ground station or a radio. Many applications demand high capacity and low latency communication links between nodes in order to transfer lots of data fast and efficiently, therefore users have to find custom solutions or purchase black box devices to solve this problem. In this thesis, we explore the field of communication for UAVs and experiment with a Wi-Fi technology to improve the connection link between a UAV and a ground station.

1.1 Problem statement & Contributions

The recent increase of unmanned aerial vehicles (UAVs) that cooperate in autonomous missions (swarms) [4] demand robust (inter)communication capabilities between nodes. More specifically, low power and long range communication mechanisms are very important, because UAVs operate mostly in outdoor environments and cover a wide range of space in a single autonomous mission. Due to the energy and performance limitations, most communications mechanisms are either very simple and cannot achieve a remarkable throughput or they are custom for specific UAVs and consume a lot of energy. Some remarkable communication protocols used for low power communication are:

- Zigbee[5] with a transmission distance of $\sim 10 - 100\ m$ and a data rate of $\sim$

$250kbit/s$.

- LoRa [6] with a transmission distance of $\sim 4.8km$ and a data rate of $\sim 50kbit/s$.
- LTE-M [7] with a transmission distance of $\sim 10km$ in rural areas and a data rate of $\sim 1Mpbs$.

The above technologies are mostly used on the Internet of Things (IoT) concept, and their data rate does not suffice to transmit a lot of data. In an effort to tackle this challenge, we explore the Wi-Fi communication area, by deploying a specific Wi-Fi protocol in order to achieve higher throughput during autonomous missions, with a minor energy cost. For our study, we use a commercial drone NXP KIT-HGDRONEK66, an omnidirectional antenna and a Raspberry Pi 4b+ (RPI). Also, we explore and evaluate the integration of a directional antenna on the UAV. First, we design the servo rotating mechanism for the antenna from scratch, test and evaluate its power consumption. Then, we write an application inside the drone's software stack (PX4), this application is responsible for commanding the servo mechanism based on Global Positioning System (GPS) information from two nodes in our network (UAV, ground station). Lastly, we design a communication software for information diffusion between two or more nodes. Our contributions are summarized as follows:

- We improve the throughput over a Wi-Fi connection by using IEEE802.11ah protocol between a UAV and a ground station with conventional low-cost devices that consume little energy compared to other Wi-Fi technologies.
- We test and evaluate the implementations in a real environment, achieving high performance compared to energy consumption.
- Finally, we explore (numerically) the capabilities of IEEE802.11ah over long distances and dynamic environments.

1.2 Related work

The authors in [8] propose an IEEE802.11ah-based Internet of Drones architecture for surveillance and remote control. In this work, among others, they create a simulation of drones that communicate using IEEE802.11ah protocol at distances of $1km$ between nodes. The authors in [9] investigate the throughput, and latency of drone-to-drone and drone-to-ground communications using different wireless technologies like IEEE802.11ac, LTE-A and 5G, we are interested only in the Wi-Fi part of the experiments. Their results show that the drone-to-ground communication throughput using Wi-Fi technology suffers huge drops and approximates $0Mbits/sec$ when the distance between the two devices reaches $\sim 200m$. The authors in [10] studied the coverage range and energy consumption of Wi-Fi networks using UAVs. Their work consisted of extensive theoretical research on the 802.11 protocol family, its range of coverage, and a real experiment with a UAV. One of their experiments was the evaluation of 802.11g/n protocols using a UAV as an Access point and two endpoints acting as transmitter and receiver in a real environment, the UAV reached altitudes of 10 and 20 m respectively with maximum coverage of $\sim 50m$ having a throughput less than $1Mbits/sec$. In [11], it is stated that live time video streaming applications require $4 - 6Mbits/sec$ throughput and signal above $-60dBm$, their system consists of a UAV as an Access Point and a ground node as a station, but the implementation works for distances below $\sim 20m$.

1.3 Overview of the content

This thesis is organized in X chapters as follows:

Chapter 2: Describes the hardware and software used in this thesis. It provides information about the IEEE802.11ah protocol, module and its properties, the directional antenna's specifications and the drone setup.

Chapter 3: Describes the software we developed. It presents the software running on the RPI, the communication system between nodes and the antenna steering algorithm.

Chapter4: Presents the experimental results, corresponding to our work contributions.

In particular, it provides the performance achieved by our design concerning throughput optimization and energy footprint.
Chapter 5: Concludes the thesis.

2 Background

In this chapter, we present theoretical and technical information about the system's components, we also give a general overview of the UAV's configuration.

2.1 IEEE802.11ah

IEEE 802.11ah is a wireless networking protocol published in 2017[12] as an amendment of the IEEE 802.11(cite 802.11) family. In contrast to traditional, Wi-Fi technologies like IEEE802.11ac/b/g/n (cite those) that operate in 2.4/5Ghz, it uses 900 MHz license-exempt bands providing extended range Wi-Fi networks. Its low power consumption can antagonize Bluetooth Low Energy is some use cases[13].Also, it is shown that this protocol can produce a data rate up to $15 Mbits/sec$ and a service distance of, $\sim 1\ km$ [14], more information about the hardware2.2.2 and software2.4.2 we used are presented in the next sections. The above observations make IEEE 802.11ah a great candidate for the concept of Internet of Things (IoT). Although its characteristics are promising, there is little or no information about its real performance in dynamic environments with high mobility between 802.11ah devices.

2.2 Hardware

This section describes each of the components we deployed on the UAV and their technical attributes. We used the NXP KIT-HGDRONEK66 [15]UAV, because it is easy to set up, maintain and repair. The UAV is powered by a Li-Po battery that provides $14.8V$ and has a capacity of $11000Mah$. The estimated flight time of the vehicle without external hardware is $\sim 15min$, which is enough for testing purposes. There are options to increase or maintain flight time with our external components by using larger capacity batteries, or use a different battery for the rest of the components. In the evaluation section4, we show that a full scale experiment needs $\sim 5min$, thus we decided not to modify the initial UAV setup.

Figure 2.1: UAV

2.2.1 Flight Management Unit FMU

The main computer of our UAV is the RDDRONE-FMUK66 FMU[16], which is developed by NXP, and it follows the open standards for drone hardware (Pixhawk FMUv4). Also, it runs PX4 Autopilot, the standard for industrial-grade drones, which gives us the freedom to customize our vehicle's hardware and software, more information about PX4 are provided in 2.4.1. This FMU has 6 PWM outputs, and all of them are attached directly to the FMU, there is no separate IO board for the critical outputs (motor controllers), thus all the outputs are considered MAIN outputs, leaving no AUX (Auxiliary)/non-critical outputs for the user.

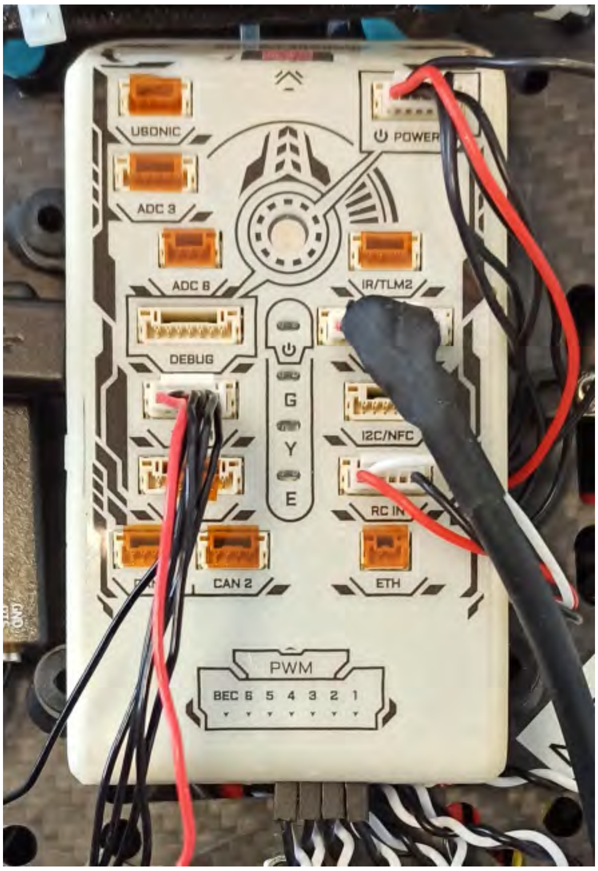


Figure 2.2: Flight Management Unit

2.2.2 IEEE802.11ah device

In our implementation, we use the Alfa Network AHMB7292S module [17] 2.3, it contains the Newracom™ NRC7292 chipset, and offers a variety of modulations (OFDM with BPSK, QPSK, 16QAM, 64QAM). Also, it supports three different channel bandwidths $1/2/4MHz$ and produces data rates up to $\sim 15Mbits/sec$, depending on configuration and conditions.

Figure 2.3: IEEE 802.11ah AHMB7292S module

The AHMB7292S module is mounted on a Raspberry Pi 3b+ (Rpi), which includes all the necessary drivers (see software section) to operate the module. The two devices are powered from the same battery that powers the UAV, thus reducing the flight time of the UAV, this is a tradeoff we accept and evaluate in 4. We also use a DC-DC step down converter in order to provide the $5V$ voltage needed by the RPI to function properly, below we present the devices mounted on a UAV.

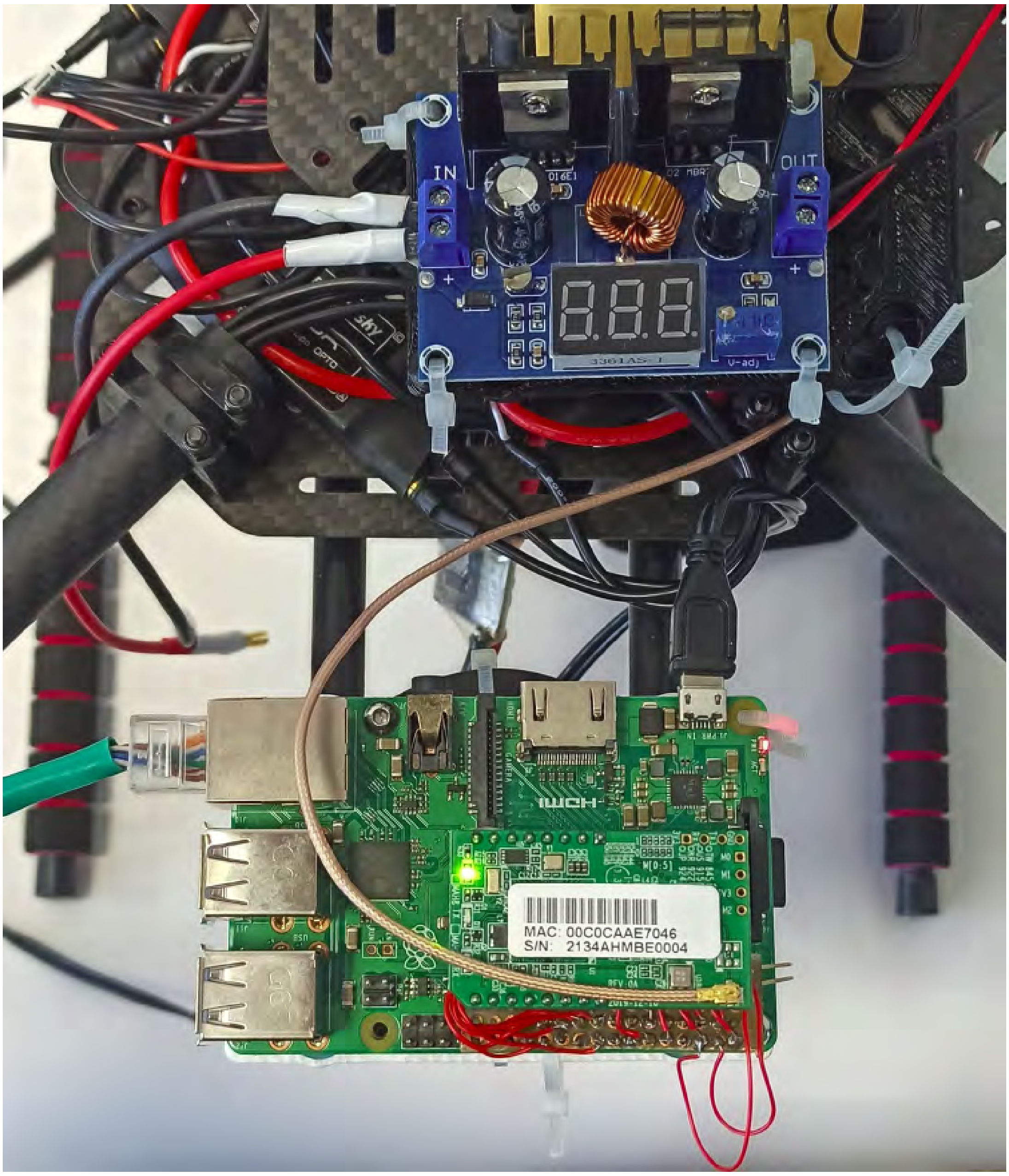


Figure 2.4: Wi-Fi system mounted on a UAV

2.2.3 Antennas

In our implementation, we use two types of antennas, omnidirectional[18] and directional[19]. A commercial, off the shelf omnidirectional antenna is used by our ground station, in order to support many clients from different geolocations. More specifically, the antenna is passive and supports a frequency range of $868 - 915 Mhz$ (ISM bands) and its Gain is 3dBi.

Figure 2.5: Omnidirectional antenna.

Figure 2.6: Directional antenna.

The Taoglas TS.89.4113 Sighunter[20] directional antenna is used by the UAV along with the steering mechanism, in order to improve throughput at long ranges. It operates at either $868MHz$ or $915MHz$ frequencies, also it has greater than 80% efficiency and $3.5dBi$ gain at $868MHz$ and 80% efficiency and $2.7dBi$ gain at $915MHz$.

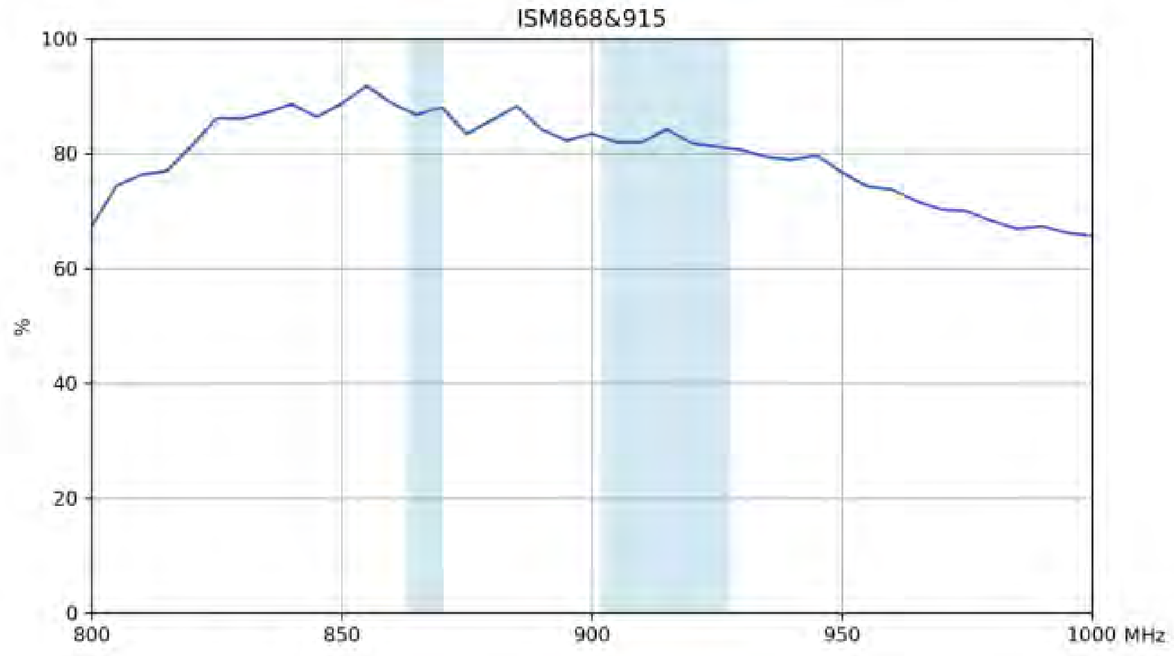


Figure 2.7: Efficiency.

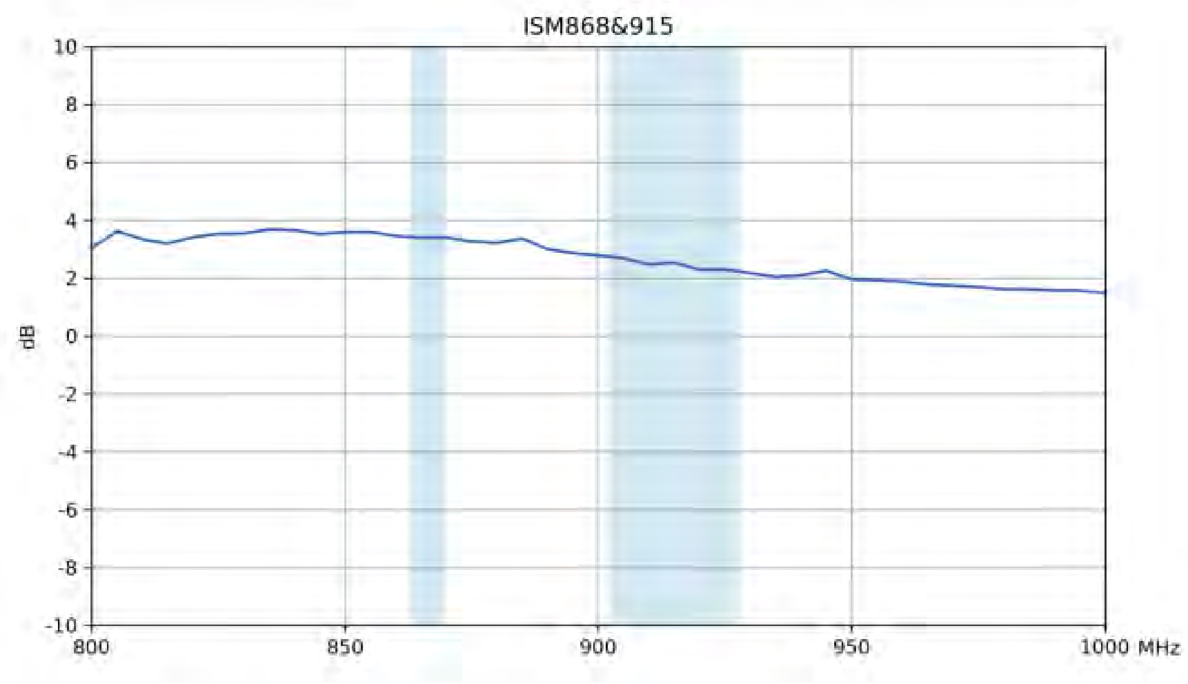


Figure 2.8: Peak gain.

Below, we provide the 3D radiation pattern, at 868/915MHz.

Figure 2.9: Radiation pattern at 868MHz.

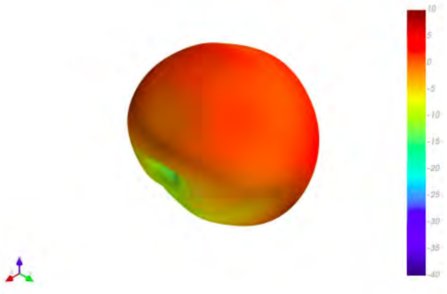


Figure 2.10: Radiation pattern at 915.

2.3 Steering mechanism

The steering mechanism consists of a commercial servo, which operates at $4.8 - 6.0V$ and has a 180^o range of motion. This servo is directly connected to the DC-DC step down converter mentioned above, and it operates at, $5V$ providing a torque of $\sim 2.79kg * cm$.

Figure 2.11: Servo motor

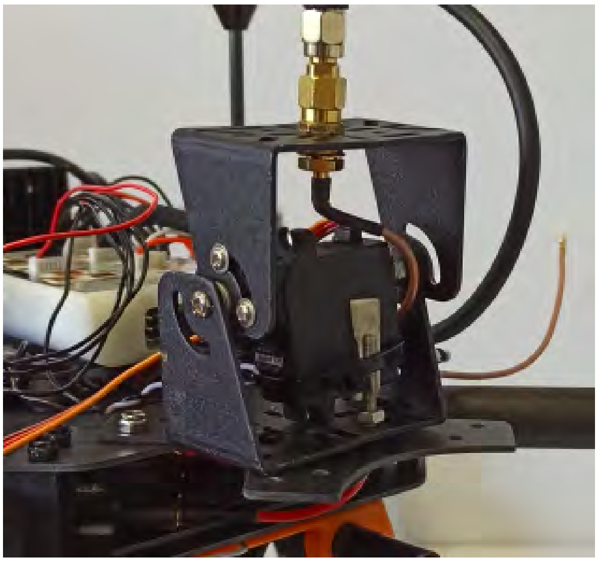


Figure 2.12: Servo mechanism

The mechanism is mounted on the back side of the UAV, using custom metal bases. We also created a 3D stabilizing backbone base for the antenna in order to prevent collision between the antenna and propellers of the UAV.

Figure 2.13: Directional Antenna mounted on a UAV

2.4 Software

2.4.1 PX4 software stack

PX4 is an open source autopilot flight stack, it is used to control many different types/frames of vehicles, and it supports custom application deployment. It is a core part of a broader platform that includes drone hardware and a ground station software, QGroundControl[21]. We use QGroundControl in this thesis to plan and execute autonomous missions, inspect/change the vehicle's state in real time, and for logging. PX4 has two main components, the flight stack which is an estimation and flight control system, and the middleware which consists primarily of device drivers for embedded sensors, communication with the external world (companion computer, GCS, etc.) and the uORB [22] publish-subscribe message bus. We will not interfere with the flight stack in this project. Below, we provide an overview of the software architecture.

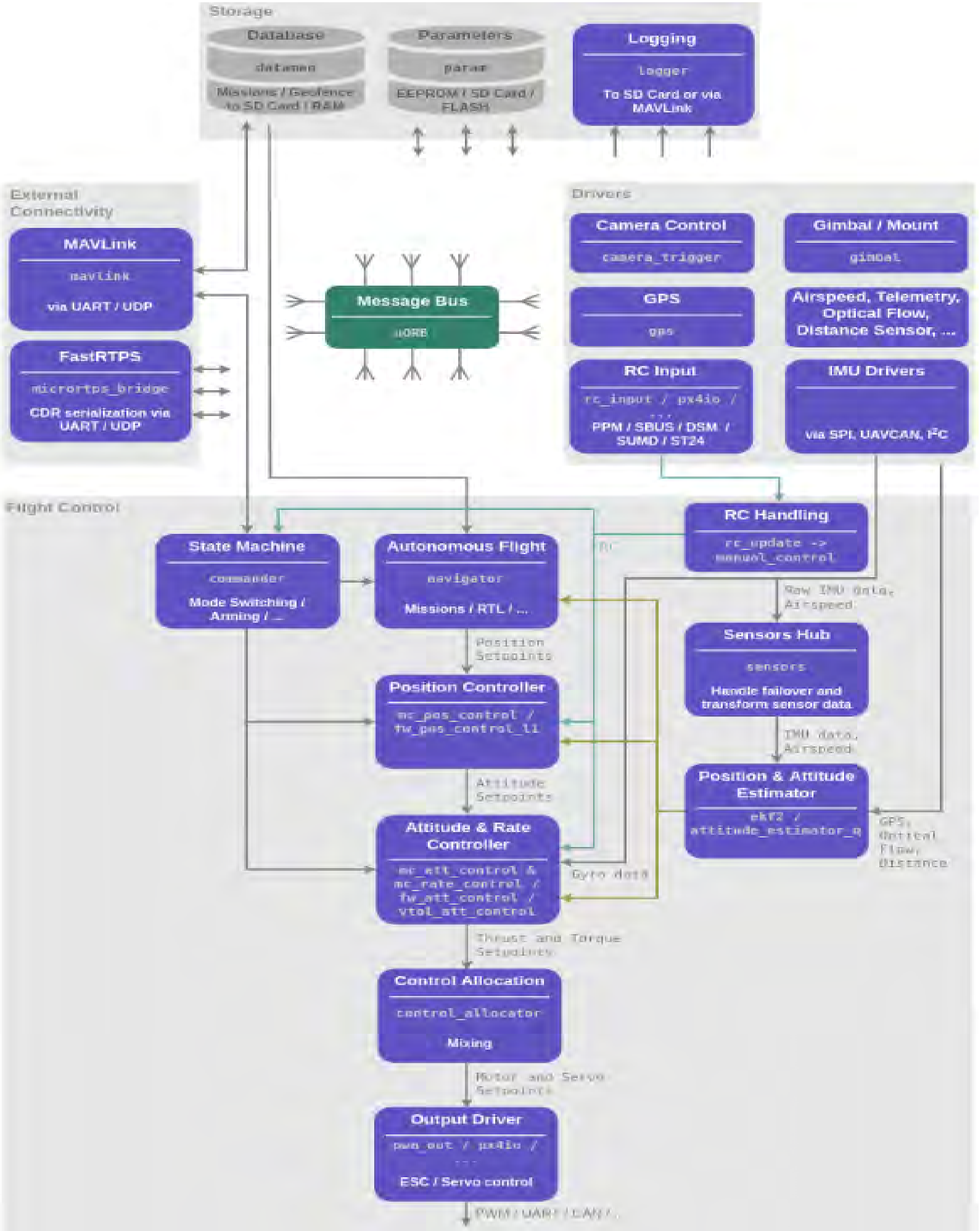


Figure 2.14: High-level software architecture
[23]

PX4 uses MAVLink protocol[24] for communication between drones, and between onboard drone components. Its key features are:

- Efficiency, it has just 14 bytes of overhead and does not require additional framing

- Reliability, it has been used since 2009 for communication between different vehicles, over challenging communication channels.
- Scalability, it allows up to 255 concurrent systems on the network, and it enables both offboard and onboard communications (e.g. between a GCS and drone, and between drone autopilot and MAVLink enabled drone camera).

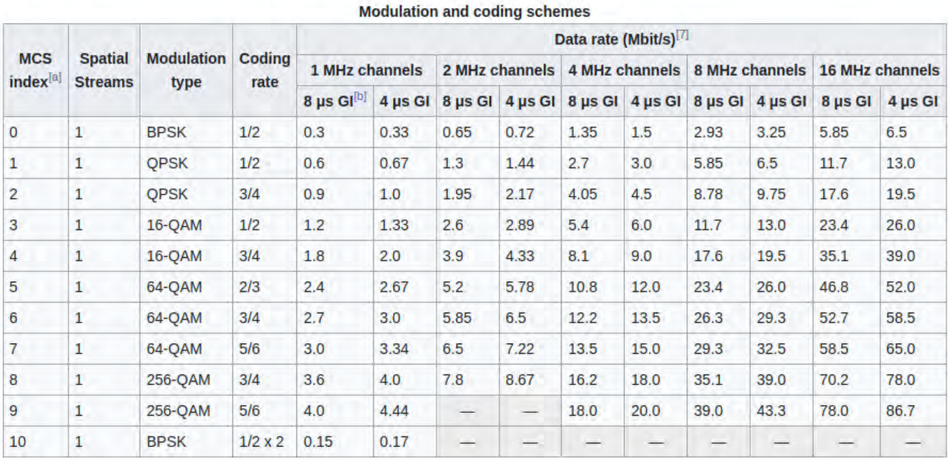
We also use the MAVLink protocol on the Raspberry Pi to transfer data through Wi-Fi to other drones, and into the autopilot through USB.

2.4.2 IEEE802.11ah software

The NRC7292 chipset2.2.2 comes with a driver developed for ARM64 architectures. This driver is fully configurable by the user, offering the chance to create many experiment scenarios. Some noticeable options are:

- *txpwr_max_default*, Board data max Tx power.
- *power_save*, (STA only) Power save mode.
- *prefer_bw*, (AP only) preferred bandwidth.
- *cqm_enable*, (STA only) Channel Quality Manager, keeps the connection regardless of Channel Quality.

Below, we present the Modulation and Coding Schemes (MCS) and the data rates based on channel bandwidth.



Modulation and coding schemes

MCS index[a]	Spatial Streams	Modulation type	Coding rate	Data rate (Mbit/s)[7]									
				1 MHz channels		2 MHz channels		4 MHz channels		8 MHz channels		16 MHz channels	
				8 µs GI[b]	4 µs GI	8 µs GI	4 µs GI	8 µs GI	4 µs GI	8 µs GI	4 µs GI	8 µs GI	4 µs GI
0	1	BPSK	1/2	0.3	0.33	0.65	0.72	1.35	1.5	2.93	3.25	5.85	6.5
1	1	QPSK	1/2	0.6	0.67	1.3	1.44	2.7	3.0	5.85	6.5	11.7	13.0
2	1	QPSK	3/4	0.9	1.0	1.95	2.17	4.05	4.5	8.78	9.75	17.6	19.5
3	1	16-QAM	1/2	1.2	1.33	2.6	2.89	5.4	6.0	11.7	13.0	23.4	26.0
4	1	16-QAM	3/4	1.8	2.0	3.9	4.33	8.1	9.0	17.6	19.5	35.1	39.0
5	1	64-QAM	2/3	2.4	2.67	5.2	5.78	10.8	12.0	23.4	26.0	46.8	52.0
6	1	64-QAM	3/4	2.7	3.0	5.85	6.5	12.2	13.5	26.3	29.3	52.7	58.5
7	1	64-QAM	5/6	3.0	3.34	6.5	7.22	13.5	15.0	29.3	32.5	58.5	65.0
8	1	256-QAM	3/4	3.6	4.0	7.8	8.67	16.2	18.0	35.1	39.0	70.2	78.0
9	1	256-QAM	5/6	4.0	4.44	—	—	18.0	20.0	39.0	43.3	78.0	86.7
10	1	BPSK	1/2 x 2	0.15	0.17	—	—	—	—	—	—	—	—

Figure 2.15: Modulation and Coding Schemes
[25]

In our implementation, we use the maximum available transmission power, no power saving mode, and the $2MHz$ channel. Also, we use an automatic way to choose Modulation type, the comparison of modulation types is not in the scope of this thesis.

3 Proposed Architecture

In this chapter, we describe our proposed software architecture that establishes the point to point communication. We begin by introducing the PX4 software stack, then we explain how we fetch information from the sensors, and how we command the drone. Below, we demonstrate an overview of the system's architecture.

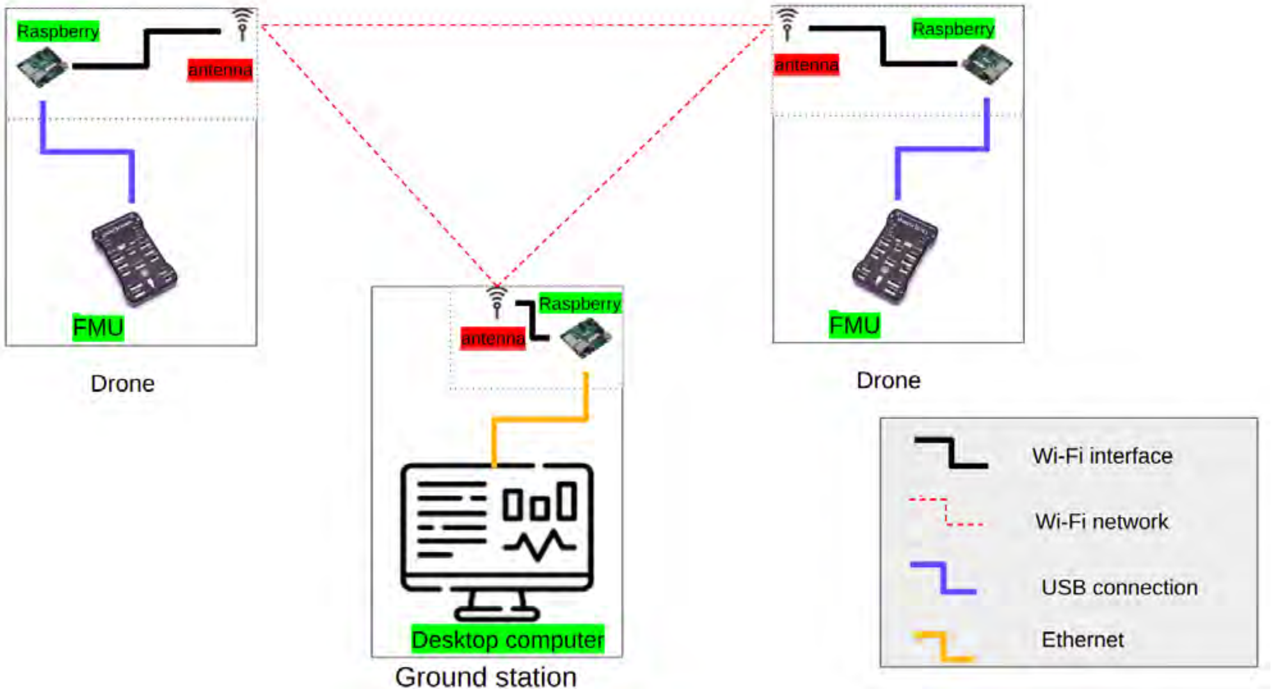


Figure 3.1: System architecture

3.1 Companion computer software

We have developed code in both RPIs (on the UAV and Ground station), responsible for commanding the FMU. This was necessary because the autonomous mission should start after two initialization processes listed below.

- Establishment of Wi-Fi connection between peers.
- iperf[26] command execution on both sides.

Due to the sensitivity of Wi-Fi modules, it is not certain that a connection will be established, also we have no access to the RPIs during an autonomous mission, thus every procedure should be automated.Also, our steering algorithm3.2 is integrated in the same code.

Algorithm 1 Drone commander algorithm

```
Require: $thread \leftarrow$ while $true$ do $receiveData()$ end while

Require: $state \leftarrow subscribe(uavState)$
  $initializeWiFi()$
  $pingPeer()$
  $armUAV()$
  $iperfStart()$
  $missionStart()$
  while $state\,!= STATE\_LANDED$ do
      $state.update()$
  end while
  $disarmUAV()$
  $iperfStop()$
```

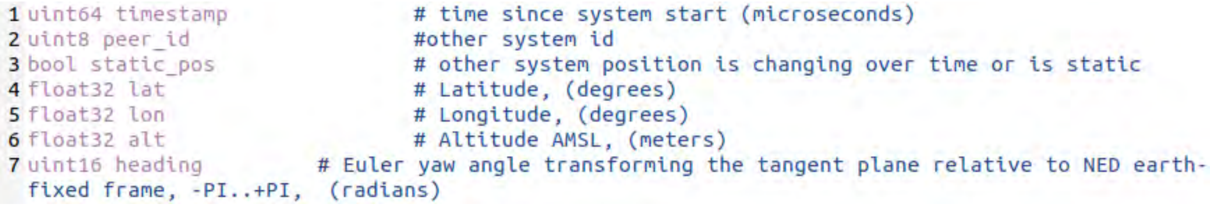
The *receiveData*() function simply listens to the serial port and handles incoming messages (GPS, state, velocity etc.) and stores essential information for further use. Next, *initializeWiFi*() function checks if the Wi-Fi interface is up and connected. Then we make sure the Wi-Fi link is working by sending a ping to other nodes using *pingPeer*() function. At last, we begin an autonomous mission by sending an appropriate command to the UAV using *missionStart*() function.

3.2 Steering algorithm

First, we define the essential data we have to transfer between nodes. Message types are defined in text files inside the PX4 stack, our custom message called peer position

(PEER_POSITION.msg) 3.2 has information about another node in the system, more specific it contains:

- Time since the system start.
- The ID of the system.
- Whether the system is moving or not.
- the global position, measured in GPS coordinates (latitude, longitude). This is not the raw GPS measurement, rather than a fusion of more sources of information, e.g. control inputs of the vehicle in a Kalman-filter implementation.
- The altitude of the system, above the mean sea level.
- heading of the system measured in radians.



```
uint64 timestamp               # time since system start (microseconds)
uint8 peer_id                  #other system id
bool static_pos                # other system position is changing over time or is static
float32 lat                    # Latitude, (degrees)
float32 lon                    # Longitude, (degrees)
float32 alt                    # Altitude AMSL, (meters)
uint16 heading         # Euler yaw angle transforming the tangent plane relative to NED earth-
fixed frame, -PI..+PI,  (radians)
```

Figure 3.2: Peer position message definition

This message is encapsulated into a MAVLink packet, and sent to other node(s), at a constant rate of $1Hz$. This is the only information sent through our Wi-Fi link. Each system is responsible for creating and transferring this message to others (this information is important for antenna steering). The next step is to receive the peer position message through the Raspberry Pi and forward it to the FMU via the USB connection. One might ask why do we need the Raspberry Pi as a middle device, as of now there is no way to integrate an IEEE802.11ah module directly to the FMU. The last step of this procedure is to decode the MAVLink (peer position) message inside the PX4 stack, and publish it using uORB. We developed an application in the PX4 stack, that is responsible for handling the geolocation information of peer nodes. The algorithm it implements is the following:

Algorithm 2 Steering algorithm

```
Require: $myGPS \leftarrow subscribe(myGPStopic)$
Require: $peerGPS \leftarrow subscribe(peerGPStopic)$
  while true do
      $myGPS.update()$
      $peerGPS.update()$
      $calculateDistance()$
      $calculateElevationAngle()$
      $ServoOutput.update(mapAngleToPWM())$
  end while
```

The distance between a UAV and the Ground station is calculated using the Haversine formula:

$$a = sin^2(\frac{\varphi_2 - \varphi_1}{2}) + cos(\varphi_1) * cos(\varphi_2) * sin^2(\frac{\lambda_2 - \lambda_1}{2})$$
$$c = 2 * atan2(\sqrt{a}, \sqrt{1-a})$$
$$d = R * c$$

where φ_1, φ_2 are the latitudes of the UAV and Ground station, λ_1, λ_2 are the longitudes of the UAV and Ground station and R is the earth's radius. The elevation angle is calculated, using the following formula:

$$\theta = arctan(\frac{\alpha_1 - \alpha_2}{d})$$

where α_1, α_2 are the relative altitudes (with reference to the mission starting point) of the UAV and Ground station, and d is the distance calculated above.

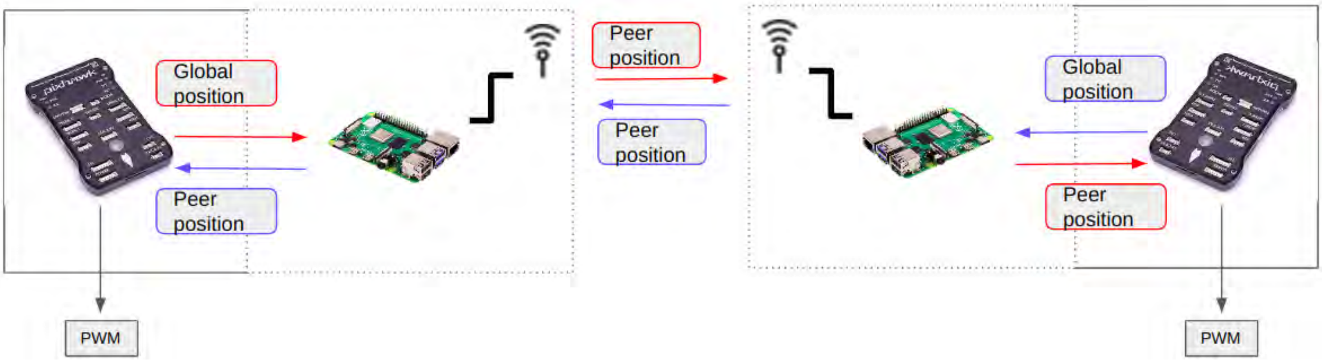


Figure 3.3: Message transfer between UAV and GS

The same application/algorithm was developed for the Raspberry Pi device. Our reasoning was to not overload this specific FMU, because it does not offer AUX outputs for servo control, and due to the sensitivity of such devices, we did not want to risk damaging the UAV during a real experiment.

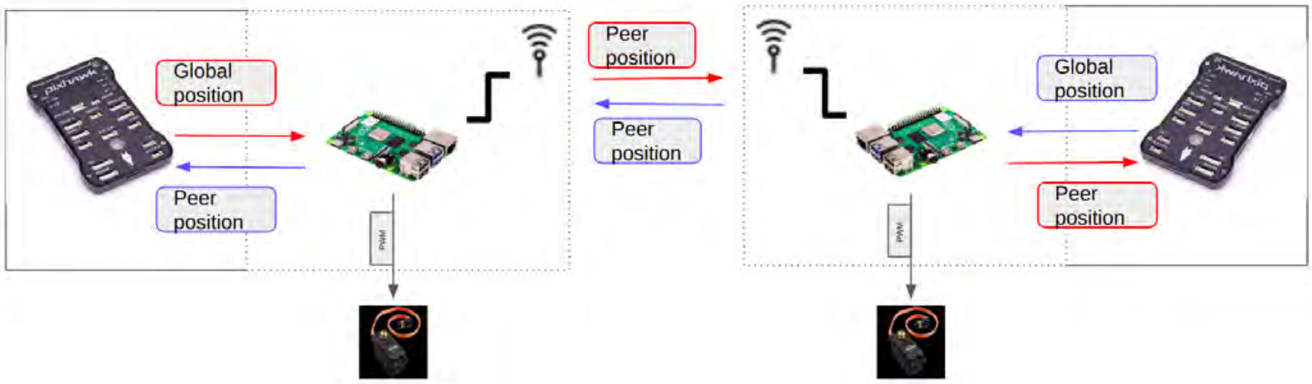


Figure 3.4: Message transfer between UAV and GS, servo mounted on RPI

4 Evaluation

This chapter, presents the performance of our system. First, we evaluate omnidirectional and directional antennas in indoor and outdoor environments, with and without mobility on the nodes. Then, we estimate the energy consumption of the components we integrated. Finally, we compare a directional antenna with an omnidirectional antenna on the drone, the ground station has an omnidirectional antenna in every experiment.

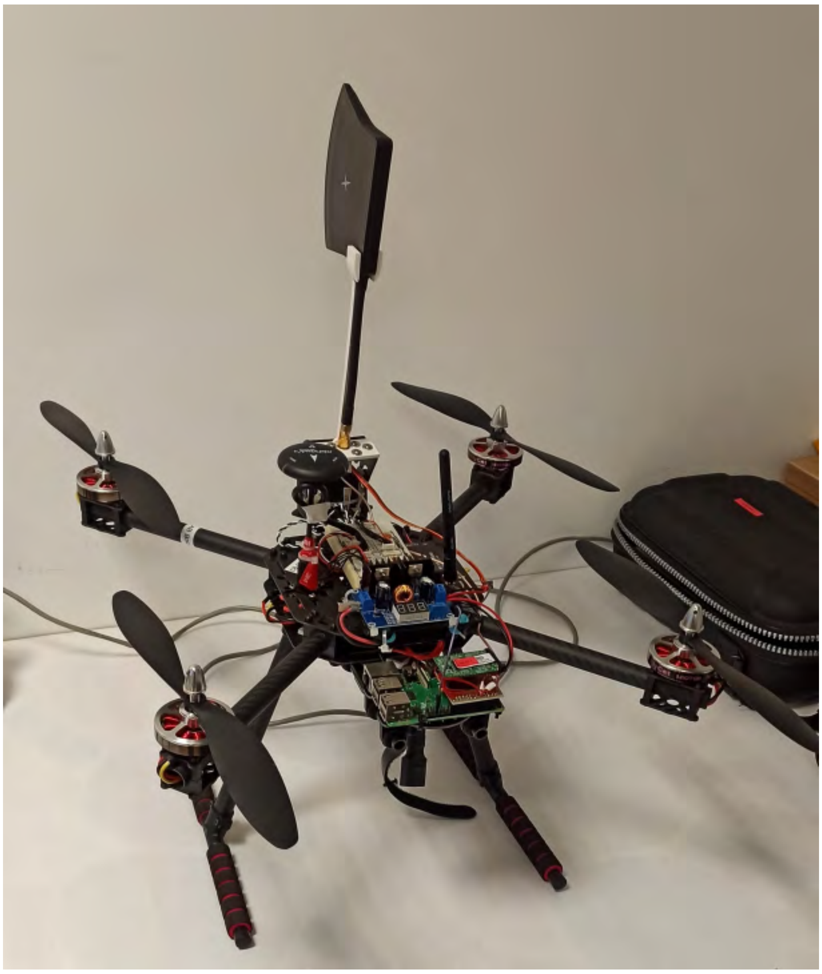


Figure 4.1: Evaluation setup

4.1 Tools & Techniques

We used several tools and techniques to evaluate our architecture. The link throughput is measured using **iperf** tool, sending UDP packets, the station is always on the UAV side and the server on the Ground station side. Also, we measured the link quality using **iwconfig**[27] command every 3 seconds. We developed extensive logging mechanisms on the UAV, to collect every significant information during a mission, some information we collect are:

- GPS location
- (relative) altitude
- velocity
- angle of elevation
- iperf output
- servo angle

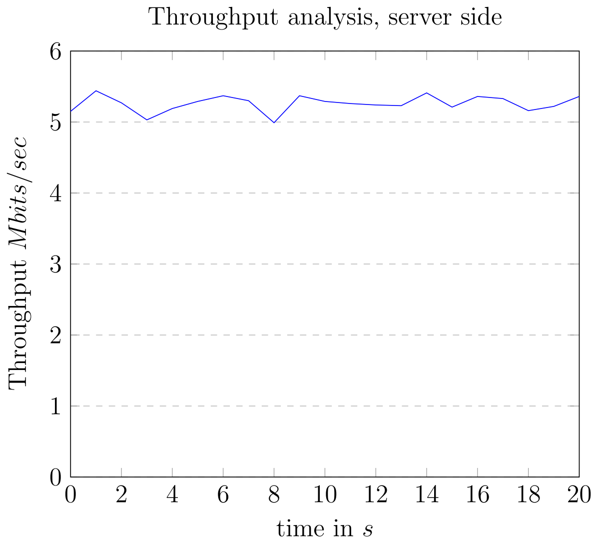
The servo mechanism was calibrated by applying different PWM values, and measuring the angle using a protractor. We managed to map the angles to PWM signal values with a drift of $\pm 5^o/$.

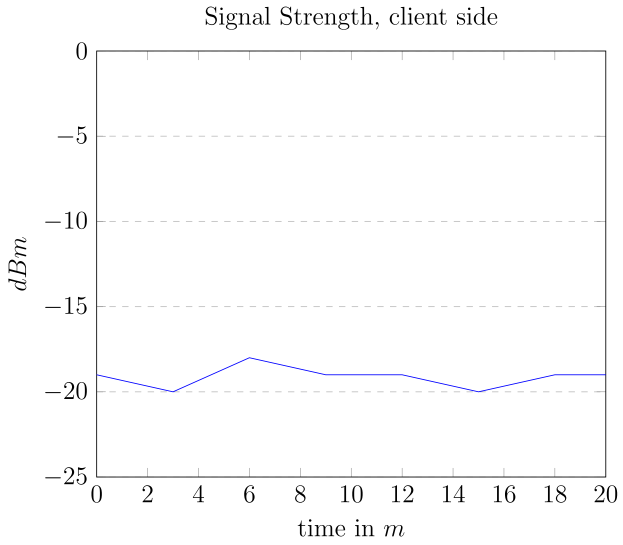
4.2 Omnidirectional antennas

Each scenario was repeated three times and the results we produced are the average measurements of those scenarios.

4.2.1 Office experiments

First, we evaluate the throughput achieved between two IEEE802.11ah modules inside the office. The maximum distance is $\sim 1m$.



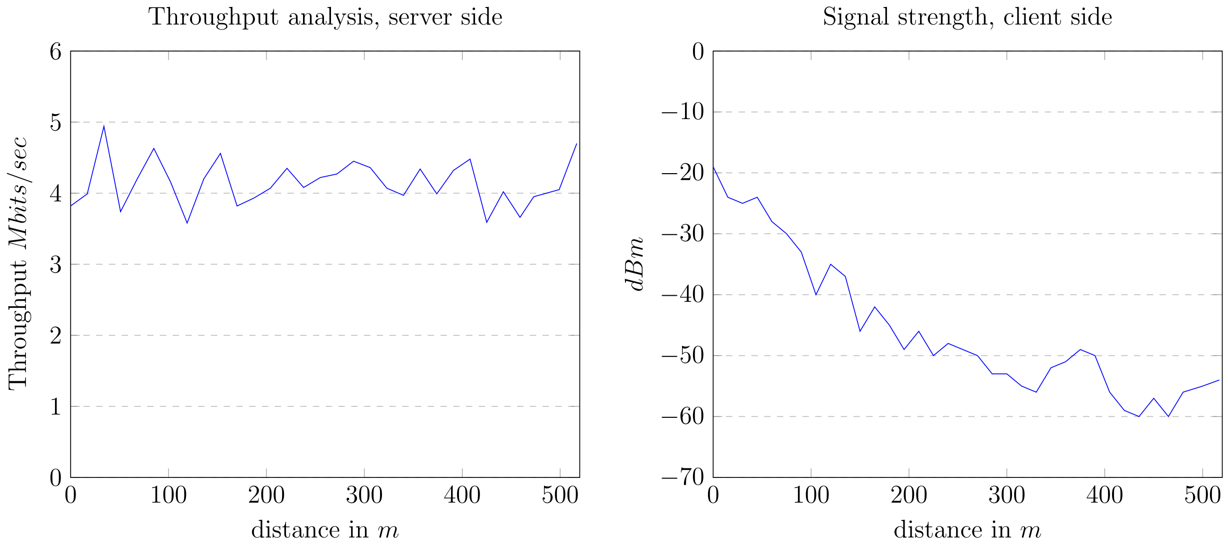


The average throughput achieved was $5.26 Mbits/sec$.

4.2.2 Outdoor experiments

The setup of this experiment consists of a drone with $3m$ angle of elevation and a very low mobility profile of $\sim 1m/s$ (a pedestrian was moving with the drone) and a ground station. In this experiment, there is LOS between transmitter, and receiver. The maximum distance between two nodes was $517m$.

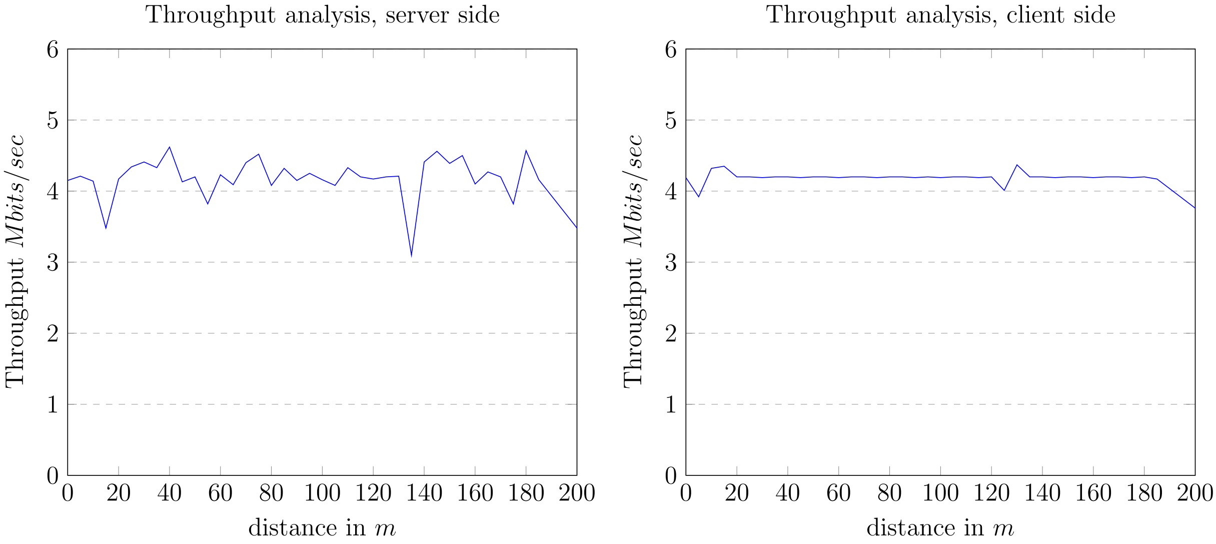
Figure 4.2: Environment of static experiments (Sesklo, Volos)



The average throughput achieved was $4.13 Mbits/sec$.

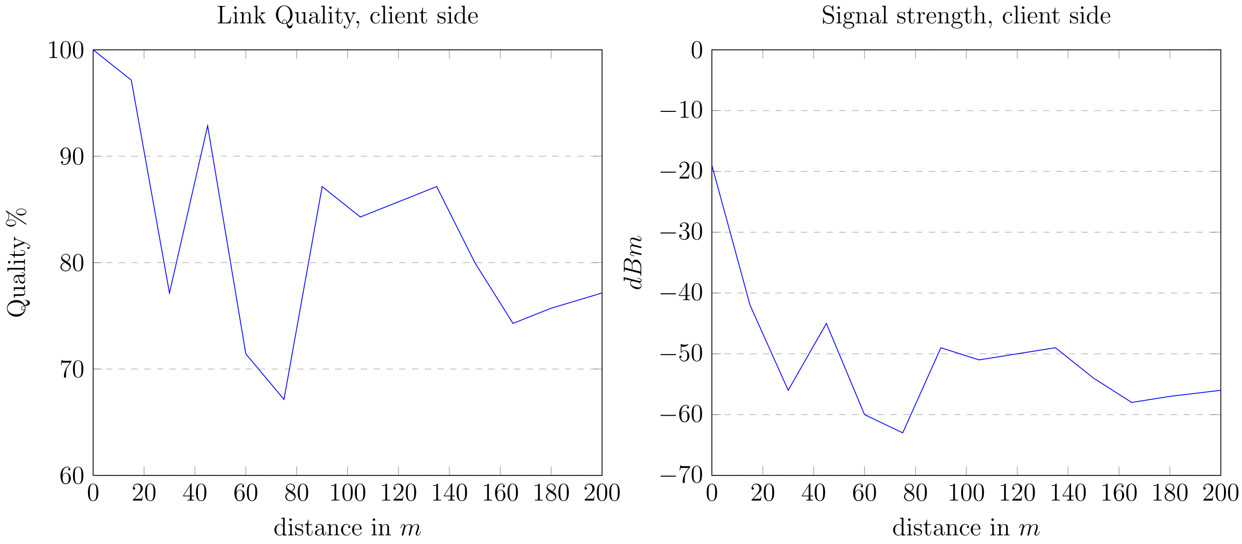
Our next experiment consisted of a ground station, and a moving UAV, the velocity of the UAV was $5m/s$, and its altitude $10m$, the maximum distance achieved was $200m$.

Figure 4.3: Environment of dynamic experiment (Sesklo, Volos)



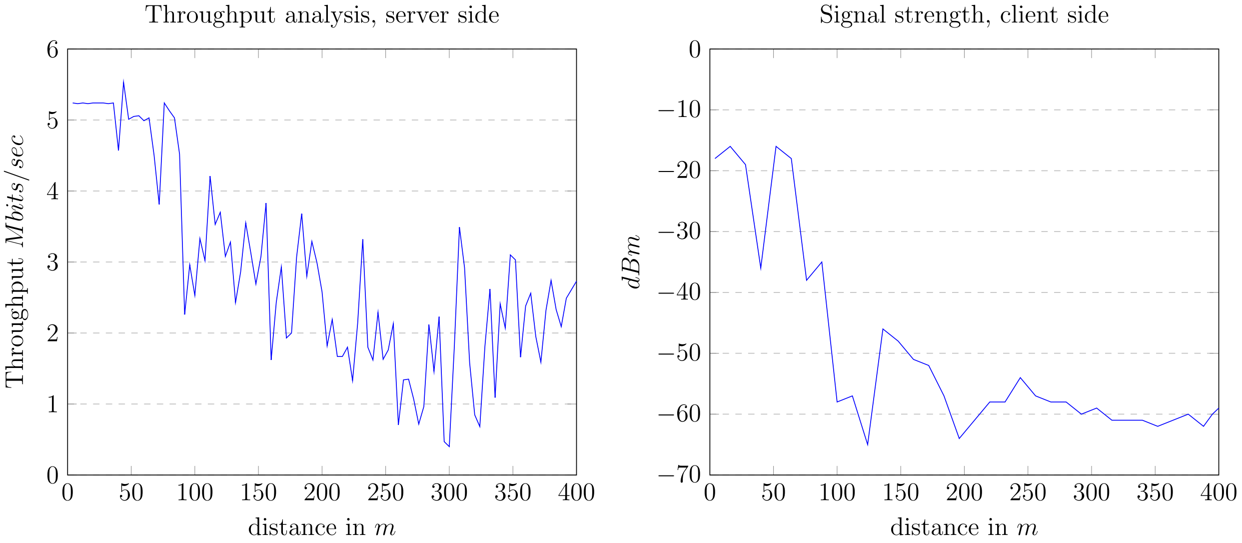
The average throughput, achieved on the server side, was $3.82 Mbits/sec$. The average throughput, measured on the client side, was $3.92 Mbits/sec$. We observe less jitter on the client side (transmitter), which is not representative of the real measurements. In the rest of the experiments, we are going to present only the server (receiver) side.

Below, we present the signal quality between the UAV and GS.



In the next experiment, the UAV was moving with a velocity of $5m/s$ and its altitude was $15m$, the maximum distance achieved was $400m$.

Figure 4.4: Environment of dynamic experiment (Lake Karla)



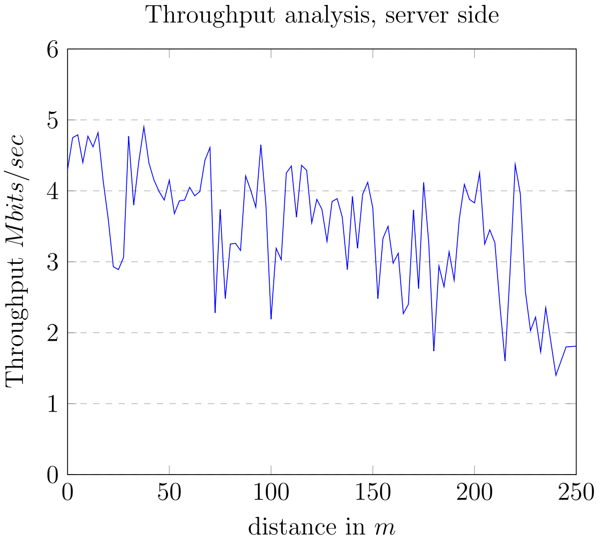
The average throughput achieved, was $2.89 Mbits/sec$. We also conducted two experiments in an urban environment where there are obstructions and no LOS between the two devices at all times. The first setup consists of the UAV at an altitude of $10m$ relative to the ground station. We moved the ground station instead of the UAV, due to environment constraints, the speed was $\sim 1.5m/s$ and maximum distance achieved was $240m$.

Figure 4.5: Environment of urban experiment (Volos)


Throughput analysis, server side


Signal strength, client side

The average throughput, achieved on the server side, was $2.75Mbits/sec$. In Urban environments we observe a degradation of the signal, due to interference[28] with other wireless technologies, reflections and obstructions. In the second setup, the altitude of our UAV is $20m$ relative to the ground station in the same environment. The maximum distance achieved was $240m$.


Throughput analysis, server side


Signal strength, client side

The average throughput achieved on the server side was $2.47Mbits/sec$, showing that altitude difference can degrade the signal quality between two nodes. Nevertheless, we can safely state that the results are promising for Wi-Fi technology in complex environments like a city. Table4.1 presents and compares the throughput achieved using our proposed system and the system developed by 1.2. We observe that for short ranges IEEE802.11ah is not the optimum solution for high capacity communication, but it outperforms conventional Wi-Fi technologies in long ranges.

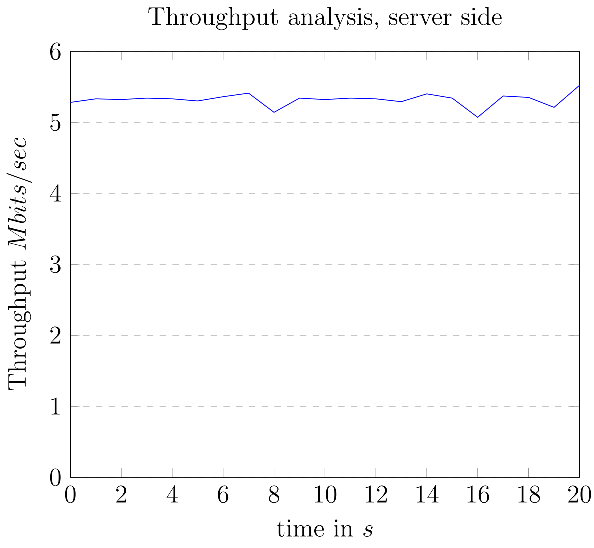
	Drone at 10m high	
Distance(m)	802.11g/n	802.11ah
15	3.5 Mpbs	5.24 Mbps
18	3.7 Mbps	5.24 Mbps
21	1.14 Mbps	5.24 Mbps
24	4.89 Mbps	5.23 Mbps
27	3.3 Mbps	5.23 Mbps
30	<1 Mbps	5.22 Mbps
33	<1 Mbps	5.22 Mbps
36	<1 Mbps	5.19 Mbps
39	3.79 Mbps	5.19 Mbps
42	4.22 Mbps	5.0 Mbps
45	<1 Mbps	5.0 Mbps
50	–	4.2 Mbps
100	–	4.16 Mbps
150	–	4.39 Mbps
200	–	3.48 Mbps
250	–	3.32 Mbps
300	–	2.92 Mbps
350	–	3.03 Mbps
400	–	2.73 Mbps

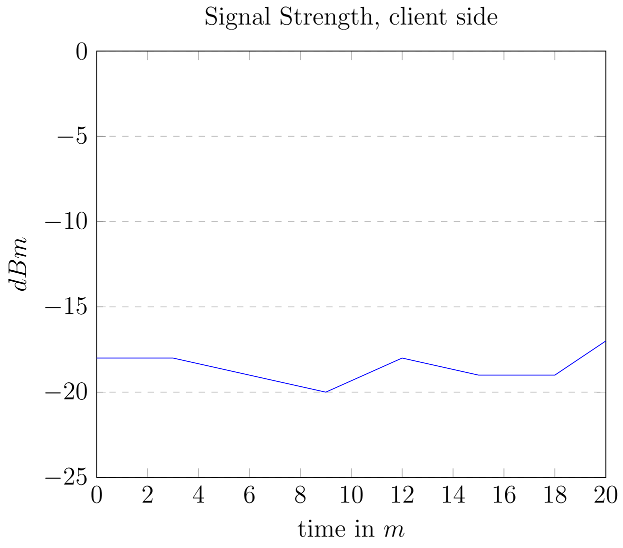
Table 4.1: Comparison of 802.11 Operational Bandwidth

4.3 Directional antenna

4.3.1 Office experiments

First, we evaluate the throughput achieved between two IEEE802.11ah modules inside the office. The maximum distance is $\sim 1m$. Our setup consists of a directional and omnidirectional antenna mounted on the modules.





The average throughput achieved was $5.31 Mbits/sec$.

4.3.2 Outdoor experiments

The first experiment consisted of a ground station, and a moving UAV with a fixed directional antenna, the velocity of the UAV was $\sim 5m/s$, and its altitude $10m$, the maximum distance achieved was $90m$.

Figure 4.6: Environment of directional antenna experiment (Sesklo, Volos)


Throughput analysis, server side


Signal strength, client side

Although, we created the steering mechanism, the evaluation of the system was not possible, because the infrastructure (UAV) was unstable during the antenna's change of direction, thus making the system prone to failure/wrecking. Also, the concept of the directional antenna was meant to be used for distances above $1km$, between two nodes, during the experimentation phase we concluded that it was difficult to send a UAV in such distances, due to battery limitations and loss of connection with the joystick (RC), hence increasing the risk of permanently losing control of the system and causing damage. We are trying to find solutions based on different directional antennas, in order to make the UAV safe during a mission.

4.4 Energy footprint

We measured quantitatively the power consumption of our components, mainly the Raspberry Pi, AHMB7292S device, and servo motor. All the experiments were conducted for $\sim 20min$ in our office bench. The duration was decided based on the estimated flight time of the UAV ($15min$), thus a longer duration would not provide more insight. We used FLUKE 179 multimeter[29] for every measurement.

4.4.1 Servo motor

The first experiment was conducted by moving the motor constantly without any payload. The motor was connected to a Li-Po battery (4 cells, 6750mAh).

For the second experiment, we mounted the directional antenna on the motor, then we commanded the motor to constantly move. The motor was connected to the same Li-Po battery as above.

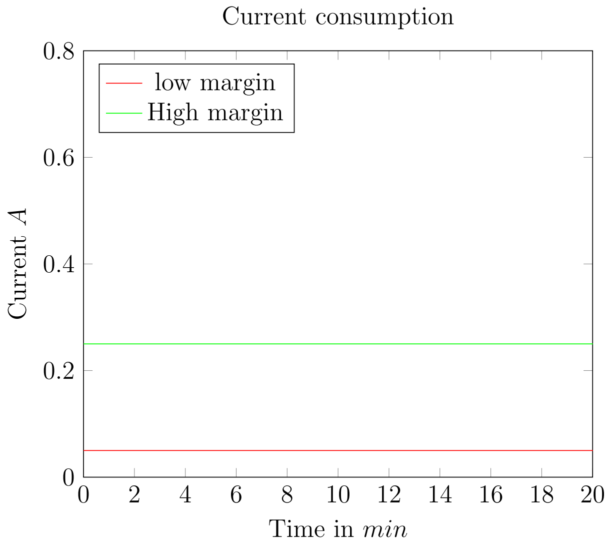


Figure 4.7: First experiment

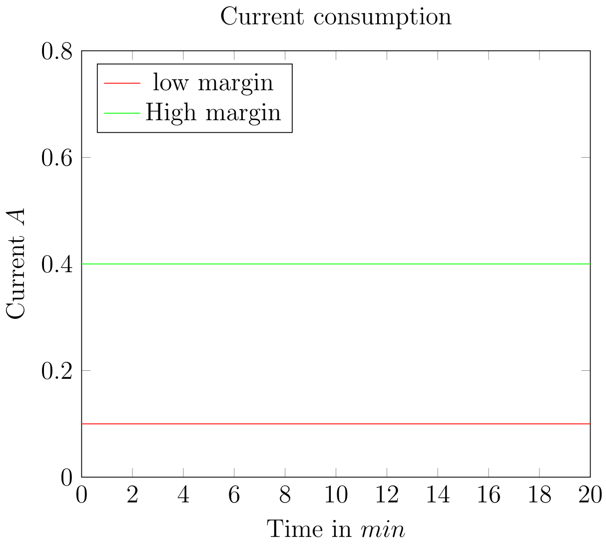


Figure 4.8: Second experiment

	First experiment	Second experiment
Voltage(V) $t = 0min$	16.62	16.81
Voltage (V) $t = 20min$	16.59	16.77
Current Draw low margin(A)	0.05	0.25
Current Draw high margin(A)	0.1	0.4

Table 4.2: Voltage drop and Current consumption

4.4.2 IEEE802.11ah device

In this experiment, we measure the power consumption of the AHMB7292S and Raspberry Pi device along with the FMU and all its components (GPS, telemetry, RC receiver). We measure the consumption while running our software (exchange GPS messages, using the Wi-Fi link) and producing heavy traffic through the Wi-Fi link using iperf.

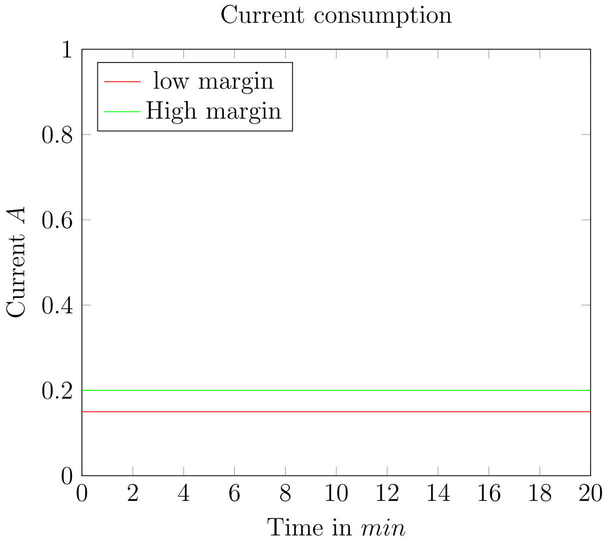


Figure 4.9: FMU & components

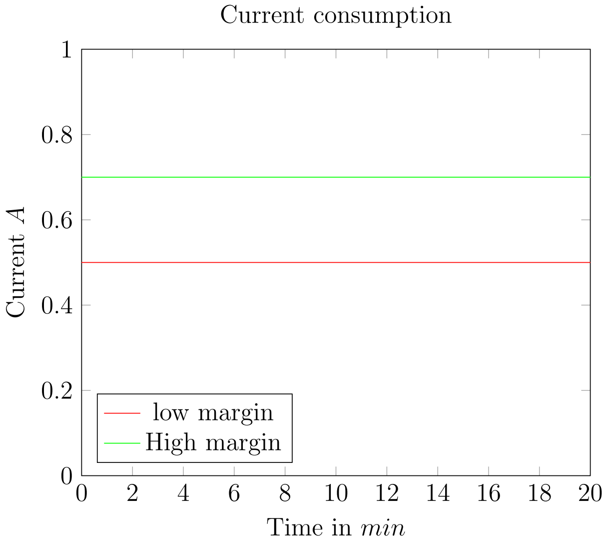


Figure 4.10: RPI & IEEE802.11ah module

Experiment	
Voltage(V) $t = 0min$	16.8
Voltage(V) $t = 20min$	16.7
FMU & components current draw low margin(A)	0.15
FMU & components current draw high margin(A)	0.2
RPI & IEEE802.11ah module current draw low margin(A)	0.4
RPI & IEEE802.11ah module current draw high margin(A)	0.5

Table 4.3: Voltage drop and Current consumption

5 Conclusion

Long range wireless networks can be achieved at a low energy cost and perform adequately, especially in non-urban areas where there is LOS between nodes. In this thesis, we have described and evaluated a Wi-Fi protocol for long range communications. We presented every hardware component used to create a functioning system, also we described the software we developed to run and evaluate the system in our experiments. We presented experiments using a real UAV in different realistic altitudes and areas. We provided information about the energy demands of such networks and their respected devices. Our system achieved $2.89 Mbits/sec$ at a distance of $400m$ and an altitude of $15m$ in a rural environment, and $2.75 Mbits/sec$ at a distance of $240m$ and an altitude of $10m$ in an urban environment. Also, we discussed and developed a steering mechanism for directional antennas, in order to improve signal quality in long range networks. Currently, we are trying to make the steering mechanism more robust to avoid any potential hazards during experimentation and finally, evaluate the system in longer distances.

References

[1] C. Kanellakis and G. Nikolakopoulos. “Survey on Computer Vision for UAVs: Current Developments and Trends.” In: *J Intell Robot Syst* 87.1 (2017), pp. 141–168. DOI: `https://doi.org/10.1007/s10846-017-0483-z`.

[2] E. Semsch et al. “Autonomous UAV Surveillance in Complex Urban Environments”. In: *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology.* Vol. 2. 2009, pp. 82–85. DOI: `10.1109/WI-IAT.2009.132`.

[3] T. Tomic et al. “Toward a Fully Autonomous UAV: Research Platform for Indoor and Outdoor Urban Search and Rescue”. In: *IEEE Robotics & Automation Magazine* 19.3 (2012), pp. 46–56. DOI: `10.1109/MRA.2012.2206473`.

[4] M. Campion, P. Ranganathan, and S. Faruque. “UAV swarm communication and control architectures: a review”. In: *Journal of Unmanned Vehicle Systems* 7.2 (2019), pp. 93–106. DOI: `10.1139/juvs-2018-0009`.

[5] *Zigbee.* URL: `https://en.wikipedia.org/wiki/Zigbee`.

[6] *LoRa.* URL: `https://en.wikipedia.org/wiki/LoRa`.

[7] *LTE-M.* URL: `https://docs.aws.amazon.com/whitepapers/latest/implementing-lpwan-solutions-with-aws/lte-m.html`.

[8] M. Alam et al. “IEEE 802.11ah-Enabled Internet of Drone Architecture”. In: *IEEE Internet of Things Magazine* 5.1 (2022), pp. 174–178. DOI: `10.1109/IOTM.005.2100099`.

[9] A. Fakhreddine et al. “Experiments on Drone-to-Drone Communication with Wi-Fi, LTE-A, and 5G”. In: *2022 IEEE Globecom Workshops (GC Wkshps)*. 2022, pp. 904–909. DOI: `10.1109/GCWkshps56602.2022.10008743`.

[10] A. Guillen-Perez et al. “WiFi networks on drones”. In: *2016 ITU Kaleidoscope: ICTs for a Sustainable World (ITU WT)*. 2016, pp. 1–8. DOI: `10.1109/ITU-WT.2016.7805730`.

[11] B. Y. Chang et al. “Design and implementation of UAV-enabled flying access points”. In: *2018 27th Wireless and Optical Communication Conference (WOCC)*. 2018, pp. 1–2. DOI: `10.1109/WOCC.2018.8372714`.

[12] “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 2: Sub 1 GHz License Exempt Operation”. In: *IEEE Std 802.11ah-2016 (Amendment to IEEE Std 802.11-2016, as amended by IEEE Std 802.11ai-2016)* (2017), pp. 1–594. DOI: `10.1109/IEEESTD.2017.7920364`.

[13] L. F. Del Carpio et al. “Comparison of 802.11ah and BLE for a home automation use case”. In: *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. 2016, pp. 1–6. DOI: `10.1109/PIMRC.2016.7794554`.

[14] T. Adame et al. “IEEE 802.11AH: the WiFi approach for M2M communications”. In: *IEEE Wireless Communications* 21.6 (2014), pp. 144–152. DOI: `10.1109/MWC.2014.7000982`.

[15] *KIT-HGDRONEK66*. URL: `https://www.nxp.com/design/designs/nxp-hovergames-drone-kit-including-rddrone-fmuk66-and-peripherals:KIT-HGDRONEK66`.

[16] *NXP RDDRONE-FMUK66 FMU*. URL: `https://docs.px4.io/v1.11/en/flight_controller/nxp_rddrone_fmuk66.html`.

[17] *Wi-Fi 802.11ah module.* URL: `https://www.data-alliance.net/ieee-802-11ah-sub-1-ghz-module-with-mikrobus-form-factor/`.

[18] *Wikipedia. Omnidirectional antenna.* URL: `https://en.wikipedia.org/wiki/Omnidirectional_antenna`.

[19] *Wikipedia. Directional antenna.* URL: `https://en.wikipedia.org/wiki/Directional_antenna`.

[20] *Taoglas TS.89.4113 Sighunter.* URL: `https://www.taoglas.com/product/ts-89-4113-sighunter-868-915mhz-directional-terminal-mount-antenna-with-flexible-gooseneck/`.

[21] *QGroundControl.* URL: `http://qgroundcontrol.com/`.

[22] *uORB Messaging.* URL: `https://docs.px4.io/main/en/middleware/uorb.html`.

[23] *PX4 building blocks.* URL: `https://docs.px4.io/main/en/concept/architecture.html#high-level-software-architecture`.

[24] *MAVLink.* URL: `https://mavlink.io/en/`.

[25] *Wikipedia. IEE802.11ah.* URL: `https://en.wikipedia.org/wiki/IEEE_802.11ah`.

[26] *iperf.* URL: `https://iperf.fr/`.

[27] *iwconfig.* URL: `https://linux.die.net/man/8/iwconfig`.

[28] A. Sheth et al. “Packet Loss Characterization in WiFi-Based Long Distance Networks”. In: *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications.* 2007, pp. 312–320. DOI: `10.1109/INFCOM.2007.44`.

[29] *FLUKE multimeter.* URL: `https://www.fluke.com/en/product/electrical-testing/digital-multimeters/fluke-179`.