UNIVERSITÀ DI PARMA

UNIVERSITÀ DEGLI STUDI DI PARMA

Dottorato di Ricerca in Tecnologie dell'Informazione

XXXVI Ciclo

UAV-to-X Connectivity and Localization Systems: Design, Implementation and Performance Evaluation

Coordinatore:

Chiar.mo Prof. Marco Locatelli

Tutor:

Chiar.mo Prof. Gianluigi Ferrari

Chiar.mo Prof. Luca Davoli

Dottorando: *Emanuele Pagliari*

Anni 2020 - 2023

Nel fare ricerca, si tratta di vedere ciò che tutti hanno visto
e di pensare ciò che nessuno ha pensato.

Summary

List of Figures

List of Tables

Acknowledgements

Intraprendere il percorso del dottorato di ricerca è stata una decisione che ha richiesto coraggio e determinazione, poiché ha comportato notevoli sacrifici e un impegno incessante nel perseguire la conoscenza e l'innovazione nel campo della mia ricerca. Questa esperienza è stata una vera e propria avventura, costellata di sfide, momenti stressanti e complessità inimmaginabili. Durante questo viaggio, ho avuto l'opportunità di incontrare molte persone straordinarie, arricchire il mio bagaglio culturale e scoprire nuove realtà e culture. È stato un percorso impegnativo che mi ha permesso di chiarire le mie aspirazioni professionali.

Tutto questo, sia nei momenti positivi che in quelli difficili, è stato possibile grazie al sostegno delle persone a me care. Desidero quindi esprimere la mia più profonda gratitudine alla mia famiglia, ai miei amici, ai miei colleghi e a tutti coloro che ho incontrato durante questo percorso. Il vostro supporto e incoraggiamento sono stati fondamentali per raggiungere questo traguardo.

In particolare, vorrei rivolgere un sentito ringraziamento al mio relatore, il Professore Gianluigi Ferrari, per la fiducia che ha riposto in me sin dall'inizio e per avermi concesso la libertà di esplorare e condurre ricerche innovative durante il mio dottorato. Un ringraziamento speciale va anche al mio correlatore, Luca Davoli, per la pazienza, l'assistenza e il sostegno che mi ha offerto nel corso di questi intensi anni di studio e ricerca. La sua efficienza, disponibilità ma anche simpatia e ironia, sono stati cruciali nel rendere possibile il mio percorso accademico.

Ringrazio l'Unione Europea per aver finanziato le mie attività di dottorato tramite il progetto europeo ADACORSA, che mi ha offerto l'opportunità di lavorare su tematiche stimolanti e sviluppare nuove idee e soluzioni nel campo dei droni, una delle mie più grandi passioni.

Desidero inoltre esprimere la mia riconoscenza al gruppo di ricerca dell'Università di Perugia, oltre al team del Professore George Nikolakopoulos dell'Università di Luleå in Svezia, che mi ha ospitato nella sua struttura consentendomi di esplorare nuove tecnologie e soluzioni in un ambiente completamente diverso dal mio contesto di origine.

Ringrazio ancora una volta, ma non basterebbe mai, la mia famgila, per il loro costante sostegno e comprensione, anche nei momenti più impegnativi.

Un ringraziamento particolare va anche ai miei amati gatti, che con la loro presenza felina ed affetto hanno reso più piacevoli e meno solitari questi anni di studio.

Infine, ringrazio i miei amici, che hanno compreso le mie scelte ed impegni e si sono dimostrati sempre disponibili e comprensivi. So di non aver sempre trascorso il tempo che avrei voluto con voi, ma la vostra amicizia è un tesoro inestimabile.

Ultimo, ma non meno importante, dedico un ringraziamento a me stesso. Per aver perseverato, nonostante le sfide e i momenti difficili. Per aver sempre dato il massimo, pensato oltre i limiti e affrontato con determinazione e passione questo straordinario percorso, fino alla fine.

List of Acronyms

3GPP	3rd Generation Partnership Project
ABP	Activation-By-Personalization
ACK	Acknowledgment
AGL	Above Ground Level
AoA	Angle of Arrival
AODV	Ad-hoc On-Demand Distance Vector Routing Protocol
AI	Artificial Intelligence
AP	Access Point
AS	Application Server
BLE	Bluetooth Low Energy
BTS	Base Transceiver Station
BVLOS	Beyond Visual Line-of-Sight
BW	Bandwidth
B.A.T.M.A.N.	Better Approach To Mobile Ad-Hoc Network
C2	Command & Control
CA	Carrier Aggregation
CDF	Cumulative Distribution Function
CBTS	Connected Base Transceiver Station
CF	Carrier Frequency
CNN	Convolutional Neural Network
CI	Confidence Interval
COTS	Commercial-Off-The-Shelf
CQI	Channel Quality Indicator
CR	Coding Rate
CSI	Channel State Information
CSS	Chirp Spread Spectrum
CV	Computer Vision

DL	Downlink
DHCPCD	Dynamic Host Configuration Protocol Client Daemon
DSS	Decision Support System
DTN	Delay Tolerant Networking
DTU	Data Transfer Unit
EASA	European Union Aviation Safety Agency
ECEF	Earth-centered, Earth-fixed
EKF	Extended Kalman Filter
eNB	evolved NodeBs
EMA	Exponential Moving Average
ETSI	European Telecommunications Standards Institute
FAA	Federal Aviation Administration
FANET	Flying Ad-Hoc Network
FaaS	Farm-as-a-Service
FC	Flight Controller
FEC	Forward Error Correction
FIFO	First-In-First-Out
FOV	Field of View
FPV	First Point of View
FSK	Frequency-Shift Keying
FSPL	Free-Space Path Loss
FTM	Fine Time Measurement
GNSS	Global Navigation Satellite System
GPS	Global Position System
GW	Gateway
HDOP	Horizontal Dilution Of Precision
HQ	High Quality
HW	Hardware
HWMP	Hybrid Wireless Mesh Protocol
ICP	Iterative Closest Point
IMU	Inertial Measurement Unit
IoT	Internet of Things
IIoT	Industrial Internet of Things
IP	Internet Protocol
IQ	Chirp Polarity
ISM	Industrial, Scientific and Medical

ISP	Internet Service Provider
KF	Kalman Filter
LAN	Local Area Network
LBT	Listen-Before-Talk
LiDAR	Light Detection and Ranging
LoRa	Long Range
LoRaWAN	Long Range Wide Area Network
LOS	Line-of-Sight
LPWAN	Low Power Wide Area Network
LS	Least Squares
LTE	Long Term Evolution
LTU	Luleå University of Technology
MAC	Medium Access Control
MAP	Mesh Access Point
MBSS	Mesh Basic Service Set
MIG	Multi Interface Gateway
MIMO	Multiple Input Multiple Output
ML	Machine Learning
MMT	Mission Management Tool
MPM	Mesh Peer (Link) Management
MPP	Mesh Portal Point
MQTT	Message Queue Telemetry Transport
MTOM	Maximum Take Off Mass
NBTS	Nearest Base Transceiver Station
NB-IoT	NarrowBand-Internet of Things
NLOS	Non-LOS
NS	Network Server
OBTS	Omni-directional Base Transceiver Station
OLSR	Optimized Link State Routing Protocol
OTAA	Over-The-Air-Activation
PDU	Packet Data Unit
PCB	Printed Circuit Board
PHY	Physical Layer
PKT	Packet
PMK	Pairwise Master Keys
POE	Power Over Ethernet

PRB	Physical Resource Block
RADAR	RAdio Detection and Ranging
RAM	Random Access Memory
RC	Radio Controller
RE	Resource Element
RF	Radio Frequency
ROS	Robotic Operating System
RPi	Raspberry Pi
rr	range resolution
RSRP	Reference Signals Received Power
RSRQ	Reference Signal Received Quality
RSSI	Received Signal Strength Index
RTF	Ready-to-Fly
RTLS	Real-Time Location System
RTT	Round-Trip Time
RX	Reception
SBC	Single Board Computer
SDB	Smart Data Broker
SF	Spreading Factor
SFD	Start-Frame-Delimiter
SINR	Signal to Interference plus Noise Ratio
SLAM	Simultaneous Localization and Mapping
SoC	System-on-Chip
SSAS	Smart Selective Antennas System
STA	Station
TCO	Total Cost of Ownership
TDoA	Time Difference of Arrival
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
ToA	Time of Arrival
ToD	Time of Departure
ToF	Time of Flight
TP	Transmission Power
TRP	Time of Reception Poll
TRR	Time of Reception of Poll
TSP	Time of Sending Poll

TSR	Time of Sending Response
TSME	Two-Step M-Estimator
TS-SDN	TempoSpatial-Software Defined Networking
TTN	The Things Network
TX	Transmission
TWR	Two-Way Ranging
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol
UL	Uplink
UWB	Ultra WideBand
VSLAM	Visual Simultaneous Localization and Mapping
VLOS	Visual Line of Sight
WGS	World Geodetic System 1984
WMN	Wireless Mesh Network
WSN	Wireless Sensor Network
Wi-Fi	Wireless Fidelity

Abstract

Unmanned Aerial Vehicles (UAVs), also known as drones, are becoming widely used in many scenarios, given their advantages with respect to other kinds of aircrafts, such as, for example, helicopters. In the last years, they have been used for surveillance applications, rescue operations in harsh environments, environmental monitoring, building inspections, aerial photography and more. Moreover, UAVs are relatively inexpensive and have a significant lower Total Cost of Ownership (TCO) if compared to small helicopters, thus making them affordable for many applications. Therefore, in the upcoming decade, thanks to the fifth generation (5G) cellular technology and other radio technologies (e.g., Wi-Fi IEEE 802.11s), they will become crucial for several new use cases (e.g., goods delivery and first aid services). Given the young age of the UAV's industry, there are still many problems to be solved and many possibilities to be explored. Among them, there are many challenges to be overcome regarding communication and positioning systems for drones, which actually heavily limit their use cases and operating capabilities, slowing down the growth of the market in this new sector. This opens the way to many significant research activities.

A key element of any drone-based application is the communication system between the UAV and its pilot, which must be always reliable and effective. Moreover, for new drone applications and scenarios, all the UAV-to-X connectivity will play a crucial role in the diffusion of UAV-based mass services. This is especially true for Beyond Visual Line of Sight (BVLOS) applications, where, given the absence of direct visibility between the *in-flight* drone and the control center or the pilot, as well

as with others UAVs or smart devices located on the ground, many advanced control and communications mechanisms are needed, in order to maintain and guarantee the safety of this complex missions over large and, often, harsh environments.

Different environments may need different technologies: for example, in indoor or obstructed scenarios, a Global Navigation Satellite System (GNSS) does not work, so a different positioning system is needed. The same happens with communication systems, since drones need to connect to heterogeneous networks on the basis of many quality index parameters and application requirements (e.g., low latency, high throughput, long range, etc.), in order to ensure seamless communication, even when switching among different protocols and network interfaces. Finally, looking at further applications with hundreds of UAVs flying in a city, an effective communication system has to guarantee Command and Control (C2) link, enabling also applications which require real-time bidirectional communications both with ground units as well as with other UAVs.

This doctoral thesis explores different approaches to enhance these critical aspects, with the aim to provide novel solutions to overcome such limitations on the UAVs of the future.

More in detail, regarding the UAV-to-X connectivity, the design and development of a system easing and enabling heterogeneous wireless communications for various UAVs' applications is presented, together with its theoretical modeling, with the aim to provide a tool suitable to evaluate the performance of the proposed solution under different tasks. Moreover, the developed system is exploited as a starting point for the UAV integration of a novel designed hybrid LoRa-IEEE 802.11s opportunistic mesh network, as well as for an advanced 4G cellular connectivity antenna system optimized for UAVs, aiming at improving the communication reliability.

With respect to the UAV localization problem, several approaches are investigated involving the usage of different radio and visual technology. In detail, a preliminary overview of the Ultra Wide Band (UWB) protocol applied to GNSS-denied environment localization is carried out, followed by the research on the usage of the (cost-effective) Wi-Fi technology for static and mobile localization purposes, exploiting different radio approaches. Finally, a novel developed hybrid radio and visual local-

ization framework for localization in harsh environment is presented, then evaluating its performance in real critical environments.

Introduction

The adoption of UAVs is steadily increasing year by year, targeting several applications related to different use cases, spacing form the most common aerial video and photography, to the environmental monitoring, as well as building inspections and finally research and rescue applications. However, there are a lot of possible new UAV applications to be yet explored, therefore the main market analysts predict a further growth of the drone industry in the next decade [2], when thousands of drones are expected to fly above (being heterogeneously connected) *smart cities* to provide different kinds of services, ranging from distributed surveillance up to the delivery of goods or first aid missions [3].

Although these *futuristic applications* can already be applied under controlled experimental environments equipped with specific infrastructures deployed with the aim to validate the feasibility and current technological limits, there are still many problems to overcome when dealing with mass market adoption of such scenarios. More in detail, among all the UAV-related research topics, involving the development of new high-density lightweight batteries, as well as, advanced control algorithms, this doctoral thesis focuses on the actual problems related to the existing wireless communication and localization solutions.

More in detail, wireless connectivity for UAVs can nowadays provide a reliable enough wireless communication link between an *in-flight* drone and its pilot, control center or others *in-flight* UAVs in all those scenarios where there is direct Visual Line of Sight (VLOS) between the two parts. This explains why, nowadays, most of the applications have been mainly carried out in VLOS conditions, where exist-

ing solutions are already reliable and valuable. Although VLOS flights still have a huge market to conquer, most of the drone manufacturing companies are nowadays targeting an enhancement of the adoption of UAVs in Beyond VLOS (BVLOS) scenarios, thus allowing complex missions and enabling advanced services over large areas. However, things can get complicated in this complex scenarios, since in BVLOS missions there is no direct visibility between the two parts, therefore the usage of alternative communication solutions is needed.

With the aim to overcome such limitations and enable new use cases, companies, researchers and start-ups investigated the adoption of different wireless communication protocols, looking for new UAV-to-X connectivity enabling technologies. More in detail, short range communication protocols, like Wi-Fi and Bluetooth Low Energy (BLE), are often used for both the UAV-to-Ground connectivity, as well as, UAV-to-UAV and UAV-to-Device connectivity, while typically long-range communications links, like ZigBee and LoRa, are often used as backup solutions for UAV-to-Ground communications.

However, sometimes these communications links are not enough, especially in BVLOS flight missions, where, as can be noticed in the representative scheme shown in Figure 1, it is often necessary to rely on third-party networks, like existing Wi-Fi or cellular networks, in order to achieve a suitable wide coverage and service quality. One of the last trends regarding the UAV connectivity is related to the integration of 4G Long-Term Evolution (LTE) and 5G NR cellular networks on-board of these devices, as well as the adoption for both C2 and applications-related data exchange. Examples might involve real-time video feeds kept from *on-board* cameras, data collection from Internet of Things (IoT) sensors deployed in the surrounding environment, and, for specific applications, providing the connectivity to ground-located devices located in the area where the UAV is flying in. Therefore, a strong and reliable bi-directional communication link would be needed in these BVLOS-like scenarios.

Commercial UAV manufactures investigated the usage of existing cellular networks for BVLOS-like missions, resulting in high-end quad-copters already able to exploit cellular technology [4] to extend their flight missions' operating range. However, as of today, these solution use 4G LTE connectivity only as a backup solution

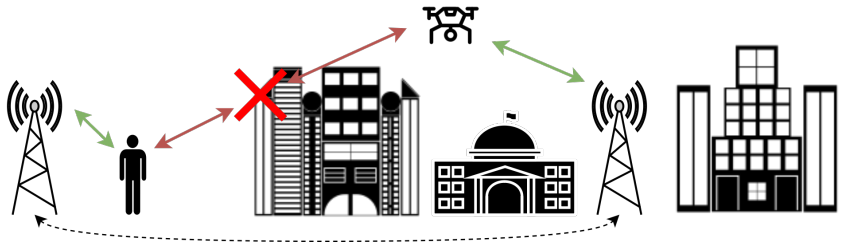


Figure 1: Generic cellular connectivity-enabled UAV BVLOS scenario.

in case of failure of traditional *point-to-point* wireless communication links between the drone and the other involved parties. This is because, despite a widespread adoption of cellular networks for terrestrial applications (whose maturity and reliability is widely known and proved), the adoption of the same paradigms for aerial applications still needs to be thoroughly investigated, since currently deployed cellular networks are optimized for a terrestrial usage.

In detail, among several technical aspects of these networks, their deployment and frequency reuse plans between nearby cells have been optimized to avoid inter-cells interference on ground-located connected terminals. Hence, since UAVs can fly at several altitudes (with the most common commercial drones typically flying close to a 100 m Above Ground Level, AGL, altitude), a cellular terminal located on a drone will feature a direct visibility with multiple Base Transceiver Stations (BTSs) of nearby cells, especially in geographically flat environments or in highly density populated regions (where several BTSs are deployed in a limited region). Hence, this often implies a strong inter-cell interference, since the system receives signals from the neighboring BTSs, whose effects have been experimentally revealed through *in-flights* cellular network signals measurements carried out by several entities [5,6], and leads to a reasonable decrease in the radio signal quality metrics, in the end affecting the network's effective performance and stability, threatening the safety of critical BVLOS missions.

In this doctoral thesis, the challenges and limitations related to UAV wireless communication have been investigated, looking for innovative and novel solutions to overcome these technical constraints, in order to enable the usage of different kinds

of connectivity solutions for various UAV applications. All the strict requirements related to UAVs have been taken into account, in order to provide both theoretical concepts and tools, along with real architectural implementations allowing to experimentally evaluate the performance of the proposed solutions.

Asides the critical wireless communications aspects, another relevant and challenging research field regarding UAVs is related to GNSS-denied localization techniques, since these solutions can allow complex BVLOS missions in harsh environments where no GNSS services are available. Among the most used localization solutions, the most commons rely on the usage of radio-based localization techniques, mainly based on commercial protocols, like Wi-Fi, BLE and, in the last years, UWB. In fact, while the reliability and accuracy of UWB-based solutions have been provided and demonstrated by several commercial solutions released on the market for industrial and robotics applications despite the higher initial deployment costs, there is still a lot of research interest with regard to Wi-Fi-based localization techniques, given their widespread infrastructure diffusion and a consequent significantly lower initial deployment cost to overcome.

However, the usage of Wi-Fi for localization presents several challenges that must still be overcome, both in terms of expected accuracy and also operational range of Wi-Fi-based localization solution. In the last years, the adoption of Wi-Fi for localization purposes has been exploiting two possible directions: (i) Wi-Fi-based range estimation and (ii) Wi-Fi fingerprinting. The first solution, similarly to the principle of GNSS and UWB, exploits the concepts of signal propagation in order to use the Received Signal Strength Index (RSSI) or Time of Flight (ToF) to estimate the distance between the target device to be localized and at least three Wi-Fi Access Points (APs), thus allowing to compute its position. Instead, the Wi-Fi fingerprinting associates the Wi-Fi APs MAC addresses and their RSSI values with the geographical information of the environment, namely the position on the map where the measurement has been done; therefore, once a significant measurement map has been built, it is possible, using different Machine Learning (ML) techniques, to localize devices within this environment based on the observed RSSI and MAC addresses of the Wi-Fi APs.

These approaches have different applications, which, based on the environment and requirements, can be suitable enough. However, for UAV localization, their performance might not be suitable enough, therefore requiring further investigation to determine which are the expected accuracy of such solution on a moving UAV flying in a GNSS-denied environment. In this doctoral thesis, the concepts behind radio-based localization using both UWB and Wi-Fi have been investigated and evaluated through some real experiments applied to UAV localization, with the aim to determine which kind of precision such system can achieve in a GNSS-denied environment. Moreover, advantages and disadvantages of both these systems are illustrated, further discussing and motivating when a solution fits better with respect to the other.

Moreover, a novel hybrid localization technique, involving the usage of both Wi-Fi and Light Detection and Ranging (LiDAR) is proposed, aiming to obtain both the benefit of visual localization techniques together with those of Wi-Fi-based localization approaches, thus compensating their limitations if used as a stand-alone solutions.

Finally, viable future research directions suggested by the results detailed and discussed in this doctoral thesis are investigating, enlighting how to possibly improve and extend the research directions which have been taken into account.

Thesis Structure

This doctoral thesis is structured as follows.

In Chapter 1, a background analysis related to the main UAV wireless connectivity solutions and GNSS-denied localization techniques is provided. More in detail, an initial overview on existing commercial UAV-to-X connectivity solutions is provided in Section 1.1, with a study on relevant literature works regarding the design, development and modelling of gateways (GWs) enabling the usage of multiple heterogeneous communications protocols for both IoT and aerial connectivity being provided. Then, a deeper analysis of existing wireless protocols already adopted on UAVs is carried out, followed by an analysis and study of the usage of cellular networks to enable drones connectivity in BVLOS scenarios, aiming to find currents limits and possible

solutions to overcome them. Then, Section 1.2 investigates on technologies and literature advancements available on the topic of GNSS-denied localization, focusing on pure radio-based localization techniques using different commercial wireless protocols, as well as hybrid radio and visual localization techniques to compensate the disadvantages of both pure radio and pure visual localization techniques.

In Chapter 2, the design and deployment of a Multi Interface Gateway (MIG) designed for IoT and UAV-oriented integrations is discussed, illustrating the MIG architecture, the chosen network communication protocols, the analytical performance modelling and the simulator developed aiming at providing a tool allowing to estimate the throughput of the system on the basis of different parameters for various use cases. Finally, some experimental results of both IoT and UAV applications of the developed MIG are provided.

Chapter 3 presents a hybrid wireless mesh communication system based on Wi-Fi and LoRa protocols, which can been integrated on board a UAV through the usage of the MIG discussed in Chapter 2. More in detail, the hybrid mesh solution definition is illustrated, presenting the overall protocol architecture for several UAV-to-X connectivity scenarios. Additionally, the design of a portable field deployable fully Wireless Mesh Network (WMN) for UAV swarming BVLOS applications, is proposed and investigated.

In Chapter 4, a novel approach to improve the reliability of 4G LTE cellular connectivity for UAV BVLOS applications is presented. More in detail, the proposed system is based on a hardware extension of the MIG prototype discussed in Chapter 2, which, through the usage of custom-made selective antennas controlled by smart selection algorithms, allows to enhance cellular connectivity over existing 4G LTE cellular networks. Moreover, the overall system architecture is detailed, and the achieved experimental results of the developed solution are illustrated.

In Chapter 5, multiple GNSS-denied radio-based localization systems developed with the aim to provide different UAV localization solutions (namely, UWB- and Wi-Fi-based) with different deployment costs, are illustrated, together with their experimental performance evaluations, as well as possible improvements and use cases.

Chapter 6 extends the radio localization techniques analyzed in Chapter 5 to the

combination of radio localization-aided techniques by exploiting visual mapping solutions, in detail aiming at mitigating the weakness of both solutions in a novel robust framework that combines the strength of both localization approaches. More in detail, the architecture, implementation and experimental results of a hybrid LiDAR and Wi-Fi-based localization technique are illustrated, together with some possible enhancement and use cases.

Finally, in Chapter 7, the main contributions of this doctoral thesis are discussed.

Chapter 1

State of the Art

Drone technology become particularly popular in the last decade, thanks to the advancement in several areas which, together with the reduction of the overall costs of such systems, significantly contribute to their diffusion in multiple fields, given the possibility and substantial improvement that these devices introduced. Several open source platforms and solutions has been developed and released in the last years [7–11], allowing to further push the research and applications in completely new fields. This has lead to new challenges and new technological limits to overcome, that, together with the definition of new applications, must be addressed in the upcoming future in order to open new innovative scenarios.

Besides the plethora of open problems to solve in the world of UAVs, this doctoral thesis focused on the wireless communications and GNSS-denied localization topics, aiming at investigating, developing and improving existing solutions to allow new use cases and build a platform suitable for heterogeneous applications of UAVs in many fields related to the IoT.

Therefore, in the following, the most relevant available state of the art solutions are investigated and discussed. More in detail, the study of the existing literature is divided according two main categories: (i) UAV-to-X connectivity and (ii) GNSS-denied radio-based localization techniques, with the first one investigating the existing solutions and research activities in the field of UAV connectivity, while the latter

evaluating the main GNSS-denied localization approaches for UAVs.

1.1 UAV-to-X Connectivity

In order to allow UAV-to-X connectivity, in detail involving both UAV-to-UAV connectivity and UAV-to-Ground connectivity, but able to also involve people (namely the pilot), the environment, like IoT sensing nodes or existing network infrastructure; several wireless communication protocols must be implemented on a UAV, allowing both interoperability and scalability. More in detail, in order to enable several applications, a platform able to provide several communication protocols and suitable to be installed on different kind of UAVs must be developed, since every protocol and its applications scenarios requires to carefully consider several aspects, thus looking for potential *trade-offs*.

Nowadays, the most widely adopted wireless communication protocols, which are summarized in Figure 1.1 and allow to enable connectivity between several kind of devices, like sensing nodes, are: BLE for *short-range* communications [12]; IEEE 802.11 (Wi-Fi) and ZigBee [13] for *medium-range* communications; and NarrowBand-IoT (NB-IoT), cellular (e.g., 4G LTE and 5G NR), Long Range Wide Area Network (LoRaWAN), and Sigfox [14] for *long-range* communications.

In order to make these protocols inter-operable and manageable by a MIG located on a flying UAV, their characteristics have to be carefully taken into account [15]. Typically, two entities support communication protocols interoperability: bridges [16] and GWs [17]. GWs should be preferred, with respect to bridges, as they support intelligent data routing, queuing policies, and data analysis mechanisms for heterogeneous (in terms of resources and constraints) communication interfaces [18]. Hence, the design and deployment of intelligent GWs plays a crucial role in the definition of the UAV-to-X connectivity with the aim to enable also advanced aerial IoT scenarios.

Even though there are different commercial GWs with multiple network interfaces available on the market (see, for example, [19]), they typically present relevant drawbacks, such as: high costs; limited communication interfaces available on the devices; “closed source” nature; high dimensions and weight, therefore physi-

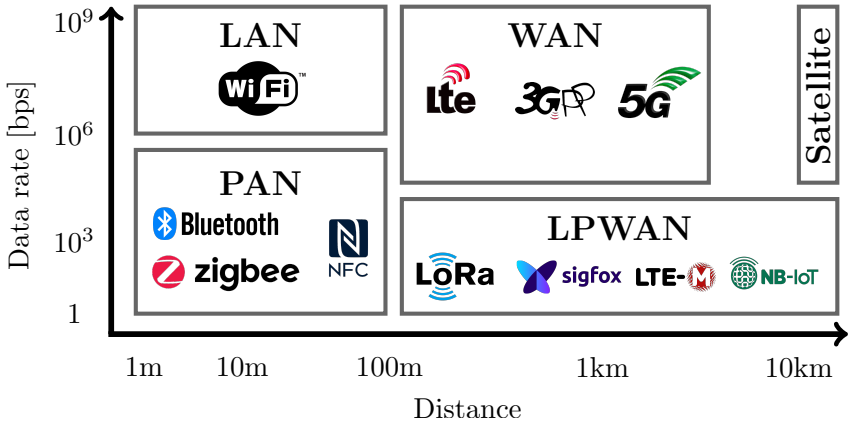


Figure 1.1: Communication protocols of interest for heterogeneous IoT scenarios.

cal problems in the integration on different kinds of UAV. The "closeness" of these solutions, together with their overall sizes, prevent the design of new routing rules, new communication interfaces and data processing features, as well as the connection of new communication protocols (e.g., the serial protocols used for the Flight Controller (FC) connection to the on-board MIG [20]) and the their adoption in complex network scenarios.

Therefore, the design and implementation of modular and open GW architecture for both UAV and IoT applications, like the one proposed in this doctoral thesis, is crucial. Moreover, in order to evaluate the performance and the applications of these GWs in the UAV and IoT worlds, where it is not always possible to perform measurements and test in a real world testbed, given the regulations behind UAVs (i.e., defined by the European Union Aviation Safety Agency, EASA [21] [21]), an analytical characterization or model tool can provide useful performance estimation and evaluations. Regarding the modelling of this kind of GWs, in [22] a hidden Markov model-based approach for latency-aware and energy-efficient computation offloading in fog computing-like scenarios is proposed.

Markov queuing models have also been adopted in [23,24], looking for a trade-off between energy consumption and latency in task assignment in next-generation

systems, as well as in [25–28]. All these research works focus on the use of Markov chain-based models for the characterization and optimization of a few network parameters only for specific IoT system aspects, while in this doctoral thesis the focus is oriented on more general applications facing from the aerial IoT field to the UAV-to-X connectivity, all taking into account the adopted communication protocols limits depending on the constraints imposed by its interaction with other protocols, e.g., when the input information flow comes from another communication interface the MIG is equipped with.

Therefore, in this doctoral thesis, the design and implementation of a MIG equipped with several communication protocols usable for both IoT and UAV applications is presented, followed by its analytical modelling and characterization, used to simulate and evaluate its performance in possible real applications, thus allowing to significantly decrease the cost of the real testing and design phases.

1.1.1 UAV Wireless Connectivity

Nowadays, commercial UAVs rely on different wireless communication protocols to transmit or receive essential data from the ground, as an example, the real-time First Point of View (FPV) video stream from the drone, the telemetry data and, of course, drone's input commands sent from the Radio Controller (RC) handled by the pilot or sent from the command and control center.

Most existing solutions adopt the Wi-Fi protocol (operating in the 2.4 GHz and 5.8 GHz frequency bands) as the communication link between the two parts, thus providing a bidirectional communication channel sufficiently reliable to allow the pilot to flight in an obstacle-free environment for several hundreds of meters. More advanced UAVs instead, use proprietary communication protocols, such as OcuSync [29], developed by DJI, able to guarantee a VLOS operational range of several kilometers. Others high-end products are already working on 4G LTE and 5G NR cellular networks for high definition video feeds or as a backup link, given the current limitations of such system, while telemetry and C2 data are exchanged using different solutions, like Spektrum [30], FrSky ACCST [31], Futaba [32]) and the newer LoRa-based ExpressLRS [9], mostly operating in the 2.4 GHz or others free Industrial, Scientific

and Medical (ISM) bands, like the 868 MHz in Europe.

Outside of specific communication protocols tailor made for UAV applications, several other wireless communication protocols have provided to be usable for UAV-to-X connectivity. Among them, LoRaWAN protocol has found some applications in UAV-to-Ground communication links, in particular as a telemetry provider, given the strength of LoRa modulation and LoRaWAN protocol implementation. In [33], LoRaWAN is employed for communications among flying UAVs and the existing LoRaWAN ground network, in particular to transmit essential telemetry data (such as GNSS latitude/longitude/altitude, drone speed, and heading direction) needed to safely handle a UAV in BVLOS scenarios.

Another UAV-to-Ground LoRaWAN-based communication application has been proposed in [34], in which the use of multiple UAVs with *on-board* sensors for air quality monitoring applications is discussed, further providing additional system's operational results. An alternative usage of LoRaWAN is discussed in [35], where multiple UAVs are used as flying LoRaWAN GWs to provide network coverage to ground LoRaWAN-based end nodes. A similar approach is presented in [36], where multiple flying UAVs, each one equipped with a LoRaWAN GW, cooperate to form a complex LoRaWAN-oriented urban surveillance system to provide an efficient and selective network coverage. Another UAV-enabled flying LoRaWAN networking approach is discussed in [37], with focus on disaster management applications, combining flying LoRaWAN GWs, with Wi-Fi communications from UAV to ground AP, and LoRaWAN end nodes, with the goal to follow the emergency operators to collect data and positions.

Among Wi-Fi-based applications involving UAV-to-X connectivity, in [38] a UAV-based emergency communication network for post-disaster management is discussed, with UAVs equipped with a Raspberry Pi (RPi) as on-board Wi-Fi-enabled device, thus providing connectivity to ground devices. A simulated double-layer mesh network, designed and evaluated for disaster recovery scenarios, is proposed in [35], where UAVs act as GWs in a Wi-Fi *ad-hoc* network for the traffic generated between mobile LoRaWAN nodes and a remote ground station.

Moving toward the wireless communications between in flight UAVs, therefore

the UAV-to-UAV branch of UAV-to-X connectivity, commercial systems can mainly rely on Wi-Fi meshing oriented protocols for VLOS short/mid-range communication links, aided by long-range protocols (e.g., LoRa, ZigBee) for communications over a large area, still mainly be limited to VLOS communication links.

Considering the adoption of Wi-Fi in UAVs swarms, in [39] an integrated solution for an autonomous flying Wireless Mesh Network (WMN), involving swarms of small UAVs, is presented. In detail, UAVs build an IEEE 802.11s-based network to let two end systems interact through intermediate airborne relays. These relays correspond to UAVs of the swarm and all UAVs act as APs.

The concept of Flying Ad-hoc NETworks (FANETs) is discussed in [40], where additional link quality routing metrics are introduced and a network simulator-based performance analysis is discussed. Algorithmic and theoretical aspects for autonomous placement of relay nodes in a mesh network are investigated in [41]. In detail, the authors aim at maximizing the network throughput, but mainly using flying UAVs as coverage extenders for ground nodes and requiring the knowledge of the three-dimensional morphological conditions of the environment. Because of this, UAVs only act as relays for data to be shared among ground stations, instead of building a flying network for flying data routing.

In [42], a dynamic mesh network needed for data transfer among high mobility nodes (i.e., RPis running Better Approach To Mobile Ad-Hoc Network advanced, B.A.T.M.A.N. adv [43]) is presented. In detail, the authors discuss their experimental results in a range between 0 m and 60 m, with a maximum effective 1-hop range of approximately 130 m. Nevertheless, their architecture requires multiple GWs connected with a ground control center, thus remaining a ground mesh network, with no aerial part. Similarly to other works, even in this case the only available network among the flying drones is based on the Wi-Fi protocol, thus limiting the maximum distance between flying drones. In [44], a solution for orchestrating and synchronizing a swarm of drones is proposed. In detail, their network focuses on a leader node (piloted by a human) that will be followed by the remaining drones composing the swarm, on the basis of the leader's Wi-Fi signal strength. Authors propose to operate a swarm of UAVs without relying on any existing infrastructure, but only on ad-hoc

communications. This approach is attractive if the drones need to strictly follow their leader; at the opposite, it needs to be improved in case if UAVs need to fly independently of each other. Moreover, in order to relay data from the point of interest to the ground control center, a certain number of intermediate UAVs (depending of the distance) will be required, rather than adopting long-range communication protocols.

Finally, the application of TempoSpatial-Software Defined Networking (TS-SDN) to UAV networks, in order to extend their network coverage, is described in [45]. In order to reach this goal, a network of stratospheric balloons can be used to extend network access, as well as involving ground stations and UAVs equipped with mechanically—or electronically—steered beams to establish highly-directional links, even to nearby peers. Moreover, the authors simulate a scenario composed of more than 500 nodes to verify the behavior of different routing protocols.

In this doctoral thesis, a hybrid Wi-Fi LoRa-based mesh for aerial UAV-to-X connectivity in VLOS flight conditions is proposed, with the aim to exploit the onboard MIG interfaces to enable both UAV-to-UAV communications for swarming applications over short and large areas, as well as to enable UAV-to-Ground communications for both aerial IoT and swarm coordination applications. Moreover, the usage of an ad-hoc designed portable fully WMN is investigated, aiming at enabling BVLOS missions in critical environment where a high throughput and low latency link is needed for swarming coordination and LiDAR point cloud data exchange, used for autonomous navigation purposes.

1.1.2 UAV Cellular Connectivity

In order to enable both long-range and wide areas missions, cellular networks are the most logical candidate for BVLOS link-based applications, where a third-party network infrastructure is needed to establish a communication between multiple flying UAVs or between a UAV flying outside of the ranges achievable by the solutions described in Subsection 1.1.1.

Regarding the usage of cellular connectivity for UAVs, both for UAV-to-UAV and UAV-to-Ground connectivity in BVLOS missions, several investigations have been carried out with the aim to verify the feasibility of UAVs enabled by cellular con-

nectivity, in detail through the use of 4G cellular networks and, recently, 5G cellular networks.

In [5], 4G LTE signal behaviour data from a UAV-mounted cellular modem has been collected, showing a decrease of the main radio quality indexes at higher altitudes, and concluding that higher levels of interference incoming from the neighboring cells (due to their signals' free-space propagation) return a relevant signal degradation with respect to ground-located cellular equipment.

Authors in [46] managed to achieve similar results in a different environments, while the overall limit of existing 4G LTE cellular networks have been also remarked in [47]. A similar investigation has been carried out in [6], confirming that: (i) at altitudes higher than 100 m AGL, the Free Space Path Loss (FSPL) of radio signals transmitted by nearby cell towers combined with the antennas side-lobes of the BTSs, can significantly impact the performance quality and reliability of existing 4G LTE networks for aerial connected UAVs; and (ii) increasing the flight height, the cell association patterns become more complex, thus afflicting the network stability.

In order to overcome these problem in deploying 4G LTE cellular connected UAVs, the 3rd Generation Partnership Project (3GPP) Release 15 [48] introduced enhancement aiming at mitigating the interference problem [49,50] where, despite technical studies and proposed interference mitigation approaches, it is concluded that, in order to fulfil the proposed enhancements, an overall refactoring of assumptions, models and techniques used in the design and deployment of cellular networks is needed, and that only newer generation of cellular networks (e.g., 5G and 6G) might take into account and satisfy them. Therefore, in order to exploit existing 4G LTE cellular networks, different solutions must be found with the aim to enhance UAV connectivity, at least in the initial phase.

In [51], an analysis of the neighboring cells interference problem and a model for aerial cellular connectivity have been analyzed and developed, together with two possible solutions to make existing 4G LTE cellular networks more reliable: (i) the use of directional antennas mounted on the UAV and (ii) the use of interference cancelling techniques. With regard to the *first* solution, the adoption of 2, 4, and 6 directional antennas has been evaluated through a simulator, then showing promising results in

the case the number of directional antennas is increased, but not being validated as a physical deployed system. Another tri-dimensional network simulator for connected UAVs is proposed in [52] suggesting (through extensive numerical results) the use of directional antennas mounted on the UAV to possibly mitigate interference problems, while another 3D model to estimate 4G LTE cellular networks performance for connected UAV is proposed in [53], in the end suggesting the use of millimeter waves (mmWaves) to overcome and control the interference on newer generation cellular networks, but lacking of applicability in existing 4G LTE networks. An alternative approach not relying on *on-board* directional antennas is proposed in [54], where the UAV's path is planned based on the radio signal quality aiming at optimizing the UAV's route, thus ensuring a safer connectivity reliability. However, this approach (i) requires the definition of a radio map, built through filed measurements, and (ii) cannot always be applied, since for some specific flight missions it is not possible to change the flight path.

Regarding the usage of directive antennas, in [55] a system composed by five reconfigurable directional antennas optimized for UAV *Air-to-Ground* communication is proposed and evaluated. Nevertheless, despite good results in both simulations and experimental evaluations, this system is designed for the 5.8 GHz band only, thus being not suitable for 4G LTE cellular communications. Moreover, the experimental evaluation in [55] has been conducted with the UAV at a fixed height and position only, and the ground base receiver in different positions, thus not simulating the possible topology and structure of a traditional cellular network.

In order to contribute to the cellular connectivity integration for UAV applications, this doctoral thesis investigated the usage of 4G LTE cellular networks with traditional omni-directional antennas cellular systems, then the design and development of a system based on selective antennas is proposed and evaluated, with the aim to improve cellular connectivity with existing cellular networks.

1.2 GNSS-denied Localization Techniques

Localization in environments where GNSS services are not available has always been challenging and complex, since alternative solutions must be found, mostly relying on a different radio infrastructures of on-board localization and mapping solutions. In the last years, two main approaches have lead the research in this field: (i) radio-based localization through the use of several commercially available wireless communication protocols, and (ii) on-board Simultaneous Localization and Mapping (SLAM) algorithms applied to different type of data gathered by cameras, LiDARs or RAdio Detection and Rangings (RADARs). The key differences between the two techniques are mainly related to the knowledge on the environment where the target has to localize itself IN, with most of the radio-based techniques requiring an *a-prior* knowledge of the exact position of APs and/or anchors. Instead, SLAM-based solution (as the name suggests) can simultaneously map the environment and localize them self within the constantly build and updated map of the surrounding environment, thus requiring a minimal *a-priori* knowledge of the environment. Both approaches are investigated in the following, but, given the MIG implementation and architecture defined for UAV and IoT applications in this doctoral thesis, most of the analyzed state of the art solutions and concepts of radio-based localization rely on commercially available protocols, as well as hybrid radio and SLAM localization techniques.

1.2.1 Radio-based Localization for UAVs

Radio-based localization has always been the most investigated and adopted localization technique for GNSS-denied environment where several anchors or APs are available, depending on the used wireless technology. The most traditional approach used for commercial wireless communication protocols, like those implemented in the MIG proposed in this doctoral thesis, namely the Wi-Fi, BLE and even cellular networks, is related to the usage of range estimation techniques between the known position of the APs or BTSs based on the RSSI, that, using the log-distance path loss model, allow to determine the distance between the transmitter and receiver. Then, through the use of at least three range estimations from different APs, it is then pos-

sible to perform a multi-lateration bi-dimensional position estimation of the target to localize.

Although several research activity, like in [56] and [57], have been adopting the RSSI ranging techniques for localization purposes in the last decades, their relative accuracy still remain an open issue, since it is often related to the radio-environment noise floor, external interference and LOS or Non LOS (NLOS) signal propagation. Therefore, the accuracy of this localization technique purely rely on the accuracy on the range estimation model. Most of the proposed work, such as [58], manage to achieve a few meters error in both LOS and NLOS environment using Wi-Fi RSSI localization approaches based on different signal propagation models to properly estimate the LOS of NLOS nature of the incoming signal. More in detail, for static objects where multiple RSSI measurements can be gathered and the channel LOS or NLOS nature can be characterized, it is possible to achieve an estimation error of a few meters, as presented in [58]. However, using more complex filtering solutions, as well as a combination of more measurements and ML techniques (such as in [59]), it is possible to achieve better results, with some proposed solutions detailed by [60], able to achieve an error below 1 m adopting a Wi-Fi RSSI fingerprinting-based approach.

However, for moving objects in an indoor environment, like a small flying UAV, where it is possible to gather only a few (or only one) RSSI measurements, the position estimation error can significantly increase and, in order to keep it suitable for UAV localization applications, it is necessary to have a significant amount of APs to obtain a sufficient position estimation, as detailed by the framework's simulation in [61]. A real experimental evaluation of a moving target localized trough RSSI measurements is instead proposed in [62], where authors managed to achieve, in a very specific environment and experimental condition, a localization error between 1 m and 5 m with respect to the ground truth, depending on the number of gathered measurement and adopted filtering technique.

Besides the RSSI range estimation technique, more consistent and reliable approaches are available on the market, mostly relying on more advanced techniques that require specific hardware-implemented features, like the ToF measurement of the

transmitted radio signals, which allows a more accurate range estimation between the transmitter and receiver, especially on high frequency and high bandwidth protocols, like the UWB or 5 GHz Wi-Fi protocols. More in detail, in [63] a localization techniques for smartphones is proposed using the Fine Time Measurement (FTM) on the 2.4 GHz Wi-Fi IEEE 802.11mc protocol, aided by the Angle of Arrival (AoA) [64] measured on the 5 GHz band, allowing to achieve a promising positioning error below 1 m in a LOS 8×9 m arena. However, in order to achieve these results, authors in [63] performed several range measurement for each tested position, therefore not evaluating the impact of mobile targets. A similar approach, still for smartphones, is shown in [65], where FTM is used in combination of RSSI fingerprinting technique to estimate their position within a 500 m^2 area, obtaining an average position estimation error below 1 m. However such solution require an intensive and tedious pre-measurement phase, which is not always feasible. Relevant results are instead achieved by the novel approach proposed in [66], where the combination of LOS and NLOS channel estimation with the usage of 5 GHz band APs allowed to achieve an average position estimation error between 1 m and 2 m on a smartphone in different kind of environments.

Moving to the application of Wi-Fi FTM on UAVs, in [67] a solution able to achieve a sub-meter positioning error with four expensive 5 GHz APs within a 5×5 m arena, with the drone hovering in a static position, is proposed. However, the size of the testbed are far from a real application for UAV localization. Different is the approach proposed in [68], where 2.4 GHz affordable ESP32S2-enabled development boards are used as APs and, together with the proposed ML-filtering solution, manage to obtain a 1.5 m positioning error within a 5×10 m testing area.

Instead, regarding the UWB technology, adopting *Commercial-Off-The-Shelf* (COTS) solutions [69] is already possible to push the localization error down to $10 \div 30$ cm, depending on the main radio parameters and equipment. Some evaluations related by the adoption of UWB for localization purposes have been investigated in [70–72], confirming the good positioning accuracy as well the limited operating range of the solution, suitable only for small environments. Regarding the application of UAVs, extensive tests have been presented in [73], confirming the overall good performance

of such solutions, as wheel as the limited operating ranges.

Among other radio localisation techniques, the AoA is significantly increasing its research interest: this is due to the fact that, for some use cases, if combined with the FTM range estimation (such as in [63]), it allows to estimate the position of a target with just one anchor or AP, which, as downside, requires a higher anchor complexity, due to the need to integrate multiple antennas at specific distances, thus more RF-to-Signal converters in the device modem which increase the cost of the hardware equipment design and production. However, both these solutions (namely: FTM and AoA) are affected by the signal reflections due to the environment, which can significantly influence both the range and angle estimation.

1.2.2 Hybrid Localization Techniques

Sometimes, the usage of pure-radio-based localization is not enough cost effective to implement, especially in large environments where the cost of the infrastructure can be relevant, with the usage of on-board SLAM-based solution possibly being more reliable and cost effective. Therefore, relying only on Wi-Fi-based localization techniques, cannot provide a sufficient accuracy value in term of distance error between the ground truth position and the estimated one, since for critical BVLOS missions (involving UAVs and other robotics platforms), the knowledge of the position with an half-a-meter error is crucial. By the way, LiDAR SLAM-based solutions can already achieve such accuracy [74], allowing autonomous complex mission in harsh environments.

In the last years, the research efforts in the field of Computer Vision (CV) applied to location recognition dominated the Visual SLAM (VSLAM) scene [75], with several solutions being developed and released for robotics localization applications. One of the widely known descriptors is SURF [76], which utilizes visual features and geometric transformations to establish correspondences between images, enabling accurate localization. However, the limitations of cameras in low light conditions, weather changes, and the lack of depth information have led to an increasing interest in LiDAR-based methods.

So as, LiDAR sensors gained appeal due to their immunity to lighting variations

and their ability to provide rich 3D information about the environment, and became particularly valuable for applications where accurate localization is crucial, such as autonomous driving or mapping. As an example, in [77], a framework denoted as PointNetVLAD, which first processes each point of a point cloud individually using PointNet to extract local features, and then aggregates these local features using VLAD, which encodes a global representation of the entire point cloud, has been proposed. In detail, VLAD represents each cluster of local features by computing the residual vectors between each local feature and a set of learned cluster centers. Then, the authors of OREOS [78] proposed a novel approach to address the heavy computational load of 3D LiDAR scans, by projecting them onto a bi-dimensional plane while preserving depth information. This technique reduces the computational burden, making LiDAR more suitable for mobile robots or UAVs without sacrificing the advantages of LiDAR-based place recognition. In OverlapNet [79], a further step has been performed by exploiting different types of information generated from LiDAR scans to provide overlap and relative yaw angle estimates between pairs of 3D scans. The range images are enhanced with information such as normals, intensity and semantic data.

The approach of Wi-Fi fingerprinting for indoor localization has been adopted for several applications, spacing from users' tracking through their smartphones [80], up to the localization of IoT devices in industrial environments [81]. In detail, this solution is well known for its low implementation cost, especially on new devices (e.g., smartphones, IoT devices, etc.) which nowadays all integrate Wi-Fi connectivity (among other communication protocols). To this end, an overview of existing Wi-Fi fingerprinting solutions for localization has been carried out in [82,83], where different localization algorithms using Wi-Fi fingerprints have been illustrated, thus showing how the use of different techniques and ML algorithms can achieve an accuracy of a few meters for indoor localization applications.

However, for autonomous robotics applications, a step forward is needed in order to allow the execution of complex missions in harsh environments. A possible solution might involve the fusion of heterogeneous data sources for localization purposes—as an example, LiDAR, Inertial Measurement Unit (IMU), and visual

odometry algorithms, to be fused with Wi-Fi fingerprints. In [84], Wi-Fi fingerprinting—although focusing on the Channel State Information (CSI) instead of the RSSI—has been combined with VSLAM algorithm. Despite the promising results and relative low cost of the platform, the proposed solution needs a specific Wi-Fi network setup to collect the CSI fingerprints, since this metric is not supported by all the Wi-Fi APs, therefore the solution cannot exploit the Wi-Fi networks available in the environment.

Another similar method intended to fuse Wi-Fi fingerprinting—this time relying on RSSI and MAC addresses—together with a VSLAM algorithm, is proposed in [85], highlighting the performance improvement that the combination of both these technologies returns for indoor localization applications. However, VSLAM is known to poorly work in low light conditions with reflections or dust, also showing a weak accuracy for environments with repetitive patterns (e.g., same wall geometry). Finally, in [86] an approach for a Wi-Fi fingerprinting and LiDAR SLAM fusion technique similar to the one presented in this doctoral thesis is proposed, achieving interesting results, although the Wi-Fi data collection equipment detailed in [86] relies on the use of several smartphones as Wi-Fi scanners, which is impractical for an implementation on a constrained platform, such as on an UAV. Moreover, during data collection, the used platform has been moving at a very low horizontal speed of 0.4 m/s, that, in favor of a richer and more complete Wi-Fi fingerprinting database, makes the initial data collection extremely time demanding, especially over a large area.

Therefore, in this doctoral thesis, the usage of a hybrid Wi-Fi fingerprint and LiDAR SLAM-based localization approach is investigated and proposed, aiming at exploiting the strength of both these solutions in order to minimize the drawbacks that each of these approaches may present if used alone.

Chapter 2

Multi Interface Gateway

This chapter describes the design, implementation, and validation of a novel IoT GW architecture based on COTS components. In detail, this architecture is well-suited for a wide spectrum of applications, encompassing terrestrial IoT scenarios, as well as UAV-enabled heterogeneous connectivity in aerial IoT environments. The focal point of discussion is the prototype development of an advanced, scalable, and modular MIG. This MIG features four distinct heterogeneous communication interfaces, namely: IEEE 802.11 (2.4 GHz) Wi-Fi, BLE, LoRaWAN, and cellular 4G LTE (Cat. 4). Then, in order to accurately predict the experimental performance of the COTS device-based MIG in various scenarios, a *novel Markov chain-based queuing model* of the MIG is derived. This model allows to evaluate the system performance under diverse workloads.

Furthermore, a *Python-based software simulator* of the proposed MIG is developed, with the aim to validate the analytical performance predictions derived from the Markov chain-based queuing model. This approach facilitates a comprehensive comparison between analytical, simulation and experimental results, allowing also to a deeper analysis of the main limitations of the employed communication protocols. Unlike classical approaches, which move from theoretical modeling to experimental validation, in this thesis the research activity moved from a COTS device-based MIG (with a specific architecture) to its analytical and simulation models, allowing further

evaluations possibilities for the design of different applications.

2.1 Overview

Among the transformative technologies and paradigms that have recently revolutionized the approach to device connectivity, the IoT occupies a central position. IoT aims to connect a diverse range of nodes, each equipped with sensors and actuators, across various environments and for a multitude of use cases. Consequently, due to the broad spectrum of IoT applications, such as smart cities, Industry 4.0, aerial IoT and precision agriculture, IoT entities are often structured as complex systems of systems.

In a typical IoT scenario, communication protocols vary widely, spanning from short-range protocols with high data transmission rates to long-range ones with typically lower data transmission rates [87]. Managing this protocol heterogeneity and enabling seamless interactions between different networks is a primary challenge, as well as enabling aerial UAV for enhanced connectivity in the field of aerial IoT and reliable communications. To address this challenge, a vital component within the IoT framework is the concept of GW, which plays a pivotal role in efficiently routing data and facilitating IoT applications. As such, the internal organization and scalability of an IoT GW must ensure the highest level of interoperability in different scenario, from in-field applications, as well as in UAV-enabled applications.

IoT GWs typically work as multi-layered network entities, referring to the ISO/OSI and TCP/IP layered protocol stacks. These GWs provide support for several critical functions, including the intelligent routing of information flows, data processing, the management of seamless connectivity and the implementation of buffering and queuing mechanisms. The importance of these queuing mechanisms becomes evident in the presence of heterogeneous communication interfaces, which often have different data-rates capabilities.

Conversely, heterogeneous networks, characterized by a multitude of nodes that collect data, often via sensors, from their surrounding environments, rely on sophisticated data processing techniques. These techniques are employed to extract rele-

vant information while eliminating redundancy. Furthermore, the processed data is adapted for transmission through low data-rate protocols, such as LoRaWAN. Importantly, this data processing can be integrated into edge GWs, enabling the migration of network intelligence from the core to the network's edge or, in the case of aerial IoT and UAV applications, on-board of a drone.

The key aspects to enable all such scenarios is associated to the central MIG architecture, which must allow both scalability and versatility, needed to enable several applications. The MIG architecture proposed in this thesis is discussed in Section 2.2, where all the software and hardware components are detailed, while in Section 2.3 a Markov chain-based queuing model to describe the behavior of the MIG is derived and described. Section 2.4 presents the evaluation of the performance predicted by the analytical queuing model and compare it with the ones of the developed Python-based simulator. The experimental results of the system in a generic IoT scenario are discussed in Section 2.5. Section 2.7 is dedicated to discussing possible improvements of the MIG. An extended MIG version optimized for UAV-to-X connectivity is finally presented in Section 2.6, where the integration with the UAV platform and FC is shown, as well as some preliminary use cases to enable aerial IoT and UAV heterogeneous connectivity. Such aerial version of the MIG architecture introduced in this chapter is the key element of this thesis, since it enables the hybrid Wi-Fi and LoRaWAN framework proposed in Chapter 3, as well as the novel cellular connectivity enhancement discussed in Chapter 4.

2.2 MIG System Architecture

The first prototypical IoT-oriented MIG implementation is based on a Raspberry Pi 4 (RPi4) Single Board Computer (SBC), equipped with an additional Dragino LoRaWAN hat [88] and a Huawei E3372 USB dongle 4G LTE Cat. 4 modem. Therefore, with these hardware components, the MIG can operate with the following communication protocols: BLE and Wi-Fi connectivity, provided by the interfaces integrated into the RPi System-on-Chip (SoC); LoRaWAN protocol, through the Dragino expansion hat; and 4G LTE cellular connectivity, through the Huawei modem. An over-

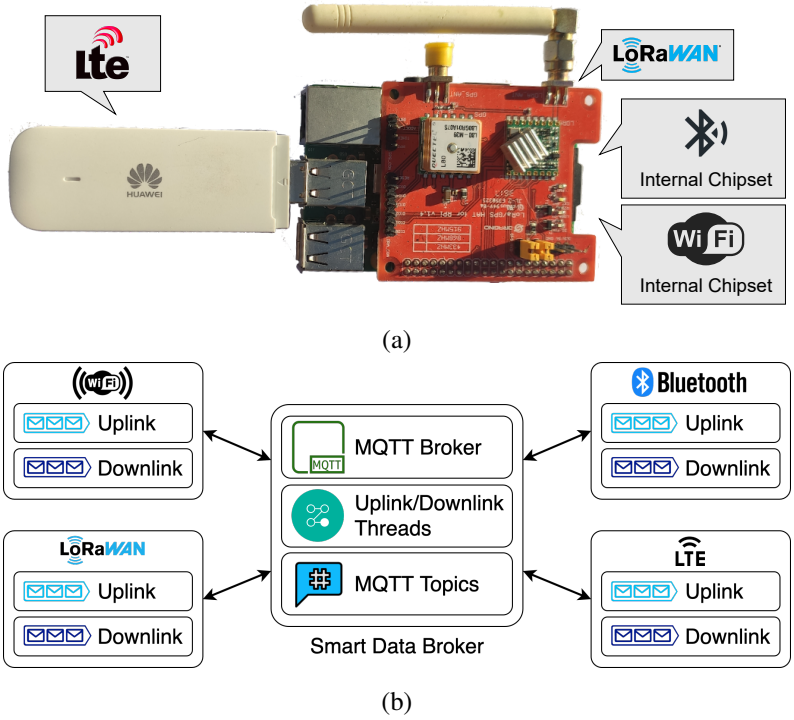


Figure 2.1: MIG: (a) COTS device-based prototype and (b) high-layer architecture.

all view of the prototype of the proposed MIG is shown in Figure 2.1, where (i) the COTS device-based MIG prototype and (b) its corresponding internal architecture are depicted.

Regarding the software layer of the MIG, the designed architecture relies on two high-layer types of modules, namely: (i) a dedicated software entity for each available communication interface, and (ii) an internal routing module, denoted as Smart Data Broker (SDB), that, jointly operating with an internal Message Queue Telemetry Transport (MQTT) broker, is in charge of handling multiple MQTT topics and is used for temporary internal traffic packets' queuing and management purposes. The role of both components, will be illustrated in the following.

More in detail, the main role of the software entities associated with each com-

munication interface is: (i) to properly handle the tasks which may be required by the corresponding communication protocol (e.g., packet processing, payload constraints' validation, transmission policies' adoption, services' execution, UL and DL operation handling, etc.) and (ii) to optimize, convert, and forward data (e.g., coming from *on-field* end nodes connected to the MIG) toward the right interface-specific MQTT topic.

For the sake of clarity, as a preliminary discussion regarding the first IoT-oriented MIG prototype discussed in this chapter, assumes that the information flows withing the GW are generated by several end nodes that exchange the gathered data to the MIG, which then pre-process and transmit them to cloud application servers (ASs) using the cellular or LoRaWAN interfaces. More in detail, the data flow, in the form of packets, from the end nodes to the GW is denoted as downlink (DL) traffic, while from the GW to the end nodes is denoted as uplink (UL) traffic. The MIG detailed in this manuscript allows DL and UL traffic flows on the BLE, cellular and Wi-Fi interfaces, while, with the current LoRaWAN expansion hat, only UL traffic, in detail from the MIG to the cloud AS[1], is possible on the LoRaWAN interface.

In order to allow the proper data handling by the MIG, the data exchanged with the *on-field* end nodes have the illustrative structure shown in Figure 2.2 (at the top). Once received by the MIG, these packets will then be processed by the proper DL interface' handler. The selected handler then "appends" a header field, denoted as ID_{ROUTE}, specifying the routing rule which should be applied by the SDB routing system (e.g., forward to the LoRaWAN interface). Furthermore, in the case of a packet coming from Wi-Fi or BLE nodes (as discussed in the previous paragraph), this packet is extended to include the following fields: (i) the source node's MAC address SRC_{MAC} and (ii) a separator field. The final packet structure will thus possess a general form with a header H and a payload PKT.

Examining the internal routing process, the SDB relies on a standard MQTT broker. This MQTT broker manages various MQTT topics, each overseen by a dedicated UL/DL handler. These handlers are responsible for processing incoming packets and

[1]Class C LoRaWAN nodes [89] can also receive DL traffic from the LoRaWAN's AS (through the LoRaWAN Network Server, NS, and LoRaWAN GWs), but they are typically not used.

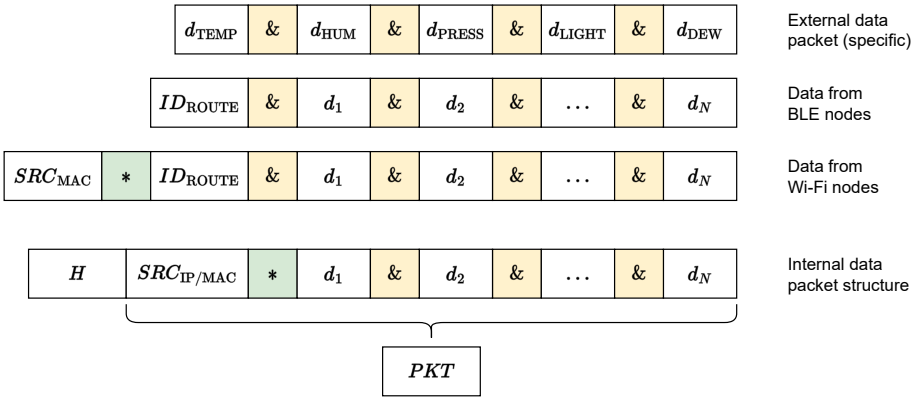


Figure 2.2: Data packets from the different communication interfaces adopted in the proposed MIG.

routing them to the corresponding UL queue of the appropriate output interface, following a *First-In-First-Out* (FIFO) policy. Then, the interfaces' servers, being subscribers to the MQTT UL topics of their corresponding communication interfaces, (i) are notified by the SDB with the updated Data Transfer Units (DTUs), leading to aggregated packets, (ii) perform the required actions on the data, and (iii) execute the final UL operation, forwarding data to the right target device through the proper interface. To this end, it should be noted that, from an operational point of view, the MIG creates a new thread each time a message is notified via the proper UL MQTT topics. In the proposed implementation, the data are temporarily stored inside the RPi4's RAM, thus limiting the processing time and increasing the overall performance.

From an operational standpoint, it is worth noting that each software entity responsible for managing its respective communication interface employs a dedicated parsing technique to encapsulate the retrieved data into DTUs. This process is essential to make the data suitable for constrained protocols. To provide more detail, IP and MAC addresses are subject to a compression that minimizes their storage requirements. Specifically, an IP address is encoded as a single integer, while a MAC address is transformed into its corresponding hexadecimal `HEX` value. This compression strategy proves particularly advantageous in reducing the payload size of

Table 2.1: Routing rules available in the proposed IoT MIG.

Field Value	Routing Destination
`MAC Address`	Specific MAC address of the BLE device
`IP Address`	Specific IP address of the Wi-Fi device
`0`	LoRaWAN communication interface
`1`	Wi-Fi server's default IP address
`2`	BLE server's default MAC address
`3`	LoRaWAN communication interface and Wi-Fi server's default IP address
`4`	LoRaWAN communication interface and BLE server's default MAC address
`5`	Wi-Fi server's default IP address and BLE server's default MAC address
`6`	All available communication interfaces, on their default server IP addresses
`7`	Server reachable through the Cellular Network

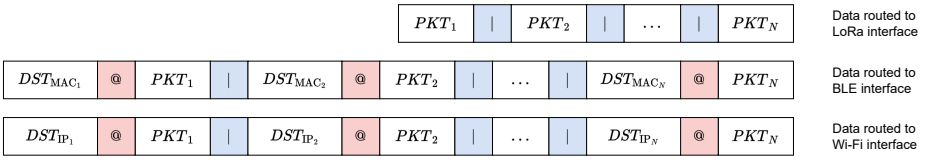


Figure 2.3: Data packets format of the proposed MIG to exchange data with end targets.

LoRaWAN and BLE messages. Finally, depending on the routing identifier, the resulting DTU is sent to the proper communication interface. An illustrative example of routing rules defined internally in the proposed MIG is shown in Table 2.1.

The proposed DTU structure, shown in Figure 2.3, allows processed data to be transmitted by BLE and Wi-Fi communication interfaces, to be inserted in an output packet with a payload composed by N aggregated payloads $\{PKT_i\}_{i=1}^{N}$ separated by

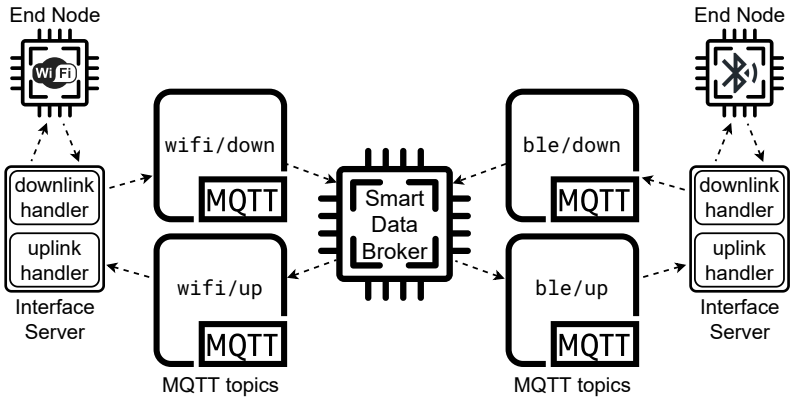


Figure 2.4: Interactions among Wi-Fi/BLE end nodes and the proposed MIG.

the separator field "`|`," each one in turn composed by the identifier of the target (either IP or MAC address), separated from the payload by a separator field "`@`." At the opposite, in the case of data to be transmitted by the LoRaWAN interface, the resulting data packet will be a sequence of processed packets separated by the separator field "`|`."

In order to better highlight the relation between *on-field* end nodes and the proposed IoT MIG, in Figure 2.4 the data flows inside the proposed architecture are shown. The introduction of new communication interfaces is possible thanks to modular architecture of the MIG. In fact, only the specific software handlers needed for a new communication interface should be written, abiding by their own constraints and rules, while MQTT broker and SDB would remain unchanged.

Considering the operational aspects, it is important to acknowledge that the interaction among MQTT topics involves some processing time. This implies that the proposed MIG is most suitable for non-real-time applications. In such scenarios, data can be collected from various sources, such as environmental monitoring or non-critical Industrial IoT (IIoT) contexts, within a reasonable time-frame, typically on the order of at least 1 s. However, it is worth noting that the system's performance could be enhanced by reducing the internal processing time by using different queuing approaches of protocols.

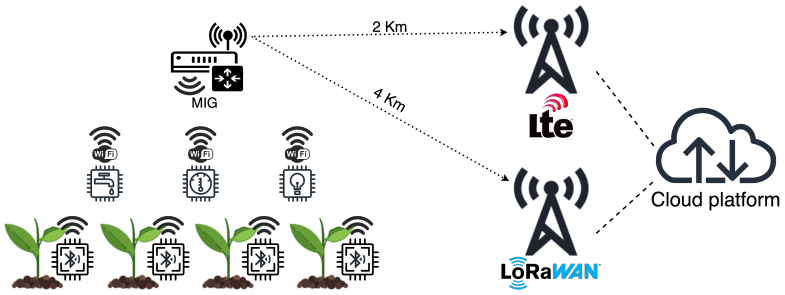


Figure 2.5: Smart farming scenario with RPi-based MIG and several Wi-Fi and BLE sensing nodes.

2.2.1 IoT-oriented Use Cases

The proposed MIG is applicable to several heterogeneous IoT scenarios where data may be collected from different data sources deployed in the environment of interest.

- A *first* representative application scenario is *smart farming*, in which several IoT sensing/actuating nodes (e.g., based on ESP32 SoC [90] and equipped with Wi-Fi and BLE connectivity, as well as several hardware sensors, such as DHT11 [91]) with short-range communication capabilities are deployed over a large area far away from an Internet access point. The collected data need to reach high-layer processing entities (e.g., cloud platforms, as well as end users, such as farmers) interested on these data, following a *Farm-as-a-Service* (FaaS) approach [92], as shown in Figure 2.5.

- A *second* scenario benefiting from the adoption of the proposed IoT-oriented MIG is related to the applications in the aerial IoT field, namely *UAV-based remote monitoring*. As an example, in a smart city a large number of short-range WSNs might be deployed to collect data of interest, possibly pre-processing them before forwarding them to high-layer consumers. Then, a UAV equipped with a Wi-Fi-, BLE- and LoRaWAN-enabled MIG can, *first*, gather data (either using short-range or long-range communication protocols) by flying over/near these WSNs and, *then*, forward the collected data to an Internet-connected node

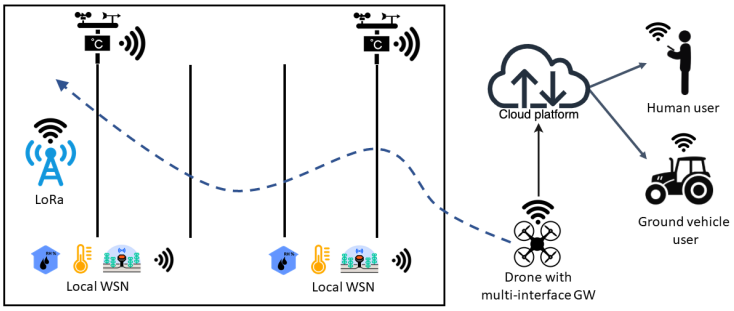


Figure 2.6: UAV environment monitoring applications involving the proposed MIG.

using a long-range or cellular communication protocol, as shown in Figure 2.6.

2.3 Analytical Queuing Model

In order to investigate how the proposed MIG harnesses communication heterogeneity by properly handling traffic data, in this Section a novel Markov chain-based queuing model is derived. In Subsection 2.3.1, the MIG is modeled through an embedded Markov chain with states corresponding to the communication interfaces: the chain is in one state if the corresponding interface is transmitting or receiving. The Markov chain transition matrix is associated with input and output flows across different interfaces. The derived model does not take into account the physical transmission channels associated with the communication interfaces equipping the MIG, since it is focused only on the internal information flow management to predict the performance of the MIG, taking also into account possible limitations of a real system deployment.

2.3.1 Embedded Markov Chain

The flows of the DTUs inside the MIG can be characterized through an embedded Markov chain with states corresponding to the MIG's communication interfaces. The transition probability associated with a link between two states depends, in general,

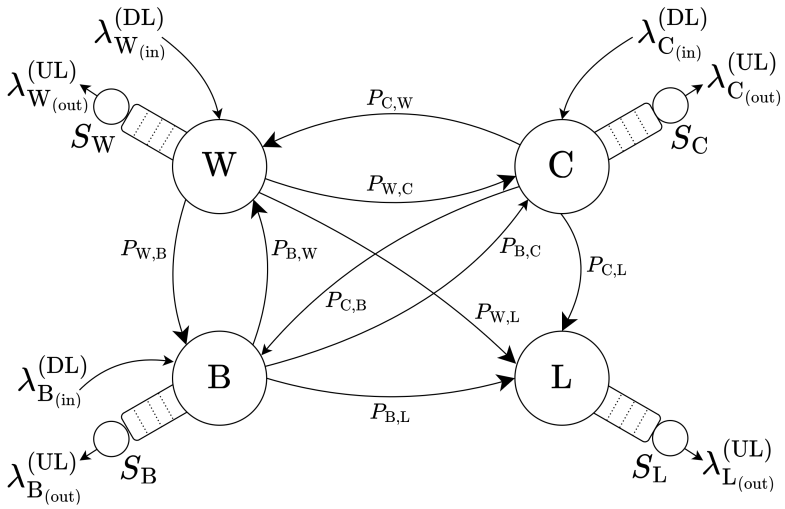


Figure 2.7: State transition diagram of the proposed IoT MIG.

on the flow from the input (DL) interface (initial state) to the output (UL) interface (final state). A high-level overview of the state transition diagram is shown in Figure 2.7, where $\lambda^{(DL)}_{W_{(in)}}$, $\lambda^{(DL)}_{C_{(in)}}$ and $\lambda^{(DL)}_{B_{(in)}}$ represent the input arrival rates (dimension: [DTU/s]) from the Wi-Fi, cellular and BLE interfaces, respectively, while $\lambda^{(UL)}_{W_{(out)}}$, $\lambda^{(UL)}_{C_{(out)}}$, $\lambda^{(UL)}_{L_{(out)}}$ and $\lambda^{(UL)}_{B_{(out)}}$ represent the departure rates from Wi-Fi, cellular (dimension: [DTU/s] for both), LoRaWAN and BLE (dimension: [pkt/s] for both) interfaces, respectively.[2] Moreover, S_W, S_B, S_C and S_L represent the service times (dimension: [s]) of the servers associated with the corresponding interfaces, respectively.

The transition matrix of the Markov chain shown in Figure 2.7 can be expressed as follows:

$$\mathbb{P} = [P_{i,j}] = \begin{pmatrix} P_{W,W} & P_{W,C} & P_{W,B} & P_{W,L} \\ P_{C,W} & P_{C,C} & P_{C,B} & P_{C,L} \\ P_{B,W} & P_{B,C} & P_{B,B} & P_{B,L} \end{pmatrix} \tag{2.1}$$

[2]It must be remarked that no arrival flow is considered in the LoRaWAN state (i.e., $\lambda^{(DL)}_{L_{(in)}} = 0$ pkt/s), as the LoRaWAN interface is assumed to support only UL communications (no Class C IoT node is considered).

where $P_{i,j}$, $i \in \{\mathrm{W},\mathrm{C},\mathrm{B},\mathrm{L}\}$, $j \in \{\mathrm{W},\mathrm{B},\mathrm{C}\}$ represents the transition probability from state i to state j or, in other words, the probability that an information flow has to be transferred from the i-th interface (receiving interface) to the j-th interface (transmitting interface).

Since the Markov chain-based queuing model relies on transition probabilities determined by (i) the internal routing rules within the MIG and (ii) the input arrival rates at the MIG's communication interfaces (DL flows), it is possible to express the corresponding arrival rates at the output queues of the communication interfaces (UL flows) as follows:

$$\begin{aligned}
\lambda^{(\mathrm{UL})}_{\mathrm{L}_{(\mathrm{in})}} &= \lambda^{(\mathrm{DL})}_{\mathrm{B}_{(\mathrm{in})}} \cdot P_{\mathrm{B,L}} + \lambda^{(\mathrm{DL})}_{\mathrm{C}_{(\mathrm{in})}} \cdot P_{\mathrm{C,L}} + \lambda^{(\mathrm{DL})}_{\mathrm{W}_{(\mathrm{in})}} \cdot P_{\mathrm{W,L}} \\
\lambda^{(\mathrm{UL})}_{\mathrm{B}_{(\mathrm{in})}} &= \lambda^{(\mathrm{DL})}_{\mathrm{C}_{(\mathrm{in})}} \cdot P_{\mathrm{C,B}} + \lambda^{(\mathrm{DL})}_{\mathrm{W}_{(\mathrm{in})}} \cdot P_{\mathrm{W,B}} \\
\lambda^{(\mathrm{UL})}_{\mathrm{C}_{(\mathrm{in})}} &= \lambda^{(\mathrm{DL})}_{\mathrm{B}_{(\mathrm{in})}} \cdot P_{\mathrm{B,C}} + \lambda^{(\mathrm{DL})}_{\mathrm{W}_{(\mathrm{in})}} \cdot P_{\mathrm{W,C}} \\
\lambda^{(\mathrm{UL})}_{\mathrm{W}_{(\mathrm{in})}} &= \lambda^{(\mathrm{DL})}_{\mathrm{C}_{(\mathrm{in})}} \cdot P_{\mathrm{C,W}} + \lambda^{(\mathrm{DL})}_{\mathrm{B}_{(\mathrm{in})}} \cdot P_{\mathrm{B,W}} \,.
\end{aligned} \tag{2.2}$$

It must be specified that although an output queue exists at each MIG interface for transmissions outside of the MIG (i.e., UL transmissions), no queues are associated with the links between pairs of states on the state diagram (i.e., between pairs of MIG interfaces). This is because: (i) packets received from end nodes are immediately processed, eliminating the need for input queues at the communication interfaces; and (ii) internal transitions are managed at software level, and the associated latencies are negligible within the Markov chain-based model. Therefore, the model focus on the output queues associated with the MIG interfaces.

Finally, it is assumed that the internal routing between the different MIG interfaces considers direct information flows from one interface to another interface (e.g., an information flow entering from the BLE interface is forwarded to the LoRaWAN interface). This *1-in-to-1-out* assumption on internal routing is meaningful for the following reasons: (i) it reflects a realistic behavior of the MIG for IoT applications, as discussed in Section 2.2; (ii) it keeps the internal Markov chain-based

model tractable. Taking into account Eq. (2.2), the *1-in-to-1-out* assumption can be formalized with the following constraints:

$$\begin{aligned} & P_{\mathrm{B,L}}, P_{\mathrm{C,L}}, P_{\mathrm{W,L}}, P_{\mathrm{C,B}}, P_{\mathrm{W,B}}, \\ & \quad P_{\mathrm{B,C}}, P_{\mathrm{W,C}}, P_{\mathrm{C,W}}, P_{\mathrm{B,W}} \in \{0,1\} \\ & \begin{cases} P_{\mathrm{B,L}} + P_{\mathrm{C,L}} + P_{\mathrm{W,L}} = 1 \\ P_{\mathrm{C,B}} + P_{\mathrm{W,B}} = 1 \\ P_{\mathrm{B,C}} + P_{\mathrm{W,C}} = 1 \\ P_{\mathrm{C,W}} + P_{\mathrm{B,W}} = 1\,. \end{cases} \end{aligned} \tag{2.3}$$

For example, assuming that an information flow from the BLE interface has to be forwarded to the LoRaWAN interface, in Eq. (2.2) one should set $P_{\mathrm{B,L}} = 1$, $P_{\mathrm{C,L}} = 0$, and $P_{\mathrm{W,L}} = 0$, obtaining:

$$\begin{aligned} \lambda^{(\mathrm{UL})}_{\mathrm{L}_{(\mathrm{in})}} &= \lambda^{(\mathrm{DL})}_{\mathrm{B}_{(\mathrm{in})}} \cdot P_{\mathrm{B,L}} + \lambda^{(\mathrm{DL})}_{\mathrm{C}_{(\mathrm{in})}} \cdot P_{\mathrm{C,L}} + \lambda^{(\mathrm{DL})}_{\mathrm{W}_{(\mathrm{in})}} \cdot P_{\mathrm{W,L}} \\ &= \lambda^{(\mathrm{DL})}_{\mathrm{B}_{(\mathrm{in})}}\,. \end{aligned} \tag{2.4}$$

2.3.2 G/G/1 Queues

In the proposed Markov chain model, each interface UL queue (outgoing traffic) is associated with a G/G/1 queue. This analytical queuing model aligns with the architectural description of the MQTT-based system outlined in Section 2.2. The G/G/1 queuing model has been chosen because it accommodates a wide range of distributions for both arrival processes and service times. In fact, in the proposed MIG, for each communication interface: (i) the inter-arrival time of DTUs has a general distribution with known parameters; and (ii) the service time distribution which depends on parameters related to the size of the packet being processed.

In order to accurately model the behavior of the G/G/1 queue at each UL interface, two remarks are expedient: (i) Wi-Fi and cellular G/G/1 queues transmit each DTU without performing any batch operation (on groups of DTUs), whereas (ii) BLE and LoRaWAN G/G/1 queues perform DTUs batching to optimize the throughput. In other words, in BLE and LoRaWAN cases the output packet size is maximized by

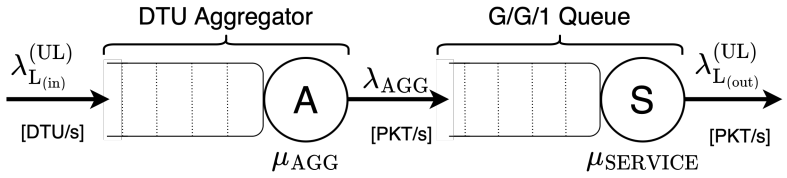


Figure 2.8: G/G/1 LoRaWAN queue composed by a DTU Aggregator (G/D/1 queue) and Packet Transmitter (G/G/1 queue).

concatenating together, in a single payload, as many DTUs as allowed by the standards, taking into account the operational settings.[3]

LoRaWAN G/G/1 Queue

LoRaWAN is the most constrained communication interface in the MIG. Its corresponding G/G/1 queue, shown in Figure 2.8, can be decomposed into the concatenation of two sub-queues: (a) a DTU Aggregator, receiving DTUs and batching them together in order to create a single LoRaWAN packet, and (b) the LoRaWAN's Packet Transmitter, in charge of processing the packets and transmitting them. In the following, we characterize these two sub-queues.

DTU Aggregator The DTU Aggregator can be modeled as a G/D/1 queue, where $\lambda^{(UL)}_{L_{(in)}}$ is the input arrival rate (dimension: [DTU/s]) of the single DTUs and $\overline{T}_{AGG}$ is the average service time (dimension: [s]) needed to aggregate a packet. The G/D/1 queue model has been chosen since the DTU arrival distribution can be any generic distribution, while the service time is deterministic, as it depends on the number of DTUs flowing into the DTU Aggregator. The DTU Aggregator's behavior strikes a balance between maximizing LoRaWAN packet length, and consequently, LoRaWAN throughput, while minimizing the DTUs waiting time within the DTU Aggregator's buffer. In particular, a maximum waiting time, defined as t_{max} (dimension: [s]), is

[3] As an example, in the case of LoRaWAN, the average number of DTUs in a single packet depends on the specific Spreading Factor (SF) chosen for the UL transmission [93].

introduced: after $t_{\max}$, even if the number of DTUs in the buffer is smaller than the maximum (denoted as n) allowed in a single LoRaWAN packet payload, the DTUs are aggregated and, *then*, sent to the Packet Transmitter. As a consequence, this approach allows DTUs inside the DTU Aggregator to incur a limited waiting time, as a trade-off between aggregated packets with small payloads (low throughput and short waiting time) and with large payloads (high throughput and long waiting time).

On the basis of the above assumptions, multiple DTUs will be aggregated together, up to a maximum of n DTUs, *if and only if* the inter-arrival times between consecutive DTUs is shorter than $t_{\max}$. Otherwise, the "incomplete" packet will be sent to the next sub-queue in Figure 2.8 (i.e., the Packet Transmitter) *as-is*. For the sake of clarity, the flow diagram detailing the behavior of the DTU Aggregator is shown in Figure 2.9 and the meanings of the indicated parameters are the following: $\lambda_{\mathrm{L}_{(\mathrm{in})}}^{(\mathrm{UL})}$ represents the average arrival rate (dimension: [DTU/s]) of the DTUs and, since the DTU Aggregator's model is based on a G/D/1 queue, the arrival rate $\lambda_{\mathrm{L}_{(\mathrm{in})}}^{(\mathrm{UL})}$ can be derived according to Eq. (2.4) (i.e., based on the *1-in-to-1-out* information flow assumption). Moreover, it is possible to express the arrival rate as a function of the average inter-arrival time as follows:

$$\lambda_{\mathrm{L}_{(\mathrm{in})}}^{(\mathrm{UL})} = \lambda_{\mathrm{B}_{(\mathrm{in})}}^{(\mathrm{DL})} = \frac{1}{\overline{T}} \tag{2.5}$$

where $\overline{T}$ is the average inter-arrival time (dimension: [s]) between consecutive DTUs and depends on the distribution of the DTU arrival process.

Considering (i) the average inter-arrival time $\overline{T}$ between DTUs, (ii) the average DTU size $\overline{L}_{\mathrm{DTU}}$, (iii) the threshold value of the waiting time $t_{\max}$, and (iv) the DTU Aggregator flow diagram shown in Figure 2.9, it is possible to evaluate the average LoRaWAN packet aggregation time and the average arrival rate λ_{AGG} (dimension: [pkt/s]) at the input of the LoRaWAN Packet Transmitter. By using the total probability theorem one can write:

$$\mathbb{E}[T_{\mathrm{AGG}}] = \sum_{i=1}^{n} \mathbb{E}[T_{\mathrm{AGG}} | \mathscr{A}_i] \cdot P(\mathscr{A}_i) \tag{2.6}$$

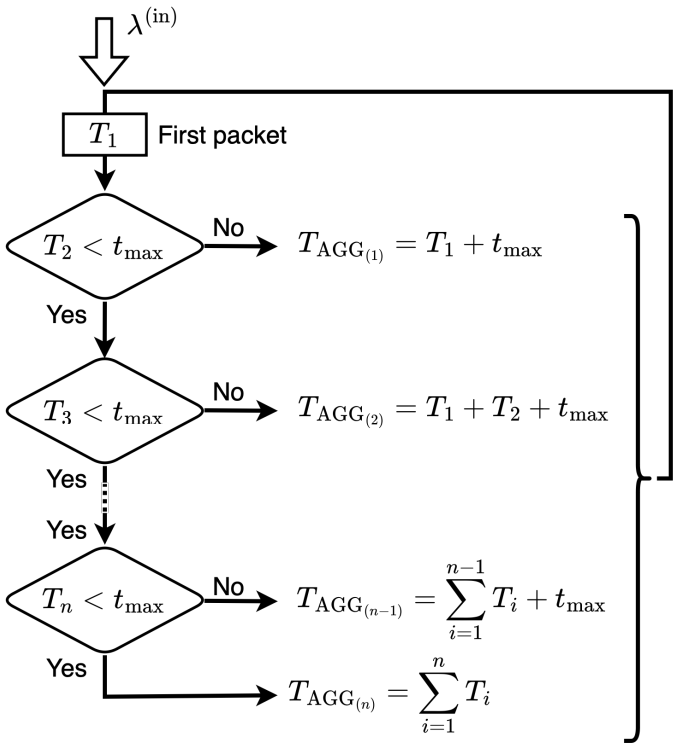


Figure 2.9: Flow diagram of the DTU Aggregator module.

where:

$$\begin{aligned} \mathscr{A}_1 &\triangleq \left\{T_2 > t_{\max}\right\} \\ \mathscr{A}_i &\triangleq \left\{T_2 < t_{\max}, \ldots, T_i < t_{\max}, T_{i+1} > t_{\max}\right\}, \\ &\qquad i = 2, \ldots, n-1 \\ \mathscr{A}_n &\triangleq \left\{T_2 < t_{\max}, \ldots, T_{n-1} < t_{\max}, T_n < t_{\max}\right\} \end{aligned} \tag{2.7}$$

and

$$\mathbb{E}[T_{\text{AGG}} | \mathscr{A}_i] = \mathbb{E}[T_{\text{AGG}_{(i)}}] \tag{2.8}$$

where:

$$T_{\text{AGG}_{(i)}} \triangleq \begin{cases} \sum_{j=1}^{i} T_j + t_{\max} & 1 \leq i < n \\ \sum_{j=1}^{n} T_j & i = n. \end{cases} \tag{2.9}$$

Since $\{T_i\}$ are independent and identically distributed, defining $P_{\max} \triangleq P\{T_i > t_{\max}\}$ one can write:

$$P(\mathscr{A}_i) = \begin{cases} (1-P_{\max})^{i-1} \cdot P_{\max} & i = 1,\ldots,n-1 \\ (1-P_{\max})^{n-1} & i = n. \end{cases} \tag{2.10}$$

From Eq. (2.6), one obtains:

$$\begin{aligned} \overline{T}_{\mathrm{AGG}} = & \sum_{i=1}^{n-1} \left(\sum_{j=1}^{i} \overline{T}_j + t_{\max} \right) \cdot (1-P_{\max})^{i-1} \cdot P_{\max} \\ & + \left(\sum_{i=1}^{n} \overline{T}_i \right) \cdot (1-P_{\max})^{n-1} . \end{aligned} \tag{2.11}$$

Finally, observing that $\overline{T}_i = \overline{T}$, $\forall i \in \{1,\ldots,n\}$, it follows:

$$\begin{aligned} \overline{T}_{\mathrm{AGG}} = & \sum_{i=1}^{n-1} \left(i \cdot \overline{T} + t_{\max} \right) \cdot (1-P_{\max})^{i-1} \cdot P_{\max} \\ & + n \cdot \overline{T} \cdot (1-P_{\max})^{n-1} . \end{aligned} \tag{2.12}$$

Similarly, one can obtain the average quadratic value of T_{AGG} as follows:

$$\begin{aligned} \mathbb{E}[T_{\mathrm{AGG}}^2] &= \sum_{i=1}^{n} \mathbb{E}\left[T_{\mathrm{AGG}}^2 | \mathscr{A}_i \right] \cdot P(\mathscr{A}_i) \\ &= \sum_{i=1}^{n-1} \mathbb{E}\left[\left(\sum_{j=1}^{i} T_i + t_{\max} \right)^2 \right] \\ &\quad \cdot (1-P_{\max})^{i-1} \cdot P_{\max} \\ &\quad + \mathbb{E}\left[\left(\sum_{j=1}^{n} T_j \right)^2 \right] \cdot (1-P_{\max})^{n-1} . \end{aligned} \tag{2.13}$$

Packet Transmitter In order to evaluate the waiting time in the buffer of the LoRaWAN Packet Transmitter, modeled as a G/G/1 queue, the Kingman's formula [94] can be adopted:

$$\overline{W}_{\mathrm{q}} = \left(\frac{\rho}{1-\rho} \right) \left(\frac{C_a^2 + C_s^2}{2} \right) \overline{S}_{\mathrm{L}} \tag{2.14}$$

where: $\rho \triangleq \lambda_{\text{AGG}} \cdot \overline{S}_{\text{L}}$ (adimensional); $\overline{S}_{\text{L}}$ is the average service time of the LoRaWAN server (dimension: [s]); and C_a and C_s are the coefficients of variation of arrival and service times (adimensional), respectively.

The coefficient of variation of arrival times C_a can be expressed as:

$$C_a = \frac{\sigma_{T_{\text{AGG}}}}{\frac{1}{\lambda_{\text{AGG}}}} = \sigma_{T_{\text{AGG}}} \cdot \lambda_{\text{AGG}} \tag{2.15}$$

where

$$\lambda_{\text{AGG}} = \frac{1}{\overline{T}_{\text{AGG}}} \tag{2.16}$$

and

$$\sigma^2_{T_{\text{AGG}}} = \mathbb{E}[T^2_{\text{AGG}}] - \mathbb{E}[T_{\text{AGG}}]^2 . \tag{2.17}$$

To calculate the coefficient of variation of the service time, denoted as C_s, an additional analysis on the behavior and policy rules of the LoRaWAN protocol is needed. *First*, the service times' distribution has to be related to the LoRaWAN packet size, thus associating each packet composed of a specific number of aggregated DTUs with a proper probability, which depends on the parameters of the DTUs' source generating distribution. Assuming an average DTU size $\overline{L}_{\text{DTU}} = 20$ bytes and considering a maximum LoRaWAN useful achievable payload size equal to 222 bytes, the maximum number of DTUs that can be inserted into a single LoRaWAN packet is equal to 11, which correspond to the specific value of the parameter n in the previous derivation (e.g., in Figure 2.9).

Then, taking into account the DTU size and the LoRaWAN constraints, it is possible to evaluate the service time of the LoRaWAN G/G/1 server. The service time of a packet containing $i \in \{1, \ldots, n\}$ aggregated DTUs, denoted as S_{L_i}, can be expressed as follows:

$$S_{\text{L}_i} = T_{\text{P}_{\text{PROC}-i}} + T_{\text{P}_{\text{AIR}}} + T_{\text{P}_{\text{DUTYCYCLE}}} \tag{2.18}$$

where: $T_{\text{P}_{\text{PROC}-i}}$ is the internal processing time (dimension: [s]) associated with i DTUs and can be expressed as

$$T_{\text{P}_{\text{PROC}-i}} = i\,\overline{T}_{\text{DTU}_{\text{PROC}}} ; \tag{2.19}$$

$T_{\mathrm{P_{AIR}}}$ is the packet airtime (dimension: [s]) and can be expressed as [95]

$$T_{\mathrm{P_{AIR}}} = (n_{\mathrm{PREAMBLE}} + 4.25)\,\frac{2^{SF}}{BW} + \left[8 + \max\left(\left\lceil \frac{8PL - 4SF + 28 + 16 - 20H}{4\,(SF - 2DE)} \right\rceil (CR+4)\,, 0\right)\right] \frac{2^{SF}}{BW}\,; \quad (2.20)$$

and $T_{\mathrm{P_{DUTYCYCLE}}}$ is the LoRaWAN duty cycle time (dimension: [s]) and can be expressed, according to the regional parameters [93], as

$$T_{\mathrm{P_{DUTYCYCLE}}} = 0.99 \cdot T_{\mathrm{P_{AIR}}}\,. \quad (2.21)$$

In the formulas above: $\overline{T}_{\mathrm{DTU_{PROC}}}$ depends on the specific platform which the MIG is deployed; the LoRaWAN preamble size, denoted as n_{PREAMBLE}, is set to 8 byte; $SF = 7$, $BW = 125$, $DE = 0$ (low data rate optimization), $CR = 4$ (coding rate); and PL (LoRa packet payload), which includes a 13 byte LoRaWAN packet header and the aggregated DTUs, can then be expressed (in the model) as $PL = 13 + n \cdot \overline{L}_{\mathrm{DTU}}$. These LoRaWAN-related parameters have been set according to the LoRaWAN regional parameters [93].

According to Eq. (2.18), the LoRaWAN packet service times $\{S_{\mathrm{L}_i}\}_{i=1}^{n}$, with $n = 11$. In particular, the service time ranges from $S_{\mathrm{L}_1} \cong 7.2$ s (packet with 1 DTU) up to $S_{\mathrm{L}_n} \cong 37$ s (packet with $n = 11$ DTUs).

The average service time, denoted as $\overline{S}_{\mathrm{L}}$, can be calculated by applying the total probability theorem on the partition $\{\mathscr{A}_i\}_{i=1}^{n}$ in Eq. (2.7) and the service time defined by Eq. (2.18), thus obtaining:

$$\begin{aligned} \overline{S}_{\mathrm{L}} &= \sum_{i=1}^{n} \mathbb{E}[S_{\mathrm{L}} | \mathscr{A}_i] \cdot P(\mathscr{A}_i) \\ &\cong \sum_{i=1}^{n} S_{\mathrm{L}_i} \cdot P(\mathscr{A}_i) \\ &= \sum_{i=1}^{n-1} S_{\mathrm{L}_i} (1 - P_{\max})^{i-1} \cdot P_{\max} + S_{\mathrm{L}_n} (1 - P_{\max})^{n-1}\,. \end{aligned} \quad (2.22)$$

Similarly, one can write

$$\begin{aligned}\mathbb{E}[S_{\mathrm{L}}^2] &= \sum_{i=1}^{n} \mathbb{E}[S_{\mathrm{L}}^2|\mathscr{A}_i] \cdot P(\mathscr{A}_i) \\ &= \sum_{i=1}^{n} \mathbb{E}\left[\left(\sum_{i=1}^{n-1} S_{\mathrm{L}_i}\right)^2\right] \\ &\quad \cdot (1-P_{\max})^{i-1} \cdot P_{\max} \\ &\quad + \mathbb{E}\left[S_{\mathrm{L}_n}^2\right] \cdot (1-P_{\max})^{n-1}\,. \end{aligned} \tag{2.23}$$

At this point, the variance of the service time, denoted as $\sigma_{S_{\mathrm{L}}}^2$, can be calculated as follows:

$$\sigma_{S_{\mathrm{L}}}^2 = \mathbb{E}[S_{\mathrm{L}}^2] - \mathbb{E}[S_{\mathrm{L}}]^2\,. \tag{2.24}$$

The coefficient of variation of the service time C_s can thus be expressed as

$$C_s = \frac{\sigma_{S_{\mathrm{L}}}}{\overline{S}_{\mathrm{L}}}\,. \tag{2.25}$$

At this point, it is possible to evaluate the average waiting time in the buffer, denoted as $\overline{W}_{\mathrm{q}}^{(\mathrm{L})}$, according to Eq. (2.14).

Finally, knowing $\overline{W}_{\mathrm{q}}^{(\mathrm{L})}$ (Eq. (2.14)), $\overline{S}_{\mathrm{L}}$ (Eq. (2.22)), and $\overline{T}_{\mathrm{AGG}}$ (Eq. (2.12)), it is possible to calculate the overall average time (denoted as $\overline{T}_{\mathrm{PKT-SOJ_L}}$) that each DTU is expected to spend at the LoRaWAN communication interface (namely, DTU Aggregator and Packet Transmitter), from the time instant of DTU arrival to the time instant of packet (in which the DTU has been aggregated) departure, as follows:

$$\overline{T}_{\mathrm{PKT-SOJ_L}} = \overline{W}_{\mathrm{q}}^{(\mathrm{L})} + \overline{S}_{\mathrm{L}} + \overline{T}_{\mathrm{AGG}}^{(\mathrm{L})}\,. \tag{2.26}$$

This is an essential parameter in order to estimate and evaluate the overall performance of the LoRaWAN model representing the relative interface handler of the proposed IoT-oriented MIG.

BLE G/G/1 Queue

Focusing on the BLE interface, its G/G/1 queue model is similar to the one detailed in Subsection 2.3.2. More specifically, the BLE DTU Aggregator has the same behavior

of the LoRaWAN DTU Aggregator, while a proper G/G/1 queue, associated with the BLE Packet Transmitter, must be defined according to the BLE protocol rules.

BLE packets can have a larger dimension (with maximum corresponding to 512 bytes) than LoRaWAN ones. Therefore, the BLE DTU Aggregator is required to aggregate up to $n = 25$ DTUs. Moreover, similarly to the LoRaWAN DTU Aggregator, even for the BLE DTU Aggregator the parameter $t_{\max}$ is introduced, which takes the same value as the one considered in Subsection 2.3.2. The same holds for the other parameters (e.g., $\lambda_{\mathrm{B}_{(\mathrm{in})}}^{(\mathrm{UL})}$), in order to fairly compare the performance of all communication interfaces. Hence, $\lambda_{\mathrm{B}_{(\mathrm{in})}}^{(\mathrm{UL})}$ can be calculated as in Eq. (2.5), while $\overline{T}_{\mathrm{AGG}}$ and $\sigma^2_{T_{\mathrm{AGG}}}$ can be evaluated as in Eq. (2.12) and Eq. (2.17) (relying on the state diagram in Figure 2.9).

The main difference between BLE and LoRaWAN models is related to the service time of the G/G/1 queue modeling the Packet Transmitter. In fact, with the BLE protocol no duty cycle is used, thus resulting in a significantly shorter average service time. However, the BLE protocol requires the MIG (active as *master*) to connect to a BLE *slave* device before being able to communicate with it. Therefore, the BLE model has to take into account this connection time, denoted as $T_{\mathrm{CONN_B}}$ (dimension: [s]). On the basis of the experimental investigation, $\overline{T}_{\mathrm{CONN_B}} \cong 7$ s. The BLE packet service time can thus be calculated as

$$S_{\mathrm{B}_i} = \overline{T}_{\mathrm{P_{PROC}}-i} + \overline{T}_{\mathrm{CONN_B}} \tag{2.27}$$

where $\overline{T}_{\mathrm{P_{PROC}}-i}$, defined by Eq. (2.19), is the processing time required to create a packet which aggregates $i \in \{1,\ldots,n\}$ DTUs.

Hence, once all BLE packet service times $\{S_{\mathrm{B}_i}\}_{i=1}^{n}$ are calculated (similarly to the service times $\{S_{\mathrm{L}_i}\}_{i=1}^{n}$ detailed in Subsection 2.3.2 for the G/G/1 queue of the LoRaWAN Packet Transmitter), it is possible to evaluate the average waiting time in the buffer of the G/G/1 LoRaWAN queue according to Eq. (2.14), which still holds for the BLE protocol. Finally, the overall average time spent by the aggregated packet in the BLE interface, denoted as $\overline{T}_{\mathrm{PKT-SOJ_B}}$, can be calculated as follows:

$$\overline{T}_{\mathrm{PKT-SOJ_B}} = \overline{W}_{\mathrm{q}}^{(\mathrm{B})} + \overline{S}_{\mathrm{B}} + \overline{T}_{\mathrm{AGG}}^{(\mathrm{B})}\,. \tag{2.28}$$

Wi-Fi and Cellular G/G/1 Queues

Given the similarity in performance and behavior between the Wi-Fi and cellular interfaces, they can be modeled in the same way. Since both Wi-Fi and cellular interfaces have significantly higher throughput than BLE and LoRaWAN interfaces, there is no need for a batching operation on incoming DTUs, and therefore, the DTU Aggregator is not present in their models. Consequently, both Wi-Fi and cellular interface UL queues can be simplified to G/D/1 queues, where the service time for each DTU is composed of a fixed processing time denoted as $t_{\mathrm{DTU_{PROC}}}$ and a fixed latency denoted as $t_{\mathrm{LATENCY_W}}$ and $t_{\mathrm{LATENCY_C}}$ for Wi-Fi and cellular interfaces, respectively. In other words, this can be expressed as:

$$S_{\mathrm{W}} = t_{\mathrm{DTU_{PROC}}} + t_{\mathrm{LATENCY_W}} \tag{2.29}$$

$$S_{\mathrm{C}} = t_{\mathrm{DTU_{PROC}}} + t_{\mathrm{LATENCY_C}}\,. \tag{2.30}$$

Since the service time is deterministic, the coefficient of variation of the service time C_s becomes equal to 0. Therefore, the waiting time in the buffer, given by Eq. (2.14), reduces, in the Wi-Fi and cellular cases, to

$$\overline{W}_{\mathrm{q}}^{(\mathrm{W})} = \frac{\rho_{\mathrm{W}}}{1-\rho_{\mathrm{W}}} \frac{\left(C_a^{(\mathrm{W})}\right)^2}{2} S_{\mathrm{W}} \tag{2.31}$$

$$\overline{W}_{\mathrm{q}}^{(\mathrm{C})} = \frac{\rho_{\mathrm{C}}}{1-\rho_{\mathrm{C}}} \frac{\left(C_a^{(\mathrm{C})}\right)^2}{2} S_{\mathrm{C}} \tag{2.32}$$

where: S_{W} and S_{C} are the Wi-Fi and cellular service times (Eq. (2.29) and Eq. (2.30), respectively); $\rho_{\mathrm{W}} \triangleq \lambda_{\mathrm{W_{(in)}}}^{(\mathrm{UL})} \cdot \overline{S}_{\mathrm{W}}$ and $\rho_{\mathrm{C}} \triangleq \lambda_{\mathrm{C_{(in)}}}^{(\mathrm{UL})} \cdot \overline{S}_{\mathrm{C}}$ (depending on the interface); $C_a^{(\mathrm{W})} = \sigma_{\mathrm{ARR}}^{(\mathrm{W})} \cdot \lambda_{\mathrm{W_{(in)}}}^{(\mathrm{UL})}$ and $C_a^{(\mathrm{C})} = \sigma_{\mathrm{ARR}}^{(\mathrm{C})} \cdot \lambda_{\mathrm{C_{(in)}}}^{(\mathrm{UL})}$.

Finally, it is possible to obtain the average waiting time $\overline{W}_{\mathrm{q}}$. The overall times spent by a DTU at the Wi-Fi or cellular interfaces can then be expressed as

$$\overline{T}_{\mathrm{DTU-SOJ_W}} = \overline{W}_{\mathrm{q}}^{(\mathrm{W})} + \overline{S}_{\mathrm{W}} \tag{2.33}$$

$$\overline{T}_{\mathrm{DTU-SOJ_C}} = \overline{W}_{\mathrm{q}}^{(\mathrm{C})} + \overline{S}_{\mathrm{C}} \tag{2.34}$$

where $\overline{W}_{\mathrm{q}}^{(\mathrm{W})}$ and $\overline{W}_{\mathrm{q}}^{(\mathrm{C})}$ can be computed as in Eq. (2.31) and Eq. (2.32), respectively.

As final remark, the main difference between Wi-Fi and cellular Packet Transmitter queues is that, according to experimental measurements, $t_{\mathrm{LATENCY_W}} \gg t_{\mathrm{LATENCY_C}}$. In other words, the cellular interface has a significantly longer sojourn time (due to technological reasons). This aspect, further depending on the specific cellular protocol version (e.g., 4G/5G), may introduce a relevant latency in some applications.

2.4 Simulator Performance Evaluation

A Python-based simulator, taking into account all the blocks considered in the analytical model, has been developed to evaluate the model performance. More in detail, the simulator includes a DTU Generator, which generates, according to a uniform distribution [96] $\mathscr{U}[T_{\mathrm{a}}, T_{\mathrm{b}}]$, DTUs to be processed by all the interface queues. In particular, the following reference values are initially considered: for all interfaces, $T_{\mathrm{a}} = 0$ s, $T_{\mathrm{MAX}} = 5$ s, $\overline{L}_{\mathrm{DTU}} = 20$ bytes, $\overline{T}_{\mathrm{DTU_{PROC}}} = 20$ ms; for the BLE interface, $\overline{T}_{\mathrm{CONN_B}} = 7$ s; for cellular and Wi-Fi interfaces, $t_{\mathrm{LATENCY_C}} = 50$ ms and $t_{\mathrm{LATENCY_W}} = 10$ ms, respectively.

By generating 1,000 DTUs, it is possible to characterize each network interface according to the following performance metrics: (i) server utilization ratio ρ; (ii) average service time $\overline{S}$; (iii) waiting time $\overline{W}_{\mathrm{q}}$ in the buffer of the G/G/1 and/or G/D/1 queues (depending on the presence or not of the DTU Aggregator), and (iv) sojourn time $\overline{T}_{\mathrm{DTU-SOJ}}$ (which includes, in the LoRaWAN and BLE cases, the DTU aggregation time). The selected metrics are relevant for the following reasons.

- The average service time allows to estimate the processing time required by each protocol to serve packets.
- The sojourn time is relevant to understand the overall time spent by the data in the system and, consequently, the latency introduced by the MIG in routing data between heterogeneous communication interfaces.
- The server utilization ratio is expedient to understand the load of the interface server, thus allowing to estimate if an information flow increment can be

tolerated. Moreover, the server utilization ratio might be useful for energy consumption optimization purposes (e.g., to maximize battery energy savings with a battery-powered MIG mounted on a UAV).

In order to estimate the accuracy of the simulated performance indicators, the Confidence Interval (CI) [97] of each simulation point is computed as follows:

$$\mathrm{CI}_{(95\%)} = z^* \cdot \frac{\sigma}{\sqrt{n_{\mathrm{sim}}}} \tag{2.35}$$

where: z^* is the z star parameter, set to 1.96 (as defined in [97]) to obtain a 95% confidence interval; σ^2 is the variance of the analyzed indicator, obtained from the simulator's output; n_{sim} corresponds to the population number, equal to the number of DTUs processed by the simulator, i.e., $n_{\mathrm{sim}} = 1{,}000$.

2.4.1 Servers Utilization Ratio ρ Evaluation

In order to analyze the stability conditions of the different communication protocols, the behavior of the server utilization ratio of each interface is investigated in function of T_{b}, with a DTUs' generation process defined according to a uniform distribution $\mathscr{U}[0, T_{\mathrm{b}}]$ (i.e., $T_{\mathrm{a}} = 0$). Therefore, the average inter-arrival time $\overline{T}$ can be calculated as follows:

$$\overline{T} = \frac{T_{\mathrm{b}}}{2}. \tag{2.36}$$

In Figure 2.10, analytical (an) and simulated (sim) server utilization ratios for the following interfaces are shown: (a) LoRaWAN; (b) BLE; (c) Wi-Fi; and (d) cellular. This allows to directly compare (and validate) the performance predicted by the Markov chain-based analytical model with that predicted by the implemented Python simulator. From Figure 2.10(a), it can be observed that $\rho_{\mathrm{L}}^{(\mathrm{sim})} = \rho_{\mathrm{L}}^{(\mathrm{an})} = 1$ for $T_{\mathrm{b}} \simeq 10$ s. Hence, it can be concluded that the LoRaWAN interface cannot support a DTU generation distribution $\mathscr{U}[0, T_{\mathrm{b}}]$ with $T_{\mathrm{b}} \leq T_{\mathrm{b}}^{(\mathrm{min-L})} \simeq 10$ s.

Regarding Figure 2.10(c) and Figure 2.10(d), related to Wi-Fi (ρ_{W}) and cellular (ρ_{C}) server utilization ratios, respectively, it can be noted that $\rho_{\mathrm{C}}^{(\mathrm{an})} = \rho_{\mathrm{C}}^{(\mathrm{sim})} = 1$ for $T_{\mathrm{b}} \simeq 0.124$ s, while $\rho_{\mathrm{W}}^{(\mathrm{an})} = \rho_{\mathrm{W}}^{(\mathrm{sim})} = 1$ for $T_{\mathrm{b}} \simeq 0.056$ s. A very good agreement between simulated and analytical performances can be observed.

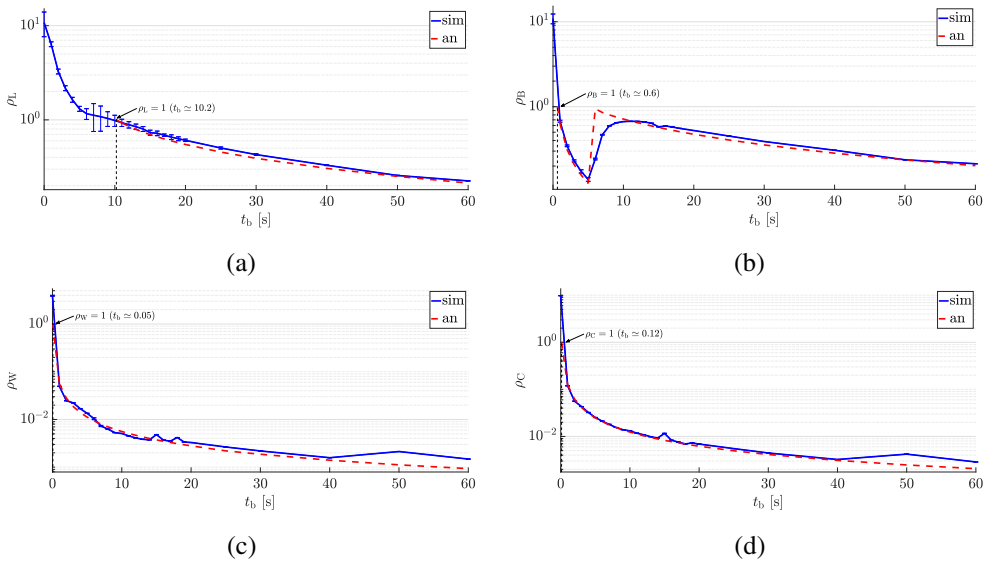


Figure 2.10: Analytical (an) and simulated (sim) interface server utilization ratio as a function of T_b (DTUs' generation distribution $\mathscr{U}[0, T_b]$), for: (a) LoRaWAN (ρ_L); (b) BLE (ρ_B); (c) Wi-Fi (ρ_W); and (d) cellular (ρ_C).

Finally, looking at Figure 2.10(b), it can be concluded that the BLE interface can properly handle incoming DTUs for $T_b > T_b^{(\min-B)} \simeq 0.6$ s. Furthermore, from the analytical results it can notice that the BLE interface reaches a peak when $T_b \simeq 5$ s. This corresponds to the value of the parameter T_{MAX}, defined in Subsection 2.3.2, that maximizes the DTU aggregation process. This analytical result is confirmed also by simulation values and is further explained through an in-deep analysis of the behavior of the DTU Aggregator carried out in Subsection 2.4.2.

2.4.2 Impact of the DTU Aggregator on ρ

With the aim to better understand the impact of the DTU Aggregator on the server utilization ratio ρ, the BLE case is investigated, i.e., the behavior of ρ shown in Figure 2.10(b). For the sake of simplicity, the inter-arrival time between DTUs is assumed exactly equal to $\overline{T} = \frac{T_b}{2}$ (deterministic). In Figure 2.11, the behavior of ρ is

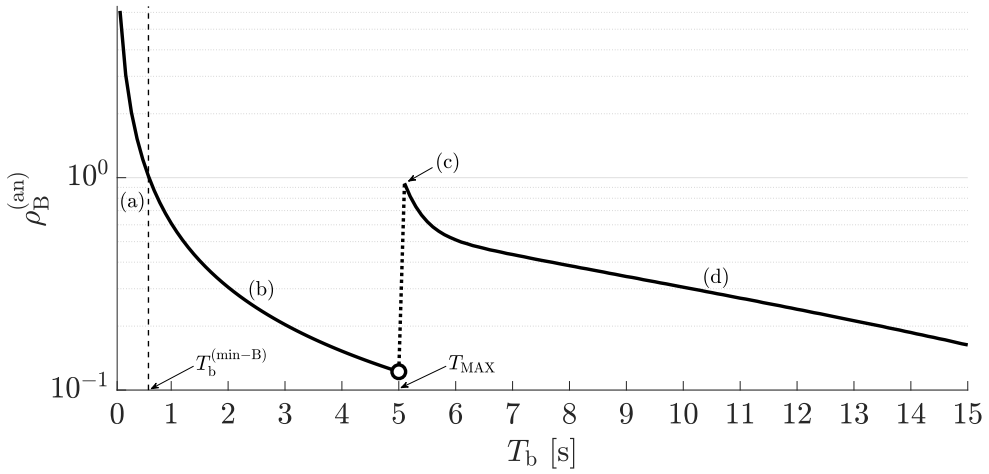


Figure 2.11: Behavior of $\rho_{\mathrm{B}}^{(\mathrm{an})}$ in the assumption of fixed DTU inter-arrival time $T = \overline{T} = \frac{T_{\mathrm{b}}}{2}$.

shown as a function of T_{b}. Therefore, it is possible to identify the following regions/-values:

- $0 < T_{\mathrm{b}} < T_{\mathrm{b}}^{(\mathrm{min-B})}$: instability region corresponding to $\rho > 1$ (interval (a) in Figure 2.11);
- $T_{\mathrm{b}}^{(\mathrm{min-B})} < T_{\mathrm{b}} < T_{\mathrm{MAX}}$: stability region, where ρ decreases until reaching a local minimum (region (b) in Figure 2.11);
- $T_{\mathrm{b}} = T_{\mathrm{MAX}}$: peak point, where ρ suddenly increases and reaches the local peak (point (c) in Figure 2.11);
- $T_{\mathrm{b}} \geq T_{\mathrm{MAX}}$: stability region, where ρ further decreases (region (d) in Figure 2.11).

It can be observed that there is a discontinuity for $T_{\mathrm{b}} = T_{\mathrm{MAX}}$.

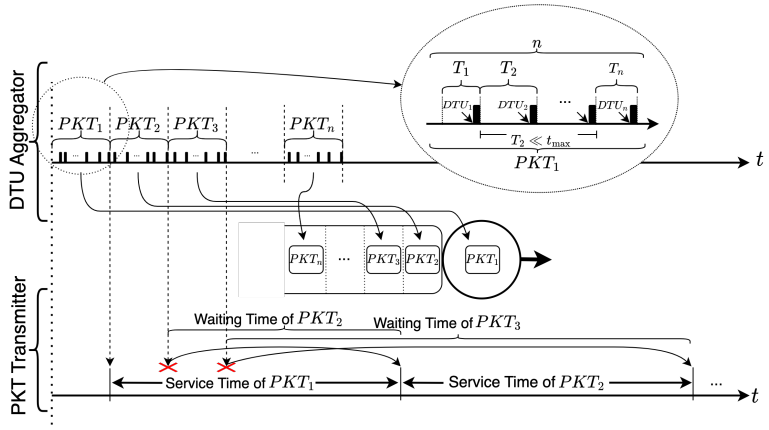


Figure 2.12: Illustrative representation of the behavior of DTU Aggregator and Packet Transmitter in the instability region, for $0 < T_{\mathrm{b}} < T_{\mathrm{b}}^{(\mathrm{min-B})}$, with $T_{\mathrm{b}}^{(\mathrm{min-B})} \simeq 0.6$.

Instability Region

For $0 < T_{\mathrm{b}} < T_{\mathrm{b}}^{(\mathrm{min-B})}$, with $T_{\mathrm{b}}^{(\mathrm{min-B})} \simeq 0.6$ s, it holds that $\rho_{\mathrm{B}}^{(\mathrm{an})} > 1$: the system is unstable, as the server cannot process all the incoming DTUs. This is due to small value of $\overline{T} = \frac{T_{\mathrm{b}}}{2}$, i.e., the average inter-arrival time of the incoming DTUs flowing into the DTU Aggregator. In this region, the DTU Aggregator can correctly process and aggregate all the incoming DTUs, thus continuously generating fully aggregated packets composed by the maximum amount of DTUs, namely n (e.g., $n = 25$ for the BLE interface). However, the server associated with the Packet Transmitter can not properly manage the incoming aggregated packets, as the average service time $\overline{S}_{\mathrm{B}}$, defined according to Eq. (2.27), is too long. Moreover, as detailed in Section 2.3, it is useful to recall that the long service time of the BLE Packet Transmitter is mainly caused by the connection time $\overline{T}_{\mathrm{CONN_B}}$, which is the main component of $\overline{S}_{\mathrm{B}}$. For the sake of clarity, this instability behavior is depicted in Figure 2.12, where it is clearly visible the arrival of multiple DTUs denoted by a short inter-arrival time $\overline{T}$ in the DTU Aggregator, and the consequent generation of multiple aggregated packets incoming in the Packet Transmitter way before it finishes to serve the first received packet, as shown in the timeline of the Packet Transmitter's service time.

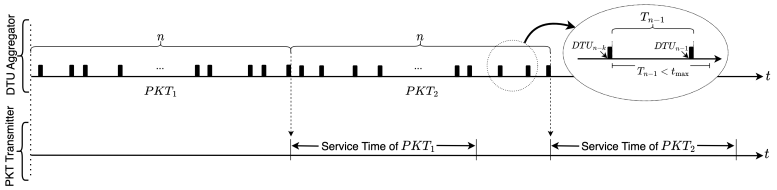


Figure 2.13: Illustrative representation of the behavior of DTU Aggregator and Packet Transmitter for $T_{\mathrm{b}}^{(\mathrm{min-B})} < T_{\mathrm{b}} < T_{\mathrm{MAX}}$

First Region of System Stability

The first stability region is defined for $T_{\mathrm{b}}^{(\mathrm{min-B})} < T_{\mathrm{b}} < T_{\mathrm{MAX}}$, where $\rho_{\mathrm{B}}^{(\mathrm{an})}$ becomes smaller than 1 and tends to decrease, reaching the local minimum for $T_{\mathrm{b}} = T_{\mathrm{MAX}}^{-}$ (with $T_{\mathrm{MAX}} = 5$ s). This behavior is due to the fact that the value of $\overline{T} = \frac{T_{\mathrm{b}}}{2}$ increases as a function of T_{b} but still maintaining its value below the T_{MAX} threshold, since $T_{\mathrm{b}} < T_{\mathrm{MAX}}$, therefore all the aggregated packets in this region still have the maximum number of DTUs n, increasing the efficiency of the system. As illustrated in Figure 2.13, in comparison with the case discussed in Subsection 2.4.2 and shown in Figure 2.12, in this case the time needed to generate a packet is slightly higher than (equal to, at the lower boundary) the service time of the Packet Transmitter, therefore the server of the Packet Transmitter can thus properly handle all the incoming packets.

Peak Point

When $T_{\mathrm{b}} = T_{\mathrm{MAX}}$, the behavior of the peak point (c) in Figure 2.11 can be explained by the fact that the analytical model computes the value of $\rho_{\mathrm{B}}^{(\mathrm{an})}$ taking as input the mean inter-arrival time between DTUs ($\overline{T}$) instead of using the deterministic values computed by the simulator, which, as shown in Figure 2.10(b), returns a more smooth $\rho_{\mathrm{B}}^{(\mathrm{sim})}$ curve in such region. This is due to the fact that when T_{b} becomes slightly bigger than T_{MAX}, the probability of a new DTU arrival with a $\overline{T} \leq T_{\mathrm{MAX}}$ is no more equal to 1, meaning that it is no more certain to receive the next DTU within T_{MAX}. This means that in the simulator, in which deterministic values of T are com-

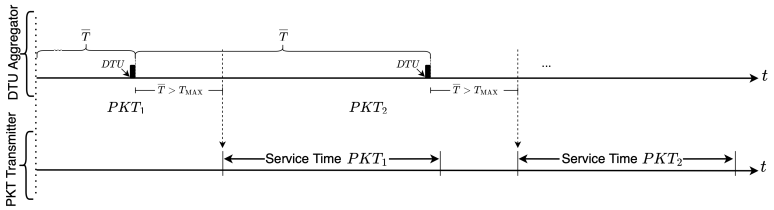


Figure 2.14: Illustrative representation of the behavior of DTU Aggregator and Packet Transmitter for $T_b > T_{MAX}$.

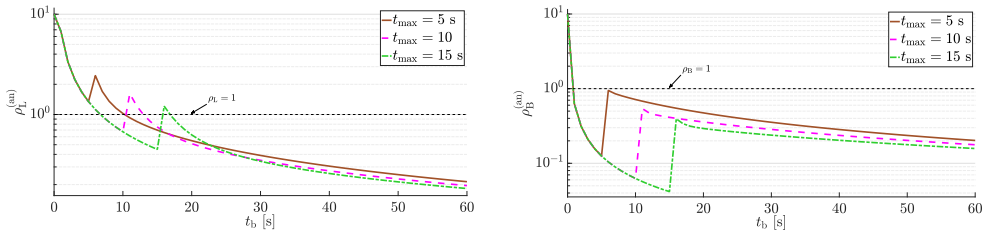
puted for each generated DTU, more packets with a number of DTUs smaller than n will be generated by the DTU Aggregator, in the end increasing the workload of the Packet Transmitter's server that has to handle, while in the analytical model, due to the mean $\overline{T}$ approximation, all the new aggregated packets will have one DTU, thus originating the peak visible as point (c) in Figure 2.11.

Second Region of System Stability

The second stability region of the system is defined when $T_b \gg T_{MAX}$, meaning that $\overline{T} > T_{MAX}$, therefore with all the new aggregated packets being composed exactly by only one DTU and with $\rho_B^{(an)}$ decreasing, as shown in Figure 2.14.

For increasing values of T_b (and, then, $\overline{T}$), $\rho_B^{(an)}$ tends to further decrease, as $\overline{T}$ is longer and longer than the service time, thus leading again to a lower server utilization.

In order to further investigate the presence of the peak in Figure 2.15, the analytical queuing server utilization ratio $\rho^{(an)}$ is evaluated as a function of the parameter T_b, associated with a uniform DTU generation distribution $\mathscr{U}[0, T_b]$, for various values of T_{MAX} (in detail, 5 s, 10 s, and 15 s), considering both LoRaWAN (Figure 2.15(a)) and BLE (Figure 2.15(b)) interfaces. The obtained results confirm how the server utilization ρ is influenced by the parameter T_{MAX} of the DTU Aggregator. This parameter affects the system efficiency by reducing, for small values of T_{MAX}, the "idle times" between aggregated packets sent to the interface server.



(a) (b)

Figure 2.15: Analytical server utilization ratio $\rho^{(an)}$ as a function of T_b (with DTUs' generation distribution $\mathscr{U}[0,T_b]$), with T_{MAX} set to 5 s, 10 s, and 15 s, for (a) LoRaWAN and (b) BLE interfaces.

2.4.3 Average Service $\overline{S}$, Sojourn Times $\overline{T}_{SOJ}$ and Evaluation

It is of interest to investigate the behavior of (i) the average service time $\overline{S}$ of each interface's server and (ii) the total average time spent by a DTU at each interface, namely: LoRaWAN ($\overline{T}_{PKT-SOJ_L}$) and BLE ($\overline{T}_{PKT-SOJ_B}$), as defined in Eq. (2.26); and Wi-Fi ($\overline{T}_{DTU-SOJ_W}$) and cellular ($\overline{T}_{DTU-SOJ_C}$), as defined in Eq. (2.33) and Eq. (2.34), respectively. The obtained results (with the corresponding confidence interval at each simulation point) are shown in Figure 2.16. For the sake of clarity, it is noteworthy to highlight that, even if these performance metrics have been studied as a function of $T_b \in [0,60]$, in Figure 2.16 the results are shown in a "restricted" version for $T_b \in [0,30]$, since for $T_b \in [30,60]$ the performance metrics experienced a flat trend and, thus, the most relevant and interesting behavior useful to validate the MIG's analytical model is for $T_b \in [0,30]$.

From the results in Figure 2.16(a), related to LoRaWAN, it can be noticed that the average service times $\overline{S}_L^{(an)}$ and $\overline{S}_L^{(sim)}$ are in very good agreement. In particular, $\overline{S}_L^{(sim)}$ reaches its saturation value when the DTU aggregation is maximized, i.e., when $T_b \leq T_{MAX}$. With regard to average analytical ($\overline{T}_{PKT-SOJ_L}^{(an)}$) and simulated ($\overline{T}_{PKT-SOJ_L}^{(sim)}$) sojourn times, given the fixed amount of DTUs processed in the simulator (namely, 1,000 as indicated at the beginning of Section 2.3), it is possible to calculate the average waiting time of a DTU even when the analytical queuing model reaches

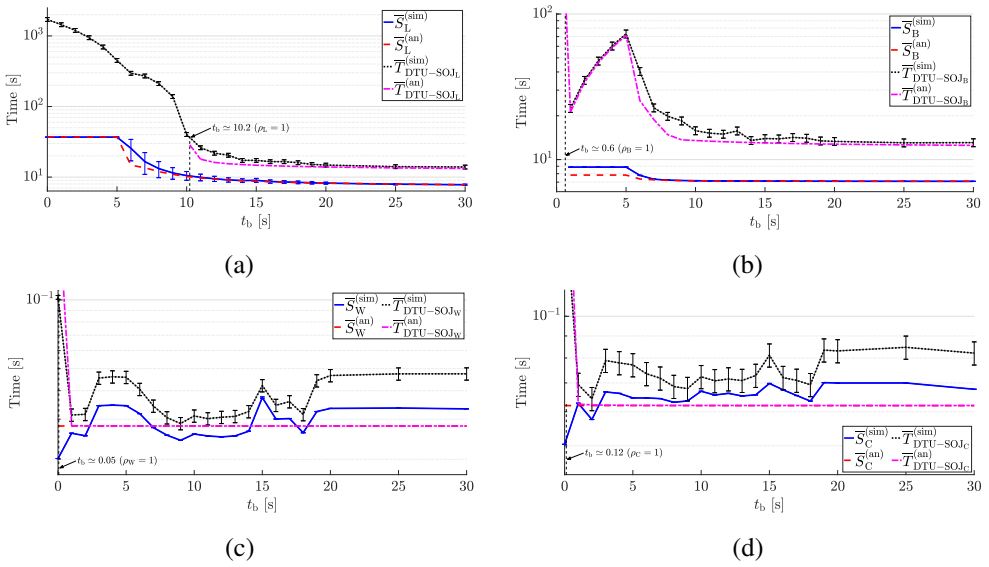


Figure 2.16: Direct comparison between analytical queuing average service/sojourn times ($\overline{S}^{(an)}/\overline{T}^{(an)}_{PKT-SOJ}$) and simulation-based average service/sojourn times ($\overline{S}^{(sim)}/\overline{T}^{(sim)}_{PKT-SOJ}$), as functions of T_b (with DTUs' generation distribution $\mathscr{U}[0,T_b]$), at the considered interfaces: (a) LoRaWAN ($\overline{S}_L$ and $\overline{T}_{PKT-SOJ_L}$), (b) BLE ($\overline{S}_B$ and $\overline{T}_{PKT-SOJ_B}$), (c) Wi-Fi ($\overline{S}_W$ and $\overline{T}_{DTU-SOJ_W}$), and (d) cellular ($\overline{S}_C$ and $\overline{T}_{DTU-SOJ_C}$).

$\rho_L = 1$.

Similar considerations can be carried out for Figure 2.16(b), referring to the BLE interface. It can be observed that both $\overline{S}^{(an)}_B$ and $\overline{S}^{(sim)}_B$ reach their maximum possible values when $T_b \leq T_{MAX}$, thus when the BLE aggregated packet size is maximized, while $\overline{T}^{(an)}_{PKT-SOJ_B}$ grows rapidly for $T_b \simeq 0.6$ s, as confirmed by $\overline{T}^{(sim)}_{DTU-SOJ_B}$. Furthermore, it can be observed the peak of both the analytical and simulator sojourn times, which occur in $T_b \simeq 5$ s, therefore when $T_b \simeq T_{MAX}$, as already detailed.

Finally, Figure 2.16(c) and Figure 2.16(d), referring to Wi-Fi and cellular interfaces' analytical and simulated service and sojourn times, respectively, confirm how analytical and simulation results are in very good agreement, given both the reduced range of the y-axis and the $\overline{T}_{DTU-SOJ}$ curves growing rapidly at $T_b \simeq 0$ s (as observed

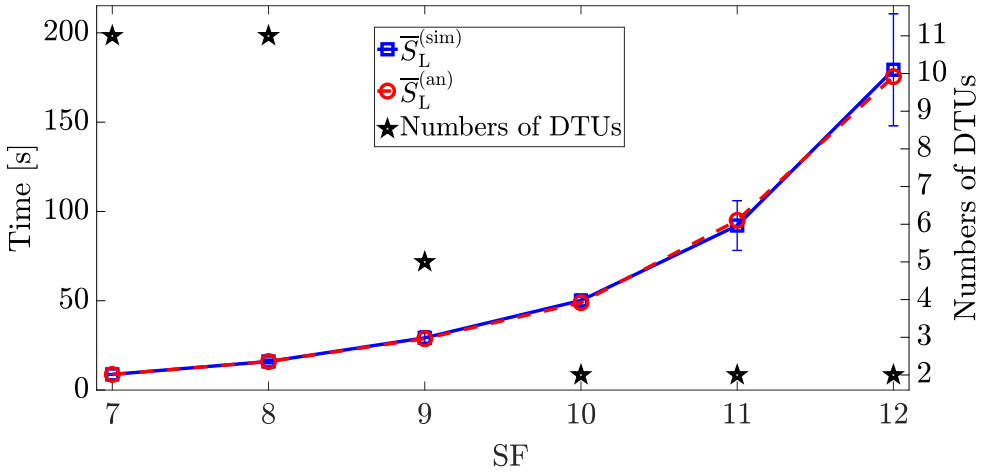


Figure 2.17: LoRaWAN analytical ($\overline{S}_{\mathrm{L}}^{(\mathrm{an})}$) and simulated ($\overline{S}_{\mathrm{L}}^{(\mathrm{sim})}$) average service time and number of DTUs per aggregated packet as a function of the SFs allowed by the LoRaWAN protocol.

in Figure 2.10(c) and Figure 2.10(d)). For the sake of completeness, the irregular behavior of the curves of both the Wi-Fi and cellular simulated service and sojourn times is due to the simulations entropy and internal processing, further accentuated by the reduced range of the *y*-axis.

2.4.4 Impact of SF on LoRaWAN DTUs' Aggregation

With regard to the LoRaWAN interface, it is of interest to investigate the average service time and the number of DTUs that can be aggregated within a single LoRaWAN packet as functions of the SF. The obtained results, both analytical and simulation-based, are shown in Figure 2.17. It can be observed that (as suggested by the LoRaWAN specifications [89]) increasing the SF (i) limits the amount of DTUs possibly being aggregated and (ii) significantly increases the average service time needed to process packets with aggregated DTUs, thus significantly increasing $\overline{S}$.

2.4.5 Final Considerations

For the sake of readability and analysis, and to ease a performance comparison between the Markov chain-based model and the implemented Python simulator, in Table 2.2 the main performance indicators investigated before, in the case of uniform distribution $\mathscr{U}[T_{\mathrm{a}}, T_{\mathrm{b}}]$ of generated DTUs, with $T_{\mathrm{a}} = 0$ s and $T_{\mathrm{b}} = 15$ s, are summarized. These results confirm that the most constrained network interface is the LoRaWAN one, followed by the BLE interface and, then, by the cellular, with the Wi-Fi interface being the best performing.

Finally, through the developed simulator it is possible to evaluate the average data rate, denoted as $\overline{\psi}$, achieved by each network interface equipping the MIG itself. More in detail, the simulated average data rates (over 1,000 generated DTUs) are:

- $\overline{\psi}_{\mathrm{L}} = 28.657$ bps for the LoRaWAN interface;
- $\overline{\psi}_{\mathrm{B}} = 46.725$ bps for the BLE interface;
- $\overline{\psi}_{\mathrm{C}} = 2{,}139.8$ bps for the cellular interface;
- $\overline{\psi}_{\mathrm{W}} = 5{,}479.32$ bps for the Wi-Fi interface.

It can be concluded that the LoRaWAN interface has a data rate $\overline{\psi}_{\mathrm{L}}$ close to the value predicted by the protocol guidelines with the same configuration (namely, 48 bps [89]). The other interfaces may be limited by the system's specific implementation, which reduces the useful payload processed by the MIG and introduces a high overhead, thus significantly lowering the achievable data rates. This is especially true for Wi-Fi and cellular network interfaces. Moreover, the cellular technology is also affected by a higher average latency introduced by the network topology. These constraints are further investigated in Section 2.5, where the experimental performance evaluation of the MIG prototype is carried out.

2.5 Experimental Evaluation and Analysis

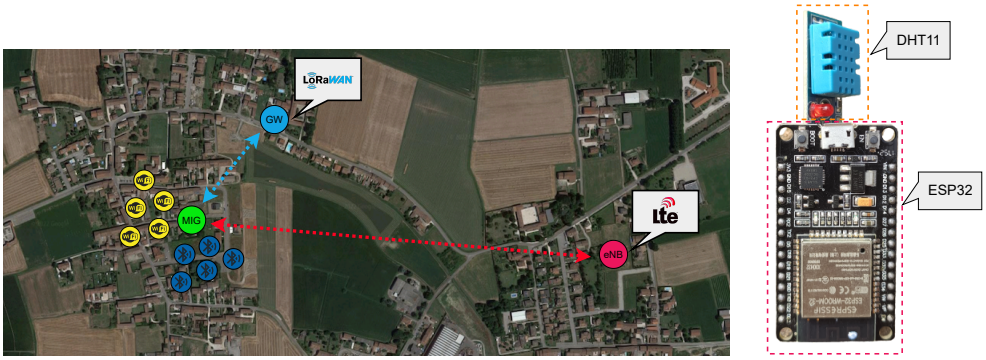
The topology of the overall experimental testbed is shown in Figure 2.18(a): the MIG (see Figure 2.1a) is connected to 10 IoT nodes (5 BLE and 5 Wi-Fi), based on the

Table 2.2: Comparison among analytical and simulation-based values with $T \sim \mathscr{U}[0,15]$.

Term	Analytical Value	Simulation Value
ρ_{W}	0.0037	0.0032
ρ_{C}	0.0083	0.0081
ρ_{B}	0.57	0.60
ρ_{L}	0.70	0.75
$\overline{T}_{\mathrm{AGG_L}}$	12.50 s	11.85 s
$\overline{T}_{\mathrm{AGG_B}}$	12.50 s	11.87 s
$\overline{W}_{\mathrm{q_L}}$	0.19 s	1.62 s
$\overline{W}_{\mathrm{q_B}}$	0.068 s	0.18 s
$\overline{W}_{\mathrm{q_W}}$	0.000058 s	0.0025 s
$\overline{W}_{\mathrm{q_C}}$	0.00029 s	0.0038 s
$\overline{S}_{\mathrm{W}}$	0.028 s	0.025 s
$\overline{S}_{\mathrm{C}}$	0.062 s	0.062 s
$\overline{S}_{\mathrm{B}}$	7.11 s	7.12 s
$\overline{S}_{\mathrm{L}}$	8.73 s	8.88 s
$\overline{T}_{\mathrm{PKT-SOJ_L}}$	21.42 s	22.35 s
$\overline{T}_{\mathrm{PKT-SOJ_B}}$	19.68 s	19.16 s
$\overline{T}_{\mathrm{DTU-SOJ_W}}$	0.028 s	0.028 s
$\overline{T}_{\mathrm{DTU-SOJ_C}}$	0.062 s	0.065 s

ESP32 SoC HW platforms and equipped with either a DHT11 humidity sensor or a temperature sensor [91]. In Figure 2.18(b), a node with a DHT11 sensor is shown. The MIG is connected to the closest cellular base station through the 4G dongle, and to a LoRaWAN GW connected to the The Things Network (TTN) [98], as depicted in Figure 2.18(a).

As highlighted in Section 2.2, the starting point of the proposed framework is a COTS device-based implementation of the MIG. While the analytical queuing model based on Markov chains and the Python-based simulator are capable of exploring



(a) (b)

Figure 2.18: (a) IoT testbed built for the experimental performance evaluation of the proposed MIG, involving: a RPi4-based MIG implementation; a cellular base station; a LoRaWAN GW; and 10 IoT nodes (5 BLE and 5 Wi-Fi). (b) IoT node based on ESP32 SoC with a DHT11 humidity and temperature sensor.

various performance metrics, the experimental testbed has limitations in this regard. Specifically, the experimental setup cannot directly measure service times, sojourn times, or server utilization ratios for different interfaces. To ensure a fair comparison between experimental results and those obtained through analysis and simulation, an evaluation was conducted to determine the time required to process a specific volume of data, namely, a defined number of DTUs at each interface. This was achieved by establishing a uniform distribution $\mathscr{U}[0,15]$ for the DTU inter-arrival time T at all interfaces. This approach enables the comparison of processing times for each interface while handling a similar quantity of DTUs.

In the following, the *experimental processing time* (denoted as $\overline{T}^{(\mathrm{exp})}_{\mathrm{PROC}}$) and the *simulator processing time* (denoted as $\overline{T}^{(\mathrm{sim})}_{\mathrm{PROC}}$) have been obtained by measuring the difference between the time instant of system initialization and the time instant corresponding to processing completion of the last DTU. The *analytical processing time* (denoted as $\overline{T}^{(\mathrm{an})}_{\mathrm{PROC}}$) for Wi-Fi and cellular interfaces has been calculated by multiplying the corresponding average sojourn time ($\overline{T}_{\mathrm{DTU-SOJ_W}}$ and $\overline{T}_{\mathrm{DTU-SOJ_C}}$, respectively) of a DTU by the number of DTUs needed to send the defined amount of

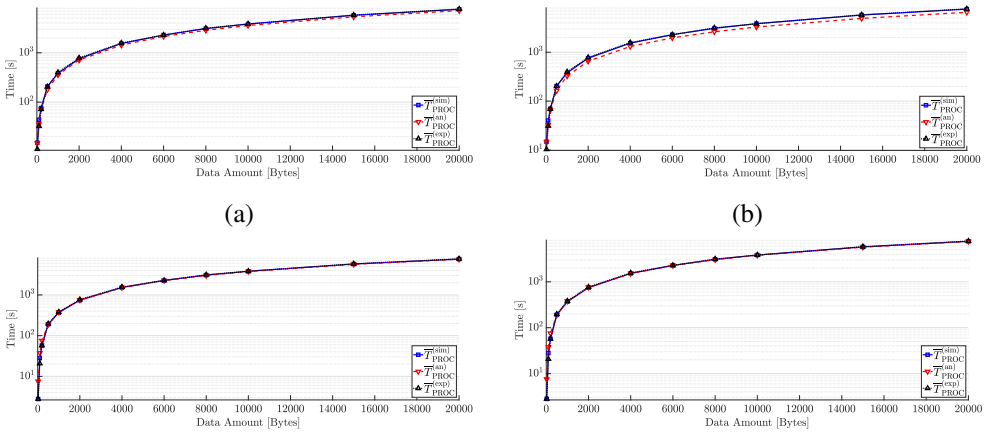


Figure 2.19: Comparison among analytical ($\overline{T}^{(\text{an})}_{\text{PROC}}$), simulated ($\overline{T}^{(\text{sim})}_{\text{PROC}}$), and experimental ($\overline{T}^{(\text{exp})}_{\text{PROC}}$) performance results obtained on the different communication interfaces equipping the MIG, namely (a) LoRaWAN, (b) BLE, (c) Wi-Fi, and (d) cellular interfaces.

information through the specific interface. In fact, this allows to obtain an acceptable estimation of the overall processing time needed to send the chosen amount of data over the designated communication interface.

The analytical, simulated, and experimental performance results for LoRaWAN, BLE, Wi-Fi, and cellular interfaces are shown in Figure 2.19(a), Figure 2.19(b), Figure 2.19(c), and Figure 2.19(d), respectively. The results depicted in Figure 2.19 reveal strikingly similar trends among the analytical queuing model, simulator, and experimental testbed. Notably, $\overline{T}^{(\text{exp})}_{\text{PROC}}$ exhibits a slightly higher value compared to both $\overline{T}^{(\text{sim})}_{\text{PROC}}$ and $\overline{T}^{(\text{an})}_{\text{PROC}}$. This discrepancy can be attributed primarily to the additional processing times introduced by the current MIG implementation, which relies on COTS components. These additional processing times may partially reduce performance relative to the predictions of both the Markov chain-based model and the Python simulator.

For LoRaWAN and BLE interfaces (whose analytical, simulated, and experimen-

tal results are shown in Figure 2.19(a) and Figure 2.19(b), respectively), $\overline{T}^{(\mathrm{an})}_{\mathrm{PROC}}$ can be expressed as follows:

$$\overline{T}^{(\mathrm{an})}_{\mathrm{PROC}} = \Upsilon^{(\mathrm{AGG})}_{\mathrm{DTU}} \cdot \Pi_{\mathrm{PKT-SOJ}} \tag{2.37}$$

where: $\Upsilon^{(\mathrm{AGG})}_{\mathrm{DTU}} \triangleq N_{\mathrm{DTU}}/\overline{n}_{\mathrm{DTU-PKT}}$ is the number of aggregated DTUs generated by the DTU Aggregator of LoRaWAN or BLE interfaces, as N_{DTU} corresponds to the amount of DTUs to be sent and $\overline{n}_{\mathrm{DTU-PKT}}$ is the average amount of DTUs inside a single aggregated LoRaWAN or BLE packet at the output of the DTU Aggregator; $\Pi_{\mathrm{PKT-SOJ}} \triangleq \overline{T}_{\mathrm{PKT-SOJ}}/2$ corresponds to the average time spent into the queuing system, where $\overline{T}_{\mathrm{PKT-SOJ}}$ represents the overall time spent by the aggregated DTUs inside LoRaWAN or BLE interfaces, respectively. In detail, $\Pi_{\mathrm{PKT-SOJ}}$ is obtained by dividing $\overline{T}_{\mathrm{PKT-SOJ}}$ by 2 to take into account the overlapping of aggregating and processing activities ot the DTU Aggregator. In fact, while a packet is aggregated by the DTU Aggregator, the Packet Transmitter is processing another packet previously aggregated

Examining the results presented in Figure 2.19(a) and Figure 2.19(b), it becomes apparent that $\overline{T}^{(\mathrm{an})}_{\mathrm{PROC}}$ (for both LoRaWAN and BLE) exhibits a slight underestimation when compared to both the values obtained through simulation and experimentation. This discrepancy may be attributed to the approximation introduced in Eq. (2.37) for $\overline{T}_{\mathrm{PKT-SOJ}}$. However, it's worth noting that the maximum approximation error is less than 10% for $\overline{T}^{(\mathrm{an})}_{\mathrm{PROC_L}}$ and less than 15% for $\overline{T}^{(\mathrm{an})}_{\mathrm{PROC_B}}$, respectively. Moreover, the relative error decreases as the quantity of aggregated DTUs increases.

Conversely, the simulated and experimental values appear to align closely, with a maximum difference of 5%. This alignment underscores the overall accuracy of the simulator when compared to the results obtained from the experimental setup.

The obtained experimental results prove that the overall performance of the proposed MIG prototype can be well estimated using both the Markov chain-based model presented in Section 2.3 and the Python-based simulator discussed in Section 2.4.

2.6 UAV-oriented MIG Implementation

In order to enable UAV-to-X connectivity through multiple scenario and use cases, a second prototypical version of the MIG, based on the architecture discussed in Section 2.2, has been developed, with the aim to provide different heterogeneous connectivity protocols for UAV applications. More in detail, the UAV-oriented MIG retains the same network interfaces (namely: LoRAWAN, Wi-Fi, BLE and 4G LTE cellular), while an additional serial communication interface has been added to enable the data exchange with the drone's FC, in particular embodied by a Pixhawk Cube Orange [11].

Therefore, an additional software entity, among those discussed and detailed in Section 2.2, has been added to handle the data exchange between the FC and the MIG, managing the serial communication through the use of the Mavlink [20] protocol.

Moreover, given the specific application of this MIG implementation, besides the MQTT-based SDB together with all the software entities managing their respective interfaces, the additional IP-level traffic is managed through the use of the Linux kernel features, such as (i) the *Dynamic Host Configuration Protocol Client Daemon (SHCPCD)* metrics, to manage the IP networks' priority between cellular and Wi-Fi interfaces, as well as (ii) the *Linux network namespaces*, to route only specific applications traffic on a well-identified IP communication interface.

More in detail, the DHCPCD is an open-source DHCP client software used on Unix-like operating systems (such as the one installed on the aerial MIG) whose primary purpose is to automatically obtain and manage network configuration information from a DHCP server. However, it supports several features, such as the possibility to set and modify a metric value for each network interface. To this end, metrics are used to prefer an interface as the default communication interface over another one, namely defining which communication interface has to be used with respect to the others, and with the interface with the lowest metric being the chosen one, then followed by the others with higher metric values [99].

Instead, the Linux namespaces represent a specific feature in the Linux kernel, in detail providing a way to isolate and partition different aspects of a Linux system,

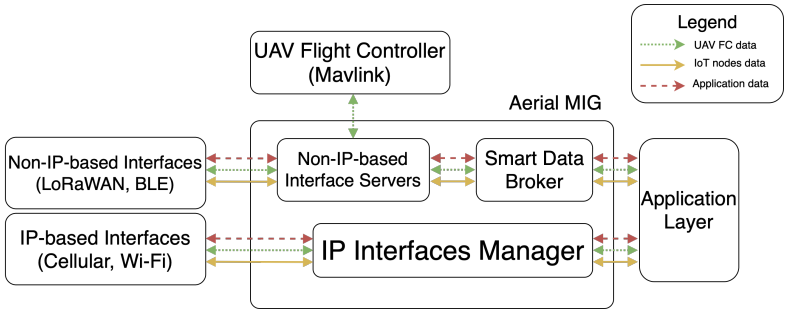


Figure 2.20: Aerial MIG architecture for UAV-to-X connectivity.

including the networking layer. Moreover, they allow to create multiple instances of certain system resources (such as network interfaces), in the way that they may appear as separate and independent to different processes running on the same host machine. More specifically, with regards to networking functionalities, Linux namespaces are used to bind an application to a specific network interface, or to create network namespace environments being isolated from each other [100].

Given these features, it is clear how these features might allow to use the IoT-oriented MIG for aerial IoT application, i.e., the collection of data generated by ground-located devices through BLE or Wi-Fi, with a consequent transmission to remote cloud ASs through LoRaWAN or 4G LTE cellular, as well as UAV-oriented MIG solutions,where multiple protocols (i.e., Wi-Fi, cellular and LoRaWAN) can be used to exchange data between the UAV and the control or application center.

A block representation of the proposed enhanced UAV-oriented MIG (allowing UAV-to-X connectivity) is shown in Figure 2.20, where non-IP and IP-enabled modules are detailed together with possible information flows among them.

More in detail, as clearly stated before, the use of long- and medium-range protocols, such as Wi-Fi, cellular connectivity and LoRaWAN, together with the heterogeneous capabilities of the MIG, enable the exploitation of different protocols for different applications, each featuring different requirements (as pictorially depicted in Figure 2.21). As an example, besides using Wi-Fi and 4G LTE cellular connectivity to transmit data gathered from on-field BLE sensors toward a remote AS, these com-

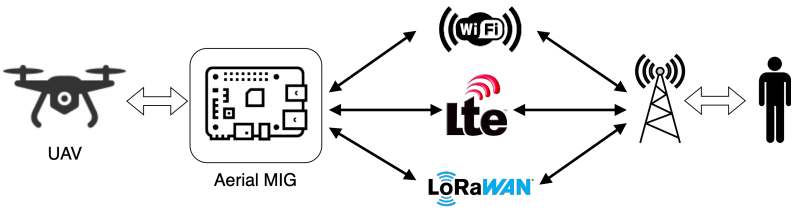


Figure 2.21: Aerial MIG adopted as for heterogeneous connectivity on UAV.

munication protocols could be used for different applications. More in detail, during the experimental evaluations of the proposed aerial MIG, LoRaWAN has been used to broadcast the UAV telemetry (carrying GNSS coordinates of the flying drone) through an available public LoRaWAN network, in this way achieving interesting results in terms of operating range, as further discussed in Subsection 2.6.1.

Instead, cellular and Wi-Fi connectivity have been expedient for transmitting to the ground control station the video feed captured by the camera attached to the on-board RPi-based aerial MIG, together with the complete telemetry data (involving the GNSS coordinates of the drone, as well as battery and altitude data). More in detail, the video feed is transmitted to the AS through the 4G LTE connection, while the telemetry is published (thanks to MQTT) toward an online MQTT broker running on the same server receiving the video feed. However, while the video transmission is assigned on the 4G LTE network interface (thanks to the Linux namespaces) on the aerial MIG, the telemetry is sent over the default network interface, which is regulated by the DHCPCD metrics system. To this end, the Wi-Fi connectivity is preferred (as per default behaviour), given its lower metric value. However, due to its limited operating range, Wi-Fi might suitable only for short-to-medium range flight missions, close enough to the ground control center. So as, given these remarks, it is clear how this brief evaluation on exploiting both Wi-Fi and 4G LTE cellular connectivities for UAV applications may leads to the need to further investigate performance, constraints, and possible improvements of the cellular connectivity for aerial devices scenarios, as it will be further carried out and discussed in Chapter 4.

2.6.1 Aerial MIG LoRaWAN Coverage Range Evaluation

The capabilities of the LoRaWAN communication link still possible between the flying UAV (enabled by the proposed aerial MIG) and the remote AS, for the drone's telemetry broadcasting, has been investigated aiming at evaluating feasibility and operating range of this particular UAV-to-Ground communication link. To this end, a Python script, in charge of retrieving GNSS latitude, longitude, and altitude from the UAV's FC, has been run on the aerial MIG. Then, these information have been periodically sent through the MIG's LoRaWAN interface, in order (i) to observe and estimate which terrestrial LoRaWAN GW (belonging to the public TTN [98] LoRaWAN open network) successfully received them, and (ii) allowing to further determine the *in-air* distance between each terrestrial GW and the reference UAV.

In detail, as shown in Figure 2.22, the aerial MIG (i) has been mounted on a custom-built UAV, based on the Tarot 650 frame, (ii) connected through a USB cable to the serial port of the Pixhawk FC, and (iii) powered using a voltage converter connected to the UAV's battery.

Then, experimental evaluations have been performed in a semi-urban environment in Sabbioneta, Italy (in an open area in the absence of mountains or hills), and carried out on December 2020, during a cloudy day with almost no wind, an air humidity of about 80%, and an environmental temperature around 3°C. More in detail, the UAV's GNSS position has been kept fixed during ascending, hovering, and descending phases, while during the flight, the drone has been kept in a hovering status at a fixed altitude of 100 m for 12 min, slowly rotating itself on the yaw axis. On the communication side, the developed script sent the new telemetry data every 10 s to the MIG. Therefore, the aerial MIG transmitted a LoRaWAN message with SF7 at a message interval rate equal to 10 s, containing the GNSS data (longitude, latitude, altitude, and Horizontal Dilution Of Precision, HDOP). Finally, the endpoint applications attached to the TTN's AS have been integrated with the TTN Mapper [101] component, in order to decode the received messages' payload and plot the data over a map, in this way highlighting those LoRaWAN GWs which have correctly received at least one LoRaWAN uplink message—allowing to easily estimate the distances among the UAV and the different GWs.

Figure 2.22: Aerial MIG located on the Tarot 650 quad-copter.

Hence, the testbed envisioned before allowed to achieve interesting results, with exchanged packets being received from GWs far away from the flying UAV. In detail, a total of 72 LoRaWAN messages have been sent from the UAV-enabled MIG during the whole flight, then being received by 14 LoRaWAN GWs registered (with known positions) on the TTN LoRaWAN network, consequently allowing to estimate the distances between them and the reference quad-copter, together with the number of packets successfully received by each GWs and the percentage of correctly received packets. For the sake of completeness, the results of this LoRaWAN coverage range evaluation are shown in Figure 2.23, where messages received by TTN GWs are shown on top of a TTN Mapper map. Moreover, the messages are listed in Table 2.3, too, where the GWs have been ranked on the basis of the number of successfully received packets. It should be mentioned that all these experimental evaluations have been performed with the UAV flying below 120 m AGL, which corresponds to the legal height allowed by the EASA [21].

Analyzing the experimental results shown in Figure 2.23 and detailed in Ta-

Figure 2.23: TTN Mapper output.

ble 2.3, it can noticed how the majority of the GWs that received the messages sent from the originator UAV are located (on the average) within a circular range with radius between 30 km and 60 km, with a few GWs placed even more than 75 km away from the transmitter UAV. Unfortunately, the only known nearby GW (identified as #6), located at about 4 km from the UAV, is an indoor GW located at the fifth floor of a building; this could explain the relatively low packet reception rate of the GW #6 achieved during the performance evaluation. Despite the absence of statistical meaning of the obtained results—due to the very small number of messages sent from the UAV—this preliminary evaluation highlights the applicability of LoRaWAN to UAV-to-Ground telemetry transmission, since the resulting communication range turned out to be much longer than the expected 10-15 km range.

Moreover, given that LoRaWAN-based communications do not entail a specific association between the end node (the UAV) and a GW, the transmitted messages can be received by any nearby GW and, thus, can be successfully processed, making the final correct packet delivery ratio in the proposed experimental evaluation equal to 95.83%, which can be considered as an acceptable rate.

Table 2.3: Aerial MIG LoRaWAN UAV-to-Ground range test results.

GW [#]	Distance [km]	RSSI [dBm]	Received PKTs	Received PKTs [%]
1	68.199	−113.72	69 of 72	95.83
2	75.040	−110.16	69 of 72	95.83
3	49.936	−110.05	61 of 72	84.72
4	31.505	−113.13	60 of 72	83.33
5	67.339	−107.98	57 of 72	79.17
6	3.933	−107.09	56 of 72	77.78
7	61.193	−113.73	49 of 72	68.06
8	44.330	−116.53	40 of 72	55.56
9	51.385	−118.86	35 of 72	48.61
10	48.904	−117.10	30 of 72	41.67
11	50.691	−118.25	12 of 72	16.67
12	68.206	−116.20	5 of 72	6.94
13	59.384	−114.75	4 of 72	5.56
14	58.800	−118.67	3 of 72	4.17

Moreover, a further analysis on the gathered data allows to observe how (with reference to the packets transmitted during ascending and descending phases) all the packets transmitted at an altitude higher than 35 m AGL have been received by at least one nearby GW, thus suggesting that the same packet reception rate should be easily achieved for almost any flying altitude above 35 m AGL, with a more realistic flying elevation set between 50 m AGL and 60 m AGL, in order to obtain VLOS communications in almost all directions.

Finally, it might argued that during the experimental evaluation, two aspects have not been analyzed: (i) the impact of the mobility on the achievable performance, in particular with a UAV flying with a linear speed between 20 km/h and 50 km/h, and (ii) the impact of messages filled with their maximum size admissible payload. To this end, the mobility of the UAV should partially limit the reception rate, given the analyses discussed in [102]. However, given the typical speeds of a UAV, the effects

of the mobility on LoRaWAN communications can be considered as negligible for payload sizes smaller than the maximum admissible (in turn leading to the longest message airtime).

2.7 Improvements

Given the simulation and experimental performance results detailed and discussed in Section 2.4 and Section 2.5, together with the modularity of the proposed MIG architecture, further improvements might be considered in order to improve its performance. To this end, possible ideas are presented and discussed in the following.

2.7.1 Queue and Packet Overhead Optimization

As confirmed by the analysis presented in Section 2.5, the use of the proposed SDB (detailed in Section 2.2) for DTUs' queuing purposes partially limits the performance of high throughput interfaces (such as Wi-Fi and cellular). To this end, the implementation of a faster and reliable queuing solution (e.g., based on the Zenoh protocol [103]) may further improve both reliability and performance of the MIG, making it applicable to more complex and critical scenarios.

Moreover, the current version of the MIG is not optimized for massive data transfer (through Wi-Fi and cellular interfaces). Hence, an enhanced implementation of the MIG with a more efficient (internal UDP socket creation and) management might improve the IP-based protocols' performance. This would significantly reduce the packet overhead currently affecting the prototypical MIG and, in turn, improve the overall performance of both cellular and Wi-Fi communication interfaces.

2.7.2 Enhance BLE Connectivity

Focusing on the BLE communication interface, as discussed in Subsection 2.3.2 the BLE capabilities seem to be mainly constrained by the BLE connection time required by external BLE end nodes to establish a communication link with the MIG. To this end, it might be possible to reduce idle times by scheduling and optimizing the

connection phase, allowing also each external BLE node to keep its connection alive and to exchange multiple BLE packets with the MIG. This approach would drastically increase the maximum throughput achievable by the BLE interface handler of the MIG, allowing to approach the theoretical upper bound of the BLE application level data rate.

Furthermore, a different BLE interaction scheme might be adopted among *on-field* end nodes and the MIG, in particular for specific time-constrained applications—e.g., those requiring only mono-directional communication, like for aerial IoT data collection performed by a *passing-by* drone. To this end, it could be possible to exploit BLE advertising channels, where BLE end nodes could act as independent GATT servers broadcasting new information (e.g., collected through their *on-field* sensors) through Packet Data Units (PDUs), which will be then advertised with a small interval on each available BLE advertising channel.

Therefore, the aerial MIG would act as a passive BLE scanner, sensing for the available PDUs in the air and processing them according to the rules detailed in Table 2.1. As a consequence, this would require the implementation of a new interface entity in the architecture shown in Figure 2.1(b), having to (i) passively scan and detect BLE devices advertising PDUs and (ii) forward the PDUs to the SDB in the proper way. Given the experimentally observed BLE connection time constraint, this alternative approach would allow to handle a larger number of end nodes and to obtain a higher throughput on the BLE communication interface (despite the limited 27 byte advertisement packet size) [104].

2.8 Final Considerations

The MIG architecture presented in this chapter enables several heterogeneous applications, spacing from IoT to UAV connectivity scenarios. Given its aim to be applied in heterogeneous IoT scenarios, then it can be claimed that (i) the development of a Markov chain-based theoretical model validated through both (ii) a Python-based simulator and (iii) experimental performance evaluations, represent a crucial tool useful for the preliminary evaluation of the capabilities of the proposed MIG in various

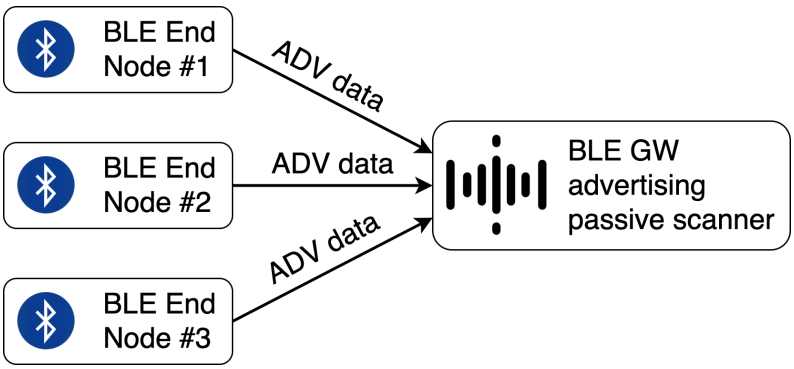


Figure 2.24: Connection-less BLE-enabled end nodes and MIG, exploiting a data transfer through BLE advertisement channels.

applications.

Moreover, the openness and modularity of the MIG allow its deployment in several scenarios, since its scalability toward additional network interfaces is guaranteed by its internal composing modules. To further prove this achievement, the proposed enhanced version of the aerial MIG for UAV-to-X connectivity has been validated, showing some UAV-related application of the aerial MIG implemented on a real UAV system. Finally, further improvements in order to increase the overall capabilities of the MIG have been detailed, thus opening to the integration of additional network interfaces for different heterogeneous IoT applications and scenarios.

Chapter 3

Wireless Mesh Communications

In this chapter, a discussion on wireless mesh communications for UAVs is detailed, aiming at both investigating and evaluating the requirements needed to enable different kind of missions involving multiple UAVs and other kind of robotics platforms in different scenarios, spacing from the VLOS conditions provided by the sky, as well as BVLOS conditions of harsh environments, i.e., underground mining tunnels.

To this end, according to VLOS and BVLOS environmental operating conditions of the wireless communications, two possible wireless mesh setups are possible: (i) an on-board WMN, where network nodes are mounted on UAVs having VLOS conditions; and (ii) an off-board WMN, where an external infrastructure provides the connectivity to the UAVs, given the BVLOS condition.

More in detail, in the first part of this chapter, possible solutions for VLOS applications are discussed. This can be the case, as an example, of a hybrid on-board wireless opportunistic mesh networking approach based on LoRa [105] and IEEE 802.11s [106] protocols for flying UAVs' swarms, involving both UAV-to-UAV and UAV-to-Ground communications. Thus, the choice of these communication protocols is motivated by the fact that the aerial MIG presented in Chapter 2 might be already suitable for exploiting these protocols, consequently with only the hybrid protocol rules to be defined.

Instead, the second part of this chapter discusses on a possible fully wireless Wi-

Fi-based mesh network to enable high throughput and low latency BVLOS missions involving several kind of robotic platforms in BVLOS scenarios. Preliminary performance evaluation, based on the throughput and operating range metrics, are shown, as well as possible improvements are considered.

3.1 Overview

An interesting aspect of UAV's connectivity is related to the possibility to allow multiple communications links between several *in-flight* drones, as well as toward multiple ground-located nodes, given the VLOS condition guaranteed by the aerospace flight environment. This is more and more true especially within 120 m AGL flight altitude, which in Europe is regulated by the EASA and in the US by the Federal Aviation Administration (FAA) [107].

Focusing on these connectivity scenarios, the proposed hybrid opportunistic mesh networking opens new possibilities for UAV-based applications over wide operational areas. LoRaWAN (at Medium Access Control, MAC, layer), as well as LoRa (at physical, PHY, layer), are attractive as they allow to reach long distances, on the order of several kilometers, as already demonstrated in Subsection 2.6.1 with regard to VLOS applications [108,109]. On the other end, IEEE 802.11s (Wi-Fi) [1] is widely adopted in a multitude of terrestrial applications and is often used, although in the IEEE 802.11 variant, for the UAV-to-Ground communication link allowing to broadcast real-time video arriving from the on-board camera, given its direct visibility operating distance ranging from a few hundreds of meters up to one or more kilometers. Nevertheless, in a complete VLOS environment (such as between flying UAVs), the selection of an appropriate Wi-Fi implementation can guarantee high bandwidth data stream between multiple nearby UAVs, thus opening additional possibilities.

In order to exploit the strengths of both LoRaWAN and IEEE 802.11s, and to limit the impact of their constraints in UAV-to-UAV connectivity-based scenarios, a "meshification" of LoRa- and IEEE 802.11s-based communication patterns (through opportunistic switching, handling, and management mechanisms) could allow to cover several use cases based on the use of UAVs swarms, as well as providing a good *trade-*

off between operational range and available bandwidth. Then, combining different heterogeneous wireless communication networks—namely a LoRa-based communication layer, for long distances and constrained payload communications, and an IEEE 802.11s-based communication layer for mid-range and unconstrained payload applications through the use of the MIG presented in Chapter 2—allow them to coexist. Being on two separate layers, a smart switching mechanism must be defined to choose the most suitable network to be used in a certain time instant, also based on task's requirements and network quality conditions (e.g., based on RSSI, and localization of flying neighbors UAVs or ground control stations). Therefore, the proposed switching mechanism takes into account the capabilities and constraints of each available communication protocol, in terms of both operational range and admitted payload, optimizing the performance in the considered application scenarios.

Furthermore, the adoption of Wi-Fi connectivity, this time for critical BVLOS applications in harsh environments involving multiple devices, easing deployable fully-wireless Wi-Fi mesh COTS solutions optimized for such applications, is discussed, investigating the performance in multi UAVs swarms missions with high throughput and low latency requirements, aiming at allowing real-time data exchange inside the swarm, i.e., point clouds generated from LiDAR or cameras, video stream or C2 tasks.

The remainder of this chapter is organized as follows. In Section 3.2, the communication protocols adopted in the proposed hybrid architecture are detailed and analyzed, discussing their strengths and limitations. Section 3.3 presents the proposed hybrid LoRa-IEEE 802.11s opportunistic mesh networking approach, while in Section 3.4 the opportunistic network interfaces switching mechanism is discussed. Section 3.5 discusses on a theoretical performance evaluation of the proposed hybrid solution, while some use cases are highlighted in Section 3.6. In Section 3.7, experimental performance results of a fully wireless Wi-Fi mesh COTS solutions are shown, while Section 3.6 discuss a few relevant use cases and scenarios. Finally, in Section 3.8 some conclusions and possible improvements of the proposed system architecture are drawn.

3.2 Disjoint IEEE 802.11s and LoRa-based Wireless Mesh Networking

The goal of this framework is to combine LoRa and IEEE 802.11s protocols to develop an innovative hybrid mesh architecture. For the sake of completeness, an overview of IEEE 802.11s and LoRa/LoRaWAN is provided in Subsection 3.2.1 and Subsection 3.2.2, respectively.

3.2.1 IEEE 802.11s Mesh Protocol

The growing interest in mesh networking over the last decade, both within industrial and academic circles, has culminated in the development of the IEEE 802.11s protocol [106,110]. This protocol represents an extension of the IEEE 802.11 standard [1] and is specifically tailored for mesh networks. Notably, it shares the same PHY layer characteristics with IEEE 802.11 but introduces new routing procedures at the MAC layer, rather than the network layer. To facilitate efficient routing, wireless nodes must possess accurate knowledge of the links connecting them to their immediate neighbors within a single hop. In detail, an IEEE 802.11s mesh network, also denoted as Mesh Basic Service Set (MBSS), is composed by different logical components [111], as highlighted in Figure 3.1: (i) mesh stations (mesh STAs), participating in the creation of the MBSS (with no hierarchical structure), having the same computational capabilities and being involved into path selection and packet forwarding (thus leading to a very simple self-organizing network); (ii) Mesh Access Points (MAPs), required to integrate a MBSS with other types of networks and to provide access to external networks; and (iii) mesh gates, corresponding to logical components enabling the integration between MBSS and infrastructure BSS, thus enabling the communication between mesh STAs and non-mesh STAs. Finally, in order to enable the communication with non-IEEE 802.11 Local Area Networks (LANs), other logical components, denoted as Mesh Portal Points (MPPs), should be used to enable interactions with external entities.

Comparing a traditional IEEE 802.11 network with an IEEE 802.11s-based mesh network, the latter has some advantages: (i) it enables low-cost and rapid deployment,

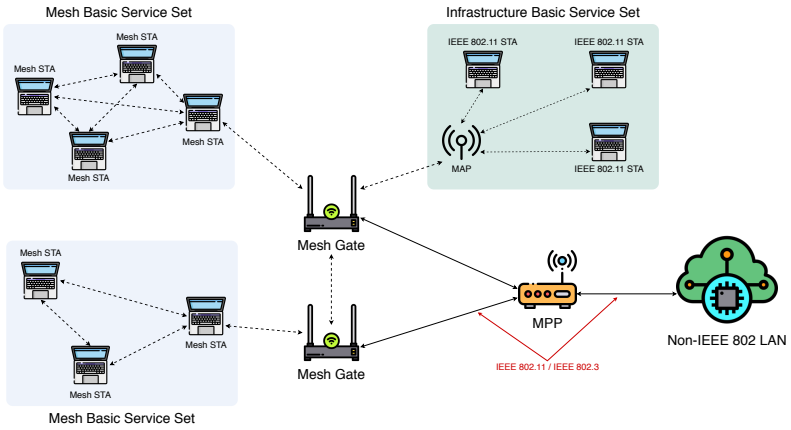


Figure 3.1: Architecture of MBSS containing mesh STAs, IEEE 802.11 STAs, mesh gates, and MPPs, as defined by IEEE 802.11s [1].

providing coverage in scenarios where wired networking is difficult; (ii) it is scalable and self-organizing, since, thanks to the routing algorithm adopted inside the mesh, the network can quickly adapt to the presence of a new joining node; and (iii) it can provide a greater coverage, exploiting multi-hop forwarding. However, there are also some drawbacks: (i) the wireless medium is less reliable than the wired medium; (ii) the data rate is a decreasing function of the number of hops and (iii) the latency significantly increase with every additional hop.

In an IEEE 802.11s-based network, the initial phase involves the formation of the network topology through the exchange of messages known as *beacons*. These beacons contain essential information about the network itself. The first step in creating the mesh network is the *discovery phase*, which can follow either a *passive* approach or an *active* approach. In the passive approach, mesh stations (mesh STAs) scan the environment to receive beacon frames actively transmitted by neighboring nodes. These beacon frames serve multiple purposes, including topology maintenance and synchronization. When a mesh STA receives a probe request frame from another node, it responds with a probe response frame, indicating its presence to neighboring nodes. Notably, one critical piece of information exchanged in these frames is

the `Mesh ID`, which uniquely identifies a specific Mesh Basic Service Set (MBSS). Once a mesh STA identifies the correct wireless node and establishes a connection, the link is maintained through the Mesh Peer (Link) Management (MPM) protocol [1]. This protocol ensures that mesh STAs share a common profile and can effectively communicate within the mesh network.

Finally, IEEE 802.11s-based mesh networks can cope with different routing algorithms. The default routing protocol for this kind of networks is the Hybrid Wireless Mesh Protocol (HWMP) [112], providing both proactive and reactive path selection and being based on the Ad-hoc On-Demand Distance Vector (AODV) [113] routing protocol. To be precise, the AODV protocol is properly adapted to support MAC layer routing and giving a tree-oriented mechanism, where a mesh STA (generally working as a MPP) forwards control messages to other mesh STAs. Nevertheless, other solutions, based on AODV and Optimized Link State Routing Protocol (OLSR) [114], can be adopted. Among them, the most relevant are B.A.T.M.A.N. [43] and Babel [115,116].

The utilization of IEEE 802.11s-based mesh networking within UAV swarms is particularly motivated by the direct VLOS among flying UAVs, especially at higher altitudes. This advantageous condition allows Wi-Fi radio signals to propagate freely without encountering physical obstacles like trees, buildings, walls, and others. In essence, it aligns with the principles of FSPL governing radio signal propagation in VLOS environments [117,118]. Furthermore, when operating in urban areas, interference from residential Wi-Fi APs can be a concern. Hence, increasing the UAV's flight altitude could further limit the detrimental effect of urban (terrestrial) Wi-Fi networks on the IEEE 802.11s mesh-based networking in the UAVs' swarm and could increase the operational range, given the limited noise floor on the Wi-Fi carrier frequency spectrum. In any case, the adoption of mesh networking in UAV-centric scenarios should either consider the inner features of these mobile environments—e.g., their volatility because of limited battery energy, frequent topological changes, etc.—and thus consider how mesh-oriented protocols perform in situations where, for limited time intervals, loops and non-optimal paths among the drones may appear.

In such WMN, flying drones close to each other (within a certain distance, rang-

ing from a few hundreds of meters up to one or more kilometers) can directly communicate with each other using the IEEE 802.11s protocol. With reference to Figure 3.2, an IEEE 802.11s-based mesh network may involve different kinds of communication nodes and may be structured at the following logical[1] layers Lx:

- one (or more) root nodes at $L0$ and connected to the Internet (e.g., through a terrestrial Wi-Fi AP, a LoRaWAN network interface, or through a UAV with an *on-board* LTE cellular network connection provided by the MIG);
- one (or more) intermediate nodes connected to an upper-layer node (e.g., at $L1$, an upper-layer is one of the available root nodes located at $L0$) and, in turn, providing connectivity to lower-layer intermediate nodes, also denoted as "leaf" nodes;
- one (or more) leaf nodes, exploiting the network connectivity offered by upper-layer intermediate nodes, and not acting as relays for any other UAV.

With reference to the layered mesh network shown in Figure 3.2, a structured mesh network could be considered to have at least one root node that, in the case of a UAVs swarm, resides at $L0$ and may be connected to a ground IEEE 802.11s AP, thus providing an Internet connectivity to the other swarm's flying UAVs. Considering the operational range of the IEEE 802.11s protocol, which can be further expanded and directed towards specific areas through the use of directional antennas, it becomes feasible to utilize the IEEE 802.11s protocol for establishing connections between flying UAVs and the ground station. This setup enables the creation of a high-speed bidirectional connection between these entities, facilitating internet connectivity for the entire UAV swarm.

3.2.2 LoRa & LoRaWAN Protocols

LoRaWAN is a long-range communication protocol [119] often used to create Low Power Wide Area Networks (LPWANs) with a transmission range up to 15 km. In

[1] Note that nodes at $L0$ may be either terrestrial or aerial, but higher layers ($L1$, $L2$, etc.) are associated with aerial nodes, i.e., UAVs.

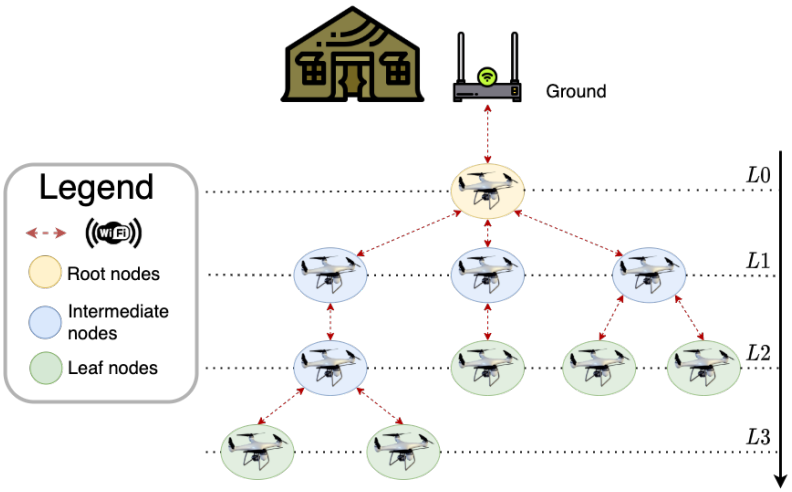


Figure 3.2: IEEE 802.11s mesh network-enabled UAVs swarm with a single "root node" connected to the Internet, involving nodes on different layers and roles.

Figure 3.3, LoRa and LoRaWAN on are reported according to the ISO/OSI layered model: LoRa is a proprietary modulation (patented by Semtech Corporation [105], the only LoRa transceiver chip manufacturer) based on Chirp Spread Spectrum (CSS) and associated with the PHY layer, while LoRaWAN is a MAC layer protocol, which relies on LoRa at the PHY layer and implements an open network architecture regulated by the LoRa Alliance. Moreover, LoRaWAN defines network media access layer rules, authentication method, device profile and data encryption.

LoRa Protocol

LoRa modems modulate symbols into increasing and decreasing frequency chirps, denoted as *up-chirps* and *down-chirps*, respectively. Then, each LoRa transmission has a *Preamble* and a *Start-of-Frame-Delimiter* (SFD), which precede the payload data in order to allow the LoRa receiver to correctly decode the incoming transmission. In turn, Preamble and SFD have different polarities [120,121], so they use up-chirps and down-chirps, respectively, depending on these polarity settings (further

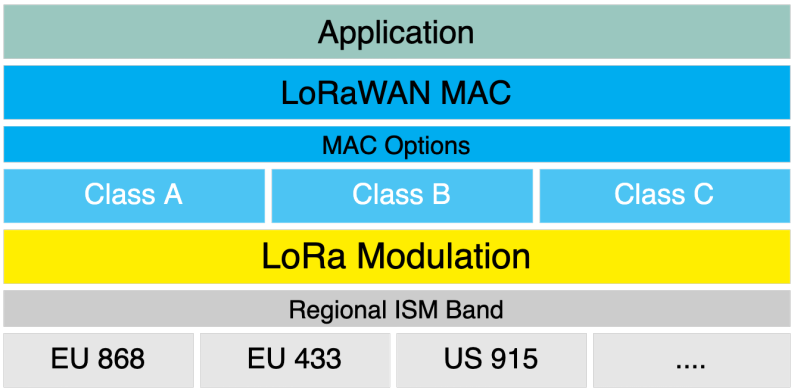


Figure 3.3: LoRaWAN and LoRa ISO/OSI model layers.

details will be provided in Subsection 3.3.2).

Depending on the geographical region of the system [122] and the PHY protocol definition, it is possible to partially manually tune the following LoRa modulation parameters.

- *Carrier Frequency* (CF), used for both transmission and listening operations and depending on the operational region: in Europe, the LoRa operational CF is the EU 863-870 MHz ISM band, while in US the 902-928 MHz ISM band is used.
- *Bandwidth* (BW), representing the width of the power spectrum density of LoRa RF signal. It is typically set to 125 kHz, but can be increased up to 250 kHz or even 500 kHz in some regions by setting specific modulations parameters.
- *Coding Rate* (CR), defining the Forward Error Correction (FEC) rate of the channel code used at PHY layer in order to limit the detrimental impact of RF interference. In particular, it affects the *symbol airtime*: decreasing the CR increases the symbol overhead (the control redundancy increases) and extends the transmission airtime. The default value of the CR is equal to 4/5.
- Spreading Factor (SF), representing the chirp spreading parameter and defining

how many chirps are sent per second. It ranges from SF7 and SF12. In detail, a large SF increases the symbol airtime and the energy consumption, thus improving the communication range, but reducing the available data rate and the messages' payload size [123].

- *Transmission Power* (TP), identifying the energy irradiated by the LoRa node's antenna. It can range from -4 dBm to $+20$ dBm, but different regions could have different power limits (for example, in Europe the upper bound is $+14$ dBm).

- *Chirp Polarity* (IQ), defining the polarity of the transmitted chirps. In detail, the polarity is often defined by the specific protocol implementation (e.g., LoRaWAN GWs transmit packets to end nodes using an inverted polarity modulation, so that these messages are discarded by neighbor GWs, while end devices transmit packets using non-inverted polarity, in order to be received by multiple GWs).

- *Sync Word*, a 1-byte value parameter defined by the last two up-chirps of the LoRa's Preamble and used to differentiate LoRa networks using the same frequency bands [124]. Therefore, any device configured with a given Sync Word will discard any incoming transmission if the Sync Word does not match its own. More precisely, default values assumed by the Sync Word byte value for *private* LoRa networks are `0x12` for Semtech SX127x devices, and `0x1424` for SX126x devices; instead, *public* LoRa networks (such as LoRaWAN or The Things Network, TTN [98]) will be represented by values equal to `0x34` for Semtech SX127x devices and to `0x3444` for SX126x devices [125].

Therefore, the choice of the LoRa modulation parameters heavily affects its operational ranges, power consumption, messages payload sizes and medium utilization, thus limiting the number of active devices in the operational area. Proper selection of the values of the LoRa modulation parameters is crucial.

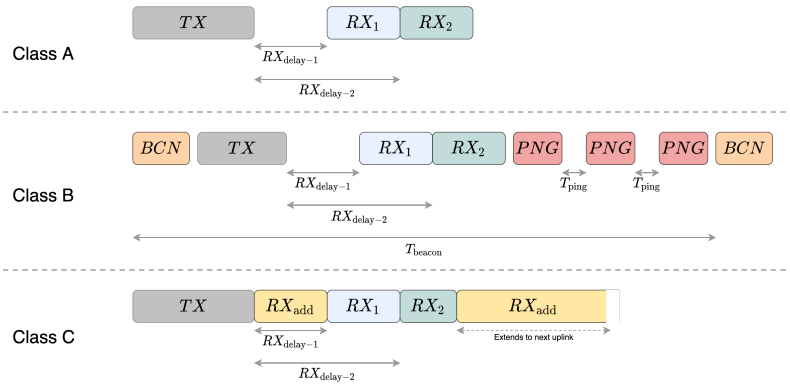


Figure 3.4: LoRaWAN Classes behavior.

LoRaWAN Protocol

In addition to the concepts discussed in Subsection 3.2.2, LoRaWAN defines different rules needed to create a multi-star network topology composed by several LoRaWAN end nodes and GWs, while its MAC layer is based on the ALOHA protocol [126]. The GWs, in turn, act as bridges between *on-field* LoRaWAN devices and upper-layer components, such as LoRaWAN NSs and ASs, thus being connected with IP-based networks and delivering data from end nodes to high layer systems and vice-versa.

LoRaWAN devices can be separated in three different classes, namely Class `A`, `B`, and `C`, according to the behavior of the end node in terms of DL/UL transmission schedule from a GW to the end node itself. All LoRaWAN end nodes must possess Class `A` features, whereas Class `B` and Class `C` devices extend these features. We now provide a few details about each class.

- Class `A` devices should support bi-directional communication, but with a specific limitation: UL (transmit) messages can be sent at any time by the end device, while DL (receive) messages can be received only during two specific reception windows at specific time instants (just after an UL transmission) before going back to sleep, as shown at the top of Figure 3.4. More in detail, after a first time interval $RX_{\text{delay}-1}$ after the end of the UL transmission inter-

val TX, a Class A device will open a short receive window RX_1 listening on the same frequency band used to transmit (UL) the previous message. Then, if no DL message is received during this interval, the end device opens a second receive window (RX_2) after a second fixed time interval $RX_{\text{delay}-2}$ (calculated from the end of TX) on a specific frequency band known by the end devices and GWs [127]. However, it is noteworthy to highlight that, despite being $RX_{\text{delay}-1}$ and $RX_{\text{delay}-2}$ generally set to 1 s and 2 s, respectively, they may assume region-specific values and can be configured by the LoRaWAN network operator [128]. Class A end devices guarantee the lowest energy consumption.

- Class B devices are suitable for more DL-demanding activities, since additional regularly-scheduled, fixed-time receive windows are defined in the LoRaWAN network in addition to those of Class A as shown in the middle of Figure 3.4. More in detail, a time-synchronized beacon *BCN* is broadcast periodically by the network via the LoRaWAN GWs, and Class B end devices must periodically receive one of these *BCN*s in order to align their internal clock with the LoRaWAN NS—beacons are transmitted by LoRaWAN GWs every $T_{\text{beacon}} = 128$ s, with this beaconing period representing a trade-off between GW transmit duty cycle's minimization and end device's power consumption [129]. Therefore, on the basis of the beacon timing reference, Class B devices can periodically open additional receive windows defined as ping slots (*PNG*s), any of which may be used by the LoRaWAN NS to initiate a DL communication, and with a ping slot periodicity $T_{\text{ping}} \in [1, 128]$ s.

- Class C devices are appropriate for DL-intensive scenarios, keeping their reception windows open unless they are transmitting—this strongly increases their power consumption, but offers the lowest latency for communication between LoRaWAN NS and end devices. As a side effect, the use of portable batteries for Class C devices is typically unfeasible. We remark that a LoRaWAN end device cannot simultaneously belong to Class B and Class C [130]. More in detail, Class C end devices perform as Class A ones—implementing the same receive

windows RX_1 and RX_2—keeping, however, their RX_2 window open until their next TX UL window: this allows Class `C` devices to receive DL messages during their RX_2 window at almost any time. Finally, between the end of TX and the beginning of RX_1, an additional short receive window RX_{add} (at the RX_2 frequency and data rate) is also opened.

In order to transmit and receive data over LoRAWAN networks, LoRaWAN end nodes must be registered and enabled on the AS of the LoRaWAN network provider, which manages the LoRaWAN GWs. A LoRaWAN-enabled device can join the network in two ways: with the Over-The-Air-Activation (OTAA) method or with the Activation-By-Personalization (ABP) method. While both approaches are effective, OTAA is more secure, since each time the end node sends a *join-request* packet, it receives a *join-accept* with (i) the network identifier (`NetID`), (ii) a 32-bit identifier of the end device within the network (`DevAddr`), and (ii) a security nonce value (`AppNonce`), which will be used by the device to generate a network's session key (`NwkSKey`) and an application session key (`AppSKey`) in a secure way. At the opposite, an ABP-activated device is internally equipped with `DevAddr`, `AppSkey` and `NwkSkey`, which will be sent over the LoRaWAN network at each transmission, in order to identify the transmitting node itself.

As discussed in Subsection 3.2.2, each LoRa transmission is associated with a specific SF defining the airtime duration of the symbols and other signal features. Focusing on the LoRa EU 868 MHz band, SF7 allows the higher data rate and shortest symbol airtime, while SF12 guarantees the highest sensitivity (the weakest signal that a receiver is able to identify and process) and longest transmission range, at the cost of lowest data rate and highest energy consumption. One of the main aspects to be taken into account is the maximum packet payload allowed for each SF. LoRaWAN Network layer (NET) typically uses a 13-byte packet header for the protocol operation: this is a non-negligible value that at high SFs significantly affects the maximum available payload. More precisely, the maximum payload size is reached with SF7, which allows a maximum of 222 bytes of data inside a single LoRa packet. The minimum, instead, is reached with SF12, with a maximum of 51 bytes for user's data. In Table 3.1, the values of payload limit, data rate, bit-rate, receiver sensitivity, and

Table 3.1: LoRaWAN parameters with regard to the adopted SF. In all cases, the bandwidth is 125 kHz and the CF is set to 868 MHz.

SF	bit-rate [bit/sec]	Range [km]	Sensitivity [dBm]	Max Payload [bytes]
12	290	12+	−136	51
11	440	10	−133	51
10	980	8	−132	51
9	1760	6	−129	115
8	3125	4	−126	222
7	5470	2	−123	222

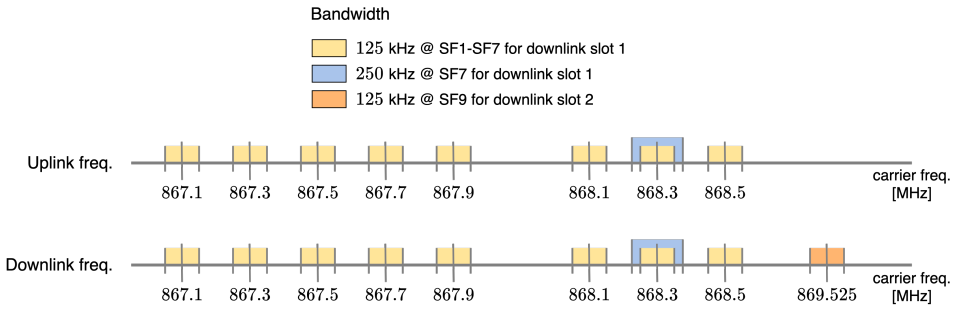


Figure 3.5: Available LoRaWAN UL and DL channels defined in the EU 863-870 MHz frequency spectrum.

typical operational range associated with each SF are summarized. In all cases, a bandwidth of 125 kHz at a CF equal to 868 MHz is considered.

The LoRaWAN protocol in Europe uses 8 UL channels (from end nodes to GWs) defined inside the EU 863-870 MHz free ISM band. These UL channels can also be used as DL channels on the *first* receiving window. There exists also a 9-th UL channel, defined at 868.8 MHz frequency and using a Frequency-Shift Keying (FSK) modulation, and a 9-th DL channel, defined at 869.525 MHz frequency and used only for the *second* receiving window. In order to better highlight the frequency availability in Europe, in Figure 3.5 we show a graphical representation of UL and DL

channels defined in the EU 863-870 MHz free ISM band.

In terms of network utilization, LoRaWAN abides by the following policy rules to be strictly followed to guarantee fair utilization:

- any LoRaWAN-enabled device has a daily maximum cumulative airtime of 30 s for UL messages;
- any LoRaWAN-enabled device has a maximum of 10 daily DL messages (including also acknowledgment, ACK, messages).

Moreover, the transmission airtime strictly depends on both SF and messages payload [131]: as an example, a fixed 10-byte payload translates to up to 20 daily UL messages with SF12, or around 500 daily UL messages with SF7. Also, LoRaWAN devices must abide by the band duty cycle [132], which depends on the regional area of use and is defined as the maximum time percentage usage of a channel [133]. For the EU 868 MHz free band, the duty cycle goes from 0.1% to 10%, depending on the used channel. This means that every time a device transmits over a LoRaWAN channel, it can not transmit on the channel for a certain amount of time, which depends on the performed transmission airtime. Finally, another constraint that LoRaWAN devices must satisfy is related to the maximum transmit power, which, for the EU868 band, is set to $+14$ dBm, as defined by the European Telecommunications Standards Institute (ETSI).

Given the discussed features of LoRa and LoRaWAN protocols (especially the long range transmissions in direct VLOS conditions), these protocols become attractive in scenarios involving UAV-to-UAV and UAV-to-Ground communications, especially in all those scenarios where a Wi-Fi AP is not available or the operational area is too large (and cannot be covered) for a mesh network among flying UAVs based on IEEE 802.11s.

3.3 Proposed Hybrid Mesh Network

The hybrid network architecture proposed in this doctoral thesis is based on two overlaid mesh layers based on different communication protocols, namely IEEE 802.11s

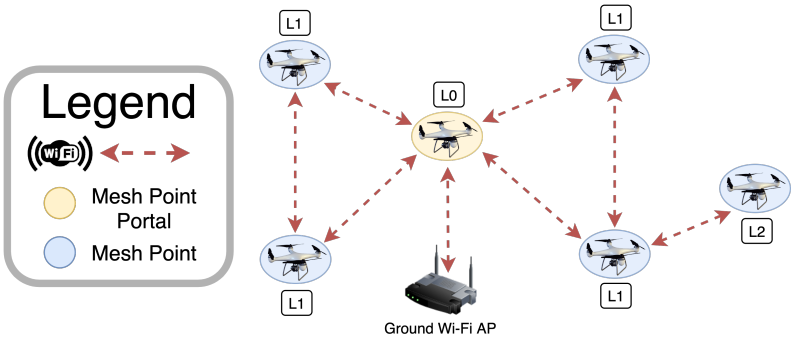


Figure 3.6: UAV IEEE 802.11s-enabled mesh nodes roles.

and a novel LoRa-based mesh protocol. The switching mechanism between the two layers and their hybrid use are detailed in Section 3.4, with a description of the various network configurations that can be implemented. In the current Section, the proposed architectures of IEEE 802.11s and LoRa-based layers are discussed, highlighting suitable implementations.

3.3.1 IEEE 802.11s Mesh Network Layer Implementation

The deployment of an IEEE 802.11s mesh network based on the concepts discussed in Subsection 3.2.1 could be based on the adoption of the IEEE 802.11s protocol, where multiple network entities are combined together in order to create a structured IEEE 802.11s mesh network.

In order to associate IEEE 802.11s' entities with flying UAVs, it is possible to map the root node with the IEEE 802.11s MPP, in turn connected to the ground through an IEEE 802.11s GW AP. Then, intermediate and leaf nodes can be seen as MPs, able to connect and reconfigure their links towards close MPs in order to keep the mesh network active. Alternatively, in order to include in the IEEE 802.11s mesh network other inexpensive IEEE 802.11-enabled UAV nodes (as STAs), some MPs could act as MAPs, thus providing network connectivity to some particular kinds of leaf nodes. In order to better clarify this association, in Figure 3.6 a reference scenario is shown.

Therefore, adopting this approach, it is possible to let the swarm to dynamically adapt to moving nodes joining the mesh network and looking for the "best" neighbor node to connect with, in turn evaluated and elected through an opportunistic selection mechanism based on heterogeneous parameters and indicators and described in Subsection 3.4.1.

3.3.2 Proposed LoRa-based Access and Mesh Networking

The adoption of long-range communications in cooperation with UAVs' swarms could find a first utilization for UAV-to-UAV communications between flying UAVs, through the use of LoRa packets. That data exchange would exploits: (i) LoRa communications, operating at SF7, 125 kHz bandwidth, and inverted polarity, in order to mitigate the interference between *raw* LoRa messages among UAVs and terrestrial LoRaWAN networks and (ii) LoRaWAN messages between end nodes and GWs. Based on the discussion on the signal polarity in Subsection 3.2.2 and owing to the fact that most of the traffic in LoRaWAN networks is given by UL packets, in the proposed raw LoRa implementation polarity inversion is adopted in order to minimize packet collisions between raw LoRa packets and LoRaWAN messages.

As an additional discussion point, the decision of adopting SF7 may be motivated by the following several reasons:

- SF7 allows a payload size of 222 bytes for LoRaWAN and 240 bytes for LoRa communications.

- SF7 guarantees the highest bit-rate and the shortest symbol airtime, thus minimizing effective channel utilization rate and packets collisions, and maximizing the number of operating devices and, therefore, exchanged messages.

- SF7 is particularly suited for mobile devices equipped with LoRa-enabled radio transceivers, such as flying UAVs (typically flying in a speed range between 20 km/h and 70 km/h, and being less affected by Doppler Shift effect [102]). At the opposite, higher SFs (e.g., SF12) are more affected by the Doppler Shift

effect, in particular at speeds higher than 40 km/h, where the higher packet loss makes the communication unreliable.

- SF7 represents the best choice even considering that the duty cycle policy defined for the EU 863-870 MHz band [132] requires that each transmission act is followed by an *off-period* without any new transmission. Given that this off-period heavily depends on the transmission airtime, then the chosen SF has an important role: SF7 has the lowest off-period between consecutive messages, given the lower symbol time.

Another key characteristic of LoRa—and, thus, of LoRaWAN—that could be exploited is related to the fact that no specific association is needed between an end node and a GW, or among two end nodes, since each LoRa-enabled device listens for incoming messages sent by neighbor devices (forwarding LoRa-modulated packets) using the same frequency and modulation parameters. This highlights how LoRa could be useful for and applicable to long-range broadcasting applications, in which one or more nodes send their messages *in-air* and nearby devices can hear them. In other words, this corresponds to a point-to-multipoint communication. Moreover, if compared with typical cellular networks, it is clear how LoRa UL and DL operations have almost the same link budget. Hence, if a device can receive a LoRa message from a nearby device, it can also send a LoRa message which can be heard by the originator node. In the end, this feature paves the way to the use of LoRa for the definition of enhanced broadcasting mechanisms between flying UAVs over a very wide area.

Focusing on the broadcasting mechanism, a Time Division Multiple Access (TDMA) scheduling of the LoRa PHY Access layer through consecutive "slots" is proposed. In turn, each slot is characterized by a transmission window structured as follows: (i) a Listen-Before-Talk (LBT) mechanism, associated with a random delay interval T_{delay} (if needed), and (ii) the LoRa message's transmission time TX_{LoRa}, depending on the adopted SF (e.g., SF7) and on the amount of symbols (thus, the amount of bytes) to be transmitted. Then, the transmission window is followed by a listening window RX_{LoRa}, during which the node listens to the LoRa channel for incoming

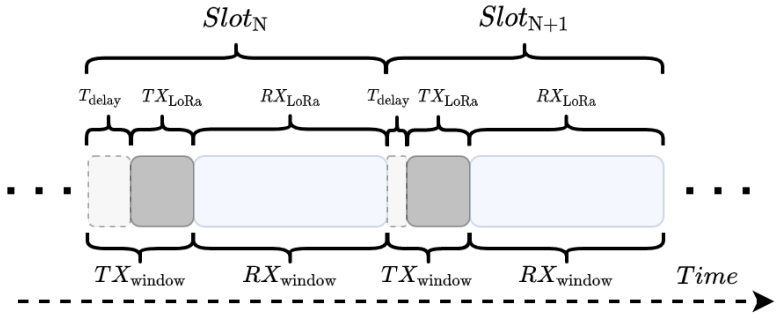


Figure 3.7: Proposed TDMA-scheduled LoRa-based broadcasting mechanism to be applied in UAV-to-UAV communications. T_{delay} is the initial random delay introduced to minimize the possible packet collision, TX_{LoRa} is the packet's transmission time, and RX_{LoRa} is the listening window on the medium.

messages broadcast by other UAVs within transmission range. For the sake of clarity, in Figure 3.7 a graphical representation of the proposed TMDA-oriented LoRa-based broadcasting mechanism is shown.

Actually, no time duration has been defined neither for the receiver window RX_{LoRa} nor for the initial random delay TX_{delay}. However, RX_{LoRa} should be at least longer than the duty cycle transmission off-period, which depends on the transmitted message's length, airtime and ETSI band duty cycle policy [134]. Moreover, the receiver window duration should also depend on the amount of active UAVs in the swarm and on the number of messages: it should be optimized in order to minimize the possible packets collisions of the LoRa PHY layer. Again, these parameters have an impact on system payload's capabilities, since the interval between messages defines the available output messages and, therefore, the aggregated payload capacity of the system.

LoRa-based UAV-to-UAV Mesh Communication

As highlighted before, while LoRa can be used for point-to-point communication and LoRaWAN for star topology networks with many end nodes and at least one

central GW, there is not a well-defined LoRa-based mesh network standard. This is likely due to the constraints of LoRa, making it unsuitable for mesh networking with multi-hop communications. This has been confirmed both theoretically and experimentally. An example is given in [135], where a hybrid LoRaWAN-LoRa mesh network with LoRa relay nodes is proposed, aiming at extending the operating range of the LoRaWAN star topology. However, this study does not consider mobile nodes and interfering signals, such as in the UAV-to-UAV application scenario of interest of us, in which a large amount of LoRa nodes is involved. Another research activity is defined in [136], where a network testbed composed of a larger (with respect to the study shown in [135]) amount of nodes is considered and the current limitations of a possible LoRa-based mesh network are investigated. Moreover, in [136] the authors observe a strong performance degradation—in terms of latency with respect to a traditional LoRaWAN star topology network. A last reference work is [137], where a LoRaWAN alternative implementation supporting mesh networking is proposed, suggesting an alternative GW implementation, in order to support multi-channels LoRa RF signal communications, to limit packets collisions, and to increase the number of supported nodes. Despite this high-level description, no theoretical or experimental results of the proposed modified LoRaWAN-mesh protocol are discussed, thus limiting its applicability for a LoRa-based UAV mesh network.

In order to overcome some of the limitations highlighted in the above cited works and to provide additional functionalities to let LoRa and IEEE 802.11s protocols coexist on mobile nodes, the proposed hybrid approach will try also to cope with the high latency and low data rate of LoRa adopting a SF equal to SF7. Moreover, in order to minimize the air time of the transmitted messages through the LoRa network interface and to maximize the payload capacity of the network, the proposed solution adopts the TDMA-oriented access mechanism presented in Figure 3.7, with alternating transmitting and listening windows with different lengths, taking also into account the number of active UAVs in the swarm. As discussed in Subsection 3.3.2, each node implements a LBT mechanism which: *first*, if an activity is detected on the channel, a node randomly delays its transmission; *then*, at the end of the random delay, it transmits; and, *finally*, it establishes a receiving window, which has to be set

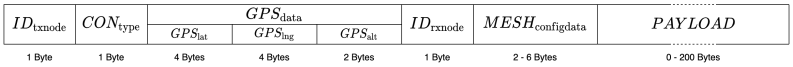


Figure 3.8: LoRa mesh packet structure, data and fields' size.

on the basis of the number of flying UAVs.

More in detail, each packet transmitted on the LoRa-based broadcast mesh network has a two-fold goal.

1. Broadcast *UAV's Position Data* (GNSS) and *Ground Connection Status* (if any): these data are needed to create a robust mesh network (both with IEEE 802.11s and LoRa) and to evaluate the position of nearby moving UAVs and their ground connections, in order to establish which will be the next node in the case of link failure or *out-of-range* position.

2. Transmit *Content Data* and LoRa *Mesh Configuration Data*: the *first* are data sent from a UAV toward another UAV in the swarm—thus an addressing mechanism is needed inside the swarm—or to the ground control center, while the *second* are essential data for mesh configuration (e.g., nearby nodes for each UAV, paths, etc.) that should be updated upon a change in the surrounding environment.

Given LoRa constraints, in order to avoid packet collisions, the above data must be aggregated in the same packet, thus requiring the definition of a proper encoding mechanism to easily allow UAVs in the swarm to evaluate the data and minimize the channel use. As can be easily understood, this may require to use the maximum payload inside each packet, in order to avoid to use the channel for multiple messages, thus reducing packet collisions and resource utilization.

A graphical representation of the proposed packet structure is shown in Figure 3.8, while the single fields composing the packet are detailed in the following.

- ID_{txnode}: a 1-byte unique identifier of the UAV, able to identify up to 256 entities. In the proposed solution, only 254 identifiers can be used, since values 0 and 255 are reserved for specific uses.

- CON_{type}: a 1-byte identifier indicating the ground connection of the node, if available. In detail, it can assume the following values:
  - 0 (default value), if the node is not connected to the ground;
  - 1, if the node is connected to the ground through an IEEE 802.11s AP;
  - 2, if the node is connected to the ground through a LoRaWAN link;
  - 3, if the node is connected to the ground through an intermediate IEEE 802.11s AP on-board a nearby UAV.

- $GNSS_{\text{data}}$: a 10-byte field, in detail containing the latitude ($GNSS_{\text{lat}}$, 4 bytes), the longitude ($GNSS_{\text{lng}}$, 4 bytes), and the auxiliary altitude ($GNSS_{\text{alt}}$, 2 bytes), encoded as `float` and `short int`.

- ID_{rxnode}: similarly to the ID_{txnode} field, it is a 1-byte unique identifier representing a preliminary field needed for data transmissions between flying UAVs (both unicast and broadcast) and to the ground center, and able to assume only 254 values (0 refers to the ground control center, while 255 to a broadcast communication between all flying UAVs).

- $MESH_{\text{configdata}}$: a variable-size field ($2 \div 6$ bytes) containing different information, such as (i) the number of hops crossed by the messages, (ii) the last hop identifier, and (iii) other parameters needed to update the created LoRa mesh network.

- $PAYLOAD$: a variable-size field ($0 \div 200$ bytes, given the LoRaWAN constraints at SF7) representing the actual data payload that a UAV needs to transmit to other UAVs.

The packet structure shown in Figure 3.8 is used both (i) for the mesh initial setup, where the $PAYLOAD$ field can be used to broadcast the initial configuration data together with CON_{type} and $MESH_{\text{configdata}}$ fields, and (ii) for the LoRa mesh network's update and data communication between UAVs.

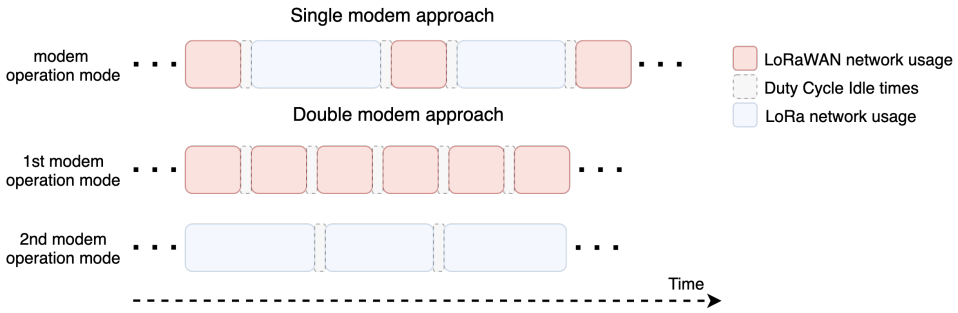


Figure 3.9: Single modem-based (top) and dual modem-based (bottom) approaches for joint LoRa (UAV-to-UAV) and LoRaWAN (UAV-to-Ground) connections.

UAV-to-Ground Communications Through LoRaWAN

While raw LoRa protocol can be used for UAV-to-UAV communications, LoRaWAN could instead be used for UAV-to-Ground communications, in order to exploit already existing LoRaWAN networks, such as The Things Network (TTN) [98]. The wide coverage of such networks, together with the achievable operational range shown in Chapter 2, guarantees a UAV-to-Ground communication alternative to an IEEE 802.11s link for essential data exchange, useful in all those cases where IEEE 802.11s is not available. However, this LoRaWAN-oriented solution has drawbacks, especially because its constraints discussed in Subsection 3.2.2, namely (i) the strict DL policy, (ii) the limited data rate, and (iii) the duty cycle, almost neglecting any real-time communication among LoRaWAN devices, thus limiting this use case only to essential data transmission.

LoRa and LoRaWAN Communications Deployment

In order to support the data transmission patterns detailed in Subsection 3.3.2 and Subsection 3.3.2, two approaches can be followed:

- (i) use of a single COTS board, equipped with two distinct LoRa modems, or (ii) the adoption of two distinct boards, each one equipped with a single LoRa modem, in order to reserve one modem for UAV-to-UAV raw LoRa-

based broadcasting and the other modem for UAV-to-Ground communications via LoRaWAN;

- use of a single COTS board with a single LoRa modem (like the MIG introduced in Chapter 2), used, according to a time division strategy, by switching between LoRa-based broadcasting (UAV-to-UAV) and LoRaWAN (UAV-to-Ground).

More in detail, the *first* approach is more expensive, in terms of price, since a complex board with two LoRa modems or two single modem boards are needed, and in terms of hardware's weight and physical space, which are limited on a typical-sized UAV. However, this approach can guarantee the best performance, since there is no off-period due to the network switch (as happens with a single modem), thus any software re-initialization phases among the interfaces. Also, since each modem is dedicated to one network, they can be used in parallel at the same time. At the opposite, in the *second* approach only one of the two networks is active at a time and switching time must be taken into account, thus reducing the effective operational time and the total system capacity. A graphical comparison between these approaches is shown in Figure 3.9.

It must be also remarked that root nodes—acting as *coordinators* in the LoRa mesh network and being the only UAVs connected to the ground control center though LoRaWAN—are the only ones which need to switch between LoRa and LoRaWAN connectivity. Therefore, only these root nodes would face the performance limits with a single-modem implementation. To this end, a possible solution to reduce performance degradation can be the pre-definition of the LoRa mesh network's root nodes, thus deploying a dual modem-based solution on them to both communicate to the ground (through LoRaWAN) and between UAVs (through the LoRa mesh network layer), as shown in Figure 3.10.

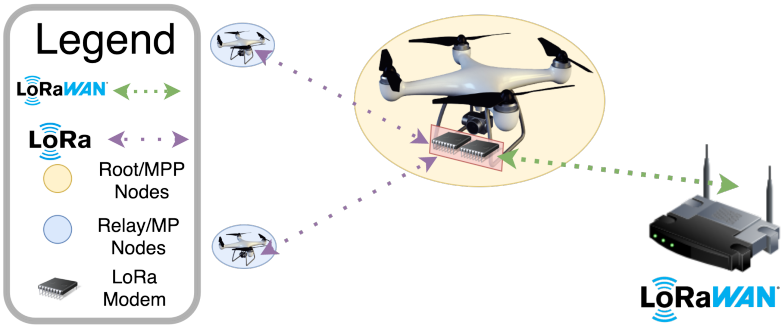


Figure 3.10: Dual LoRa modems-based implementation for root nodes.

3.4 Combined Opportunistic Mesh Networking

As highlighted in Section 3.3, the use of two different network layers allows the creation of a hybrid network involving multiple possible configurations, with a heterogeneous use of IEEE 802.11s links and LoRa-based communications. To this end, the key element enabling the definition of such a hybrid multi-protocol mesh network is the selection mechanism between them, which should properly select the appropriate network by evaluating several parameters representative of the requirements of a UAVs swarm. In the following, the proposed selection mechanism is described in Subsection 3.4.1, while Subsection 3.4.2 and Subsection 3.4.3 focus on possible UAV-to-UAV and UAV-to-Ground network configurations, respectively.

3.4.1 Network Selection Mechanism

One of the main advantages of flying communications is the almost direct VLOS between UAVs, which allows a wide operating range even with protocols typically used for short-range communications. Behind the IEEE 802.11s mesh built-in link creation mechanism [138], often based on link quality and path-based cost functions, when dealing with a mobile mesh network (as a UAVs swarm) there are several parameters to take care of, in order to establish reliable communications and ensure seamless connectivity. In the following, the parameters considered in the proposed

opportunistic hybrid architecture are detailed.

Network Communication Interface Type

As detailed in Section 3.3, each flying UAV broadcasts the data packets described in Subsection 3.3.2 through the LoRa network interface, advertising its established connection type (if any). On the basis of the best performance, in terms of data rate and reliability, each UAV will always put a *first priority* to the IEEE 802.11s connectivity for both UAV-to-UAV and UAV-to-Ground communications. *Then*, if the current node cannot join the IEEE 802.11s mesh network (e.g., it is too far from either ground or UAV-transported APs), it will consider the use of its LoRaWAN interface to communicate with the ground. If even LoRaWAN communications are not possible, then a raw LoRa communication with other close UAVs will be considered, using the mesh networking approach presented in Subsection 3.3.2 for both UAV-to-UAV and UAV-to-Ground communications.

GNSS Position of the APs

As UAVs have the flexibility to move within a swarm, their GNSS positions play a crucial role in determining connectivity within the network. When deciding which node to connect to, considering the GNSS positions of nearby mesh points, whether on the ground or carried by UAV-based Mobile Payload Platforms (MPPs), becomes valuable. This enables a flying UAV to make informed decisions regarding its connection strategy. By having knowledge of the positions of APs and considering its flight direction, a UAV can proactively prepare for AP switches, minimizing network downtime caused by AP outages.

Furthermore, GNSS positions are instrumental in aerial communication. A UAV's top priority is to establish a connection with the nearest AP-equipped UAV, prioritizing IEEE 802.11s connections over LoRa. In an advanced implementation of this opportunistic approach, the flight directions of nearby UAVs could be analyzed to optimize and facilitate seamless handovers. Additionally, altitude information from nearby UAVs can serve as a decision parameter. UAVs flying at higher altitudes are

more likely to maintain VLOS with both ground APs and nearby UAV-transported APs, reducing the number of hops required to reach an AP.

RSSI of Nearby Nodes

As an alternative to GNSS data, particularly in scenarios where GNSS is unreliable or unavailable (e.g., indoor environments or UAVs navigating close to tall buildings), another valuable metric is the RSSI collected by each network interface while actively listening for transmissions from nearby nodes. In such cases, RSSI proves to be a useful criterion for selecting the most suitable node among neighboring options, effectively serving as a selection mechanism based on link quality.

Network's Hierarchical Value of Nearby Nodes

Another vital parameter transmitted alongside other mesh-related data is the network layer of the UAV, which specifically signifies the UAV's position within the mesh network's hierarchy, as depicted in Figure 3.2. This network layer information holds significant importance and can be considered a critical indicator. For instance, when faced with multiple IEEE 802.11s mesh point UAVs exhibiting nearly identical RSSI values and located at similar distances, a UAV in search of neighboring nodes will prioritize the one with the most favorable hierarchical indicator, often the smallest depth within the hybrid mesh network. This approach minimizes the overall path cost by reducing the number of hops required for data transmission to traverse.

Amount and Type of Data to be Transmitted

The final parameter worth considering is associated with the nature and volume of data that needs to be transmitted from the source UAV to the destination node, whether it's another UAV or a terrestrial device. Given that LoRa and LoRaWAN are suitable primarily for low data rate and high-latency communications, it becomes evident that specific applications not adhering to these constraints, such as real-time video transmission, cannot rely on these long-range protocols. In such scenarios, the

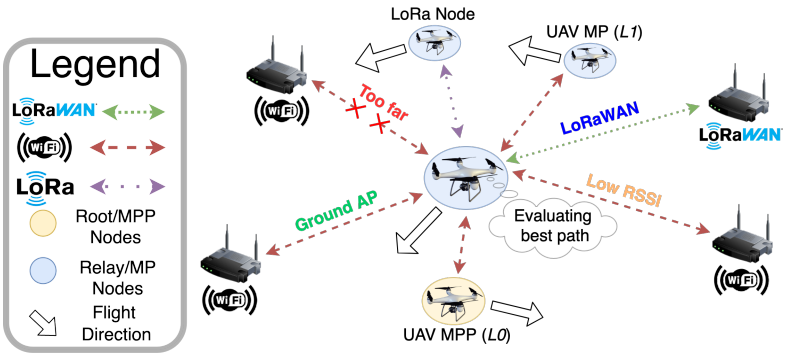


Figure 3.11: Dynamic best AP selection operated by a flying UAV, aiming at establishing connection.

establishment of mesh networks with nearby nodes is achievable exclusively through the IEEE 802.11s network layer.

Summary

As a general summary on the selection parameters outlined above, a reference scenario involving the best neighboring APs selection mechanism is shown in Figure 3.11, where a flying drone is "sensing" multiple possible mesh points and is estimating which is the best one to establish a connection with. Besides link quality/availability, the decision needs to take into account the flight plan of the UAV and the parameters described in Subsections 3.4.1÷3.4.1.

3.4.2 UAV-to-UAV Communications

In the proposed hybrid mesh architecture, UAV-to-UAV communications happen through two layers: (i) an IEEE 802.11s mesh network layer and (ii) a LoRa-based mesh layer. However, multi-hop communications between UAVs can happen in a third way, based on both IEEE 802.11s and LoRa links between UAVs. In the following, IEEE 802.11s- and LoRa-based UAV-to-UAV communications are discussed in Subsection 3.4.2 and Subsection 3.4.2, respectively, while a hybrid approach is detailed

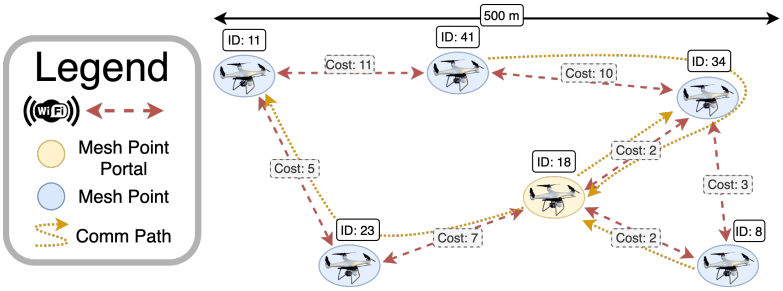


Figure 3.12: IEEE 802.11s-enabled communications between UAVs.

in Subsection 3.4.2.

IEEE 802.11s-enabled UAV-to-UAV Communications

High-performance communication among multiple airborne UAVs is best achieved by leveraging IEEE 802.11s mesh networking, capable of supporting data rates in the order of several Megabits per second (Mbps) depending on the equipment used. The setup and configuration of the IEEE 802.11s mesh network adhere to the protocol specifications. This process involves activating the path selection mechanism, as detailed in Subsection 3.4.1, and considering routing between multiple UAVs serving as MPs or, in the case of the root node, as a MPP, as discussed in Subsection 3.3.1.

Routing within this framework relies on each device's network interface MAC address, as defined by the IEEE 802.11s specification. This address is used in conjunction with the assigned drone ID, which serves as an internal parameter for data routing between both the LoRa and IEEE 802.11s layers. Figure 3.12 provides an illustrative example of an airborne IEEE 802.11s mesh network among UAVs. The figure explicitly indicates UAV IDs along with the cost associated with each link. This cost is determined based on factors such as the number of hops required to reach the root node, the data rate of the link, and the link's signal RSSI.

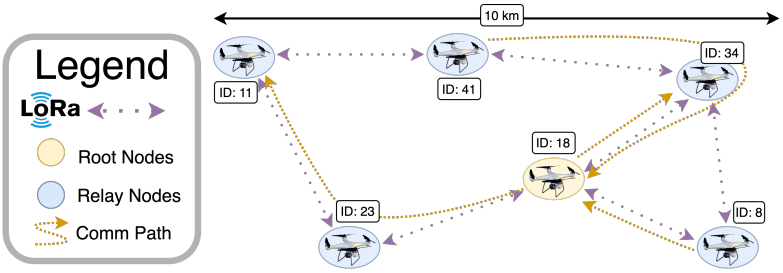


Figure 3.13: LoRa-enabled communications between UAVs.

LoRa-enabled UAV-to-UAV Communications

In order to complement the IEEE 802.11s mesh network, proposed in Subsection 3.4.2, used for UAV-to-UAV short-range high data rate communications, the proposed LoRa-based broadcast mesh network can be used for long-range communications (up to several tens of kilometers in direct VLOS), as well as for essential data communications in urban environments, where IEEE 802.11s signals, despite VLOS visibility between UAVs, could be affected by interference and signal shadowing. In such cases, the LoRa-based meshed network remains reliable for retrieving crucial data. Within this network, neighboring UAVs act as LoRa relays, facilitating long-range communications among swarm nodes, as defined in Subsection 3.3.2. An illustrative representation of such a LoRa-based mesh network configuration is shown in Figure 3.13.

Hybrid Configuration-enabled UAV-to-UAV Communications

With reference to Subsection 3.4.2 and Subsection 3.4.2, it is clear how IEEE 802.11s and LoRa have their own advantages and disadvantages: IEEE 802.11s suits low latency, high data rate, and short-to-mid range communications, while LoRa supports very long transmission ranges, but with a very limited data rate and high latency. Therefore, an effective combination of these technologies should guarantee an optimized trade-off between operational ranges and data rates.

Given the two layers discussed and detailed in Section 3.3, a hybrid adoption of

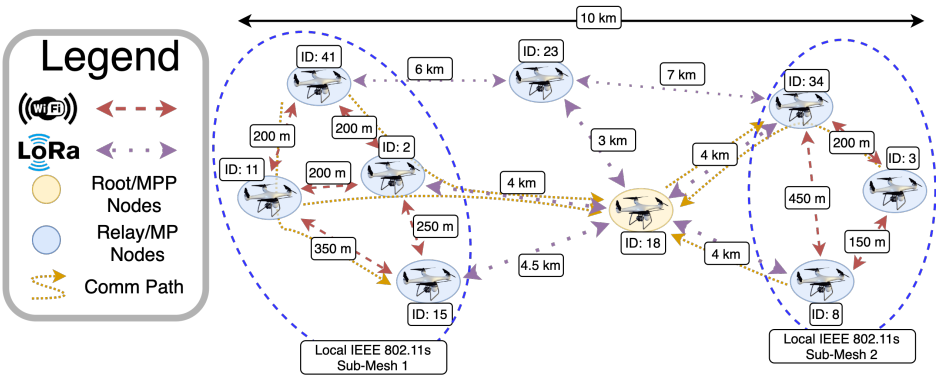


Figure 3.14: Hybrid-meshed network for communications between UAVs.

these network layers by the UAVs in a swarm could involve multiple IEEE 802.11s "local" mesh networks between close UAVs, in turn connecting together with disjoint (far) IEEE 802.11s meshed networks through the LoRa layer. This leads to the implementation of a wide area hybrid mesh network where all the UAVs can forward at least a minimum amount of data to each other, without the need of a ground infrastructure. An illustrative representation of the overall architecture is shown in Figure 3.14.

3.4.3 UAV-to-Ground Communications

Considering UAV-to-Ground communications, in the proposed infrastructure two different solutions can be identified: (i) an IEEE 802.11s-based UAV-to-Ground connection, discussed in Subsection 3.4.3, and (ii) a LoRaWAN-based one, detailed in Subsection 3.4.3. Both these solutions involve flying UAVs mesh root nodes, in charge of collecting data from the remaining flying UAVs of the swarm and transmitting it to the ground control center.

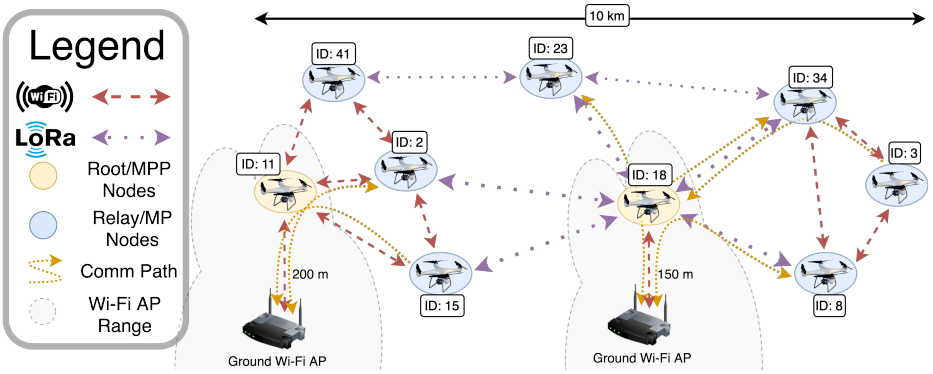


Figure 3.15: Wi-Fi-enabled ground connected UAVs swarm.

IEEE 802.11s-Enabled Ground-Connected UAVs' Swarm

Given a reference scenario in which at least one UAV is flying within the communication range of a ground control station's Wi-Fi AP, such UAV could establish a bi-directional connection with the AP itself using (i) a traditional STA-AP IEEE 802.11 Wi-Fi connection or (ii) an advanced IEEE 802.11s connection (in case multiple APs are available). Therefore, the UAV connected to the ground via Wi-Fi *first* broadcasts to all nearby swarm's nodes its connection status, *then* assumes the root node role inside the flying mesh configuration, thus providing a bi-directional connectivity among the ground center and the other UAVs. To this end, such a network configuration could be represented as in Figure 3.15, where two root nodes act as flying GWs for the intermediate UAVs (with reference to Figure 3.2).

LoRaWAN-Enabled Ground-Connected UAVs Swarm

A second approach to provide ground connectivity to the swarm's UAVs is through the LoRaWAN protocol. If compared with the IEEE 802.11s-based approach described in Subsection 3.4.3, a LoRaWAN-based approach will be characterized by an increased communication range of at least one order of magnitude, thus supporting new operational scenarios. On a practical side, a UAVs' swarm could use LoRaWAN both to interact with the ground control center and to simply send essential data (e.g.,

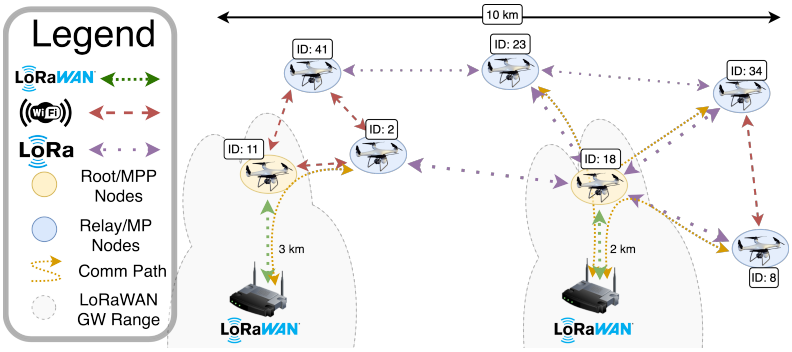


Figure 3.16: LoRaWAN-enabled ground connected UAVs' swarm.

UAVs telemetry) to a LoRaWAN AS, joining an existing open LoRaWAN network (e.g., TTN). This is possible even in the presence of more than one UAV acting as root node inside the swarm, thus providing a bi-directional connectivity between the ground center and the flying UAVs composing the hybrid mesh network. To better clarify this long-range approach, in Figure 3.16 an illustrative scenario is shown, with two root nodes acting as flying GWs for all the other UAVs that cannot directly communicate to the ground.

3.5 Theoretical Performance Evaluation

Although the proposed hybrid opportunistic mesh approach has not been implemented, some performance evaluations on the operating range of the network protocols discussed in Section 3.2 have been carried out analyzing some literature works, in particular to determine which operating ranges can an IEEE 802.11s mesh network reach and which performance can be expected. The same has been done for the LoRa protocol, with a study of existing real-world tests to extend the LoRaWAN UAV-to-Ground communication experiment discussed in Chapter 2. In Subsection 3.5.1 and Subsection 3.5.2 the LoRa and IEEE 802.11s, literature experimental performance are investigated.

3.5.1 LoRa Communication Operating Range Estimation

Given the mobile nature of the proposed hybrid architecture and the interest in the use of the LoRa protocol for inter-UAV communications in a networking range up to 15 km, a preliminary performance estimation has been initially focused on the reliability and the communication range of LoRa-based communications with at least one node moving in the considered environment. In detail, in [102], various scenarios (i.e., urban, sub-urban, and an almost obstacle-free environment) are considered, highlighting how SF12 allows to achieve a packet delivery ratio equal to 85% in both urban and sub-urban scenarios (within a range of 5 km), then decreasing to about 70% in increasing the range from 5 km up to 15 km (and depending on the environment). In [139], the authors focus on VLOS LoRa communication range evaluation, achieving a point-to-point (between a hill and a nearby flat-land) range of more than 12 km with a small packet loss ratio. Moreover, another range estimation is given by TTN Mapper, that allows to monitor, evaluate, and analyse all the data provided by the community, thus highlighting which are the real capabilities of a LoRa-based communication in various real world scenarios. To this end, among the plethora of data, the most interesting ones are those related to extreme VLOS conditions, with the transmitting node mounted over a helium-filled balloon and reaching a propagation range up to hundreds of kilometers, achieving an astonishing LoRa communication range of about 832 km [140].

As can be understood from these literature works, there is a certain potential for LoRa-based communications in terms of operating range, also allowing to evaluate the behavior and reliability of the communications in both VLOS environments, and thus making possible to use of the LoRa protocol for flying UAVs communications in a wide area between of several km, which is the goal of the proposed hybrid opportunistic network architecture.

3.5.2 IEEE 802.11s-based Communication Operating Range

With regards to *in-air* data transmission among drones through the IEEE 802.11s protocol available in the literature, some investigations have been conducted, in order

to estimate how it could fit different coverage needs. The adoption of IEEE 802.11s networking in a UAVs' swarm is investigated in [141], where an open area with no obstacle and no interference is assumed. In this context, the use of 2.4 GHz and 5.8 GHz Wi-Fi has been able to provide a high data rate link between the AP and the UAV within a radius of several hundreds of meters (up to 200 m at 300 Mbps PHY data rate on a 2.4 GHz Wi-Fi link), also depending on the adopted radio equipment and link direction (UL or DL). Also, in [141] the authors show that using a lower PHY data rate, a longer communication distance could be reached (up to 500 m at 54 Mbps PHY data rate), further expanding the IEEE 802.11s operational ranges of both UAV-to-UAV and UAV-to-Ground communications proposed in the hybrid mesh network model. Another IEEE 802.11s-based UAV network, where a UAV acts as AP for ground STA nodes, is presented in [38], showing a throughput of about 40 Mbps over a 200 m^2 area, also considering multiple hops inside the network.

Taking into account the results in [141] and [38], and the typical home Wi-Fi network operational range, the application of this technology for both UAV-to-UAV and UAV-to-Ground communications should be possible for short-to-mid range VLOS communications.

3.6 Hybrid Mesh Use Cases

In order to motivate the design and implementation of the proposed opportunistic mesh networking approach, in the following relevant use cases are discussed.

3.6.1 Aerial IoT for Farming Application

The combination between the proposed MIG and hybrid mesh network, can allow new use cases requiring UAV-to-X connectivity, as in a smart agricultural scenario with different lots far from each other, where more than one flying drones can collect data from *on-field* BLE sensors and through on-board sensors (e.g., hyper-spectral cameras). Then, in order to avoid drones to fly for long distances (between their assigned agricultural lot and the the farmer's house), the swarm can self-organize exchange the gathered data according to the hybrid opportunistic mesh network [142,

143], thus delivering them to the final point of collection. More in detail, small data can be transmitted through the LoRa-based mesh network, while large data to be transferred in a short time can rely on the IEEE 802.11s-based network, when available. This opens to innovative trajectory and task planning, aiming at maximizing the data collection rate and minimizing the "redundant" time wasted while in static flight waiting for information transfer. For example, this can rely on the adoption of Mission Management Tools (MMTs) [144] and Decision Support Systems (DSSs) [145], thus opening to the possibility of processing data both locally or in the cloud [146].

3.6.2 BVLOS Instrumental Flight Backup Telemetry

Nowadays, BVLOS flights often rely on direct communication between the UAV and the pilot on the ground. However, as detailed in the Introduction , this communication scheme is often disturbed when flying in the presence of obstacles (e.g., buildings, trees, hills, etc.) [147]. To overcome these limitations, there are two possible solutions: (i) use existing cellular networks or (ii) use relay nodes, thus allowing to provide both an FPV real-time video feeds and telemetry data to the pilots [148,149]. However, since some BVLOS flights do not always require a real-time video stream but simply the UAV's telemetry data (e.g., GNSS coordinates, altitude, speed, orientation, batteries status, etc.), in order to (i) monitor the UAV on different map layers and (ii) perform complex operations without UAV's VLOS visibility. Therefore, the use of M ($M \geq 1$) intermediate UAVs connecting indirectly (through IEEE 802.11s or LoRa intermediate links) the BVLOS UAV and its pilot, can represent a valid alternative in those cases in which, for example, cellular connectivity is not fully available or reliable, thus providing a robust backup tool to keep track of flying UAVs performing partially-automated BVLOS operations in difficult environments [150].

3.6.3 Long-Range UAVs Swarms Coordination and Communications

The proposed hybrid mesh network opens new possibilities for long-range UAVs' swarms coordination and communications, letting many UAVs communicate together and with one or more ground control centers placed over a wide operating region [151].

Ground-connected UAVs could act as swarm coordinators, communicating to the other swarm's UAVs new missions' instructions (e.g., new positions to reach or be maintained). Moreover, the proposed mesh networking scheme could simply be used to keep track of the UAVs positions, in order to provide the control center with an updated overview of the swarm topology [152]. The control center, in turn, could send to the coordinators—and, then, to specific UAVs—the information needed to restore the desired swarm formation. However, given the constraint of LoRa-based communications, the amount of data which can be transmitted between *in-flight* UAVs located over a large area is limited to essential information, such as positions to be reached, mission tasks, and other limited swarm-related information.

3.7 Portable Wireless Mesh for BVLOS Missions

One of the main advantages of the hybrid opportunistic mesh proposed in Section 3.3 is that, besides the ground located Wi-Fi APs or LoRaWAN GWs used to enable UAV-to-Ground connectivity, it does not require any ground node for pure UAV-to-UAV communications exploiting the LoRa or IEEE 802.11s mesh layers. However, while the proposed solution can work pretty while in VLOS scenario and even in BVLOS scenario which do not require high throughput and low latency, as in the case detailed in Subsection 3.6.2, there are several applications involving UAVs and other robotics platforms operating in harsh environment (e.g., in an underground mining tunnel or in a dense forest) which need a strong, reliable, high throughput and low latency wireless link to enable swarming applications. Given the constraint in terms of payload capacity and energy consumption of UAV platform, most of the IEEE 802.11s APs that can be embedded on aerial platform have limited performance, already suitable for a few tens of Mbps throughput over relatively short distances, as detailed in Subsection 3.5.2 but far away from reaching hundreds of Mbps required by some critical applications where, as an example, the point clouds generated by on-board LiDARs are shared through all the members of the swarm for autonomous navigation purposes. Therefore, it is still often required to exploit external ground deployed IEEE 802.11s networks, which, given the limited amount of

constraints, especially regarding the size and weight limits, are still one of the most effective way to provide high throughput wireless connectivity of a swarm of UAV flying within a critical BVLOS environment.

While IEEE 802.11s COTS solutions are already able to provide high throughput Wi-Fi connectivity, it is still challenging to design and deploy a portable fully wireless mesh Wi-Fi network where multiple APs disposed in a *linear* topology, like in mining tunnels, can allow low latency and high throughput bi-directional data exchange between at least two UAVs of the swarm. In the following, the COTS-based architecture of a battery-powered fully wireless portable Wi-Fi mesh solution designed to be field deployable and to allow high throughput communications between two or more robotic platform, each equipped with a LiDAR and multiple high resolution camera, is described, motivating the chosen components, configurations and then presenting the achieved results in a BVLOS mission done in an underground environment.

3.7.1 System Architecture

The proposed portable wireless mesh for BVLOS missions is composed by 5 Wi-Fi 6 Ubiquiti U6 Pro [153] APs, each one working on both the 2.4 and 5 GHz bands, with a maximum bandwidth of 160 MHz on the 5 GHz band and 40 MHz on the the 2.4 GHz band. In order to be portable and field deployable, they are powered by five Power Over Ethernet (POE) power banks, connected through an Ethernet cable. Also, in order to provide a wider good enough coverage, the 5 APs are mounted on some tripods, which are tall 1.5 m. The APs have a 4×4 Multiple Input Multiple Output (MIMO) built-in antennas operating in the 5 GHz band, which allow to reach a maximum throughput of 4.8 Gbps, while the 2×2 MIMO antennas operating in the 2.4 GHz band allow to reach a maximum throughput of 573.5 Mbps. Moreover, the 5 APs are configured to create a fully wireless Wi-Fi mesh network, where all the APs communicate in a complete wireless way. The first AP of the chain, denoted as *master* AP, is connected through the Ethernet port to the network controller, which is executed on a RPi4 SBC that runs a DHCP server as well as the network controller provided by Ubiquiti, which manages the APs configurations.

In order to allow a fast and smooth transition of the moving Wi-Fi client (e.g.,

the connected UAV) between multiple APs, the APs support Wi-Fi *Fast Roaming*, often referred to as IEEE 802.11k, v, r [154]. This technology encompasses several techniques used to facilitate smooth and rapid transitions (typically below 50 ms) between APs in wireless networks. These methods are essential for scenarios requiring seamless mobility and rely on three main concepts, as follow.

- *Key Caching (k)*: involves storing security credentials, like Pairwise Master Keys (PMKs), at an AP after initial authentication. When a client device roams to another AP within the same network, the new AP can retrieve cached security keys, eliminating the need for re-authentication and reducing re-connection time.
- *PMK Caching (v)*: known as Venue Pre-Authentication, this technique allows client devices cache the PMK of the target AP they plan to roam to before actual association. By pre-loading security credentials, the client can transition quickly when needed.
- *Opportunistic Key Caching (R)*: leverages nearby APs and their cached security information. When a client roams, it opportunistically selects an AP with cached credentials, expediting the re-connection process and ensuring minimal disruptions in service.

On the client side, the robotic platforms involved in the BVLOS experiment rely on an on-board computer based on the Intel NUC [155], which has a built-in wireless network adapter supporting both Wi-Fi 6 and IEEE 802.11k, v, r, as well as a 2×2 MIMO external high gain antennas configuration which, together with the bandwidth of 160 MHz on the 5 GHz band, allows to reach a maximum theoretical throughput equal to 2.4 Gbps.

3.7.2 Testbed and Network Setup

The performance evaluation, in terms of both throughput and operating range of the designed portable fully WMN in the aim to enable a critical BVLOS mission involving at least two devices, has been carried out in the underground tunnel of the

(a) (b)

Figure 3.17: The (a) rover adopted in the wireless mesh experiment and the (b) quadcopter involved in the wireless mesh experiment.

University of Technology of Luleå, Sweden, characterized by multiple turns, so as in the absence of LOS between the first and last hop of the wireless network. Then, in order to guarantee a high throughput and low latency communication link between the two target devices, namely a wheeled rover, shown in Figure 3.17(a), and a quadcopter, shown in Figure 3.17(b), both equipped with a LiDAR sharing its own point cloud with the other member of the swarm, then the proposed portable wireless mesh is adopted.

On the operating side, the 5 APs have been deployed in a linear topology along the tunnel in order to maintain a VLOS condition between each triple of APs (thus creating a chain of APs), as shown in the buildings map in Figure 3.18—the five APs are denoted as a numbered circle and shown with their estimated radio coverage. In the end, an overall 240 m distance has been covered through the deployment of the portable wireless mesh in the underground tunnel.

Then, in order to optimize the performance of the wireless mesh, the APs' settings have been optimized to force a prioritization of the 5 GHz band usage instead of 2.4 GHz band, in this way easing to reach a maximum throughput of 2.4 Gbps on each of the two nodes—considering the 2×2 MIMO configurations of the Wi-Fi

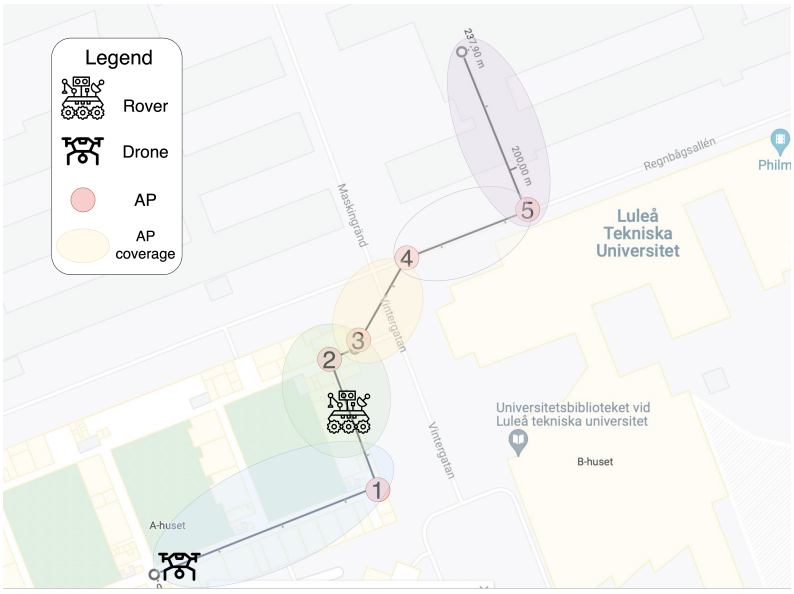


Figure 3.18: University of Luleå underground tunnel map with the APs, drone and rover positions.

network adapters installed on the devices. Moreover, given the 4×4 MIMO configurations of the antennas of the APs, together with their dual-radio support, only two antennas have been dedicated to the pure Wi-Fi connectivity for the devices nearby each AP, while the other two antennas have been configured to maintain the connectivity with the previous hop of the chain, thus performing as a back-bone for the fully WMN. Then, in order to maintaining the highest possible throughout, the APs have been located in VLOS condition each other, at a distance suitable to allow an RSSI of the previous AP greater or equal than -65 dBm—that, according to the APs control panel, allows to maximize the Wi-Fi channel capacity in terms of maximum throughput.

Finally, the experimental test has been performed with the reference quad-copter steady on a table at the beginning of the tunnel (as marked by the UAV icon in Figure 3.18), then listening for receiving the LiDAR's point cloud (sent on the proper

ROS 2 topic [8] broadcasted over the Wi-Fi network) by the rover, which in turn was moving along the tunnel. This has been beneficial for properly evaluating the network capabilities (in terms of latency and throughput) given the different number of hops and distances. Moreover, it should be specified that the point cloud broadcasted by the rover has an adaptive bit-rate that, based on the network constraints, may scale *up* or *down* the overall amount of points sent through the network, thus performing an automatic compression of the data when needed.

3.7.3 Experimental Results

The experiment results can be analysed according to the 5 "connectivity chunks" shown in Figure 3.18, especially on the basis of the number of hops (namely APs) between the drone and the mobile rover.

In the *first* scenario $\mathbb{S}_1$, with both swarm nodes connected on the AP #1, the proposed system managed to maximize the bit-rate of the LiDAR point cloud broadcasted by the rover, even with a VLOS distance between the quad-copter and the AP up to 30 m. On the experimental side, the average measured bit-rate of the point cloud fluctuated between 300 Mbps and 350 Mbps, which represents the maximum value achievable for the point cloud output of the used LiDAR, while, in terms of latency, the average measured latency between the two swarm nodes has been around 10 ms.

In the *second* scenario $\mathbb{S}_2$, with the quad-copter still being connected to the AP #1 and the rover being connected to the AP #2, the average measured bit-rate has been around 250 Mbps, while the average measured latency between the two nodes has been around 15 ms, thus higher then in $\mathbb{S}_1$, since the second hop introduced an additional latency due to internal processing.

In the *third* scenario $\mathbb{S}_3$, with the rover being connected to the AP #3, the average measured bit-rate dropped significantly to about 150 Mbps, and the same holds for the latency, whose average measured value between the two nodes has been around 23 ms. It can be argued that this sudden link's performance degradation has been due to the significant increase of the processing time that each intermediate hop adds to the communication link, as each intermediate AP has to relay the information to the previous node of the connection chain.

The *fourth* scenario $\mathbb{S}_4$, with the rover being connected to the AP #4, returned a first instability performance, with an average measured bit-rate of about 50 Mbps (but fluctuating between 20 Mbps and 100 Mpbs). It might be said that, on the overall, it can be considered as a good value, especially considering the distance between the two nodes (closer to 200 m), but with the overall fully WMN being struggling to keep the sustained bit-rate of the point cloud stream, given the relevant number of intermediate hops that, despite the steady stable wireless back-bone between them, suffers the relay processing introduced by each node. To this end, the average latency measured in $\mathbb{S}_4$ is around 42 ms, with some temporary peaks up to 100 ms.

Finally, surprisingly, in the *fifth* scenario $\mathbb{S}_5$, with the rover being connected to the last AP #5, despite the lower bit-rate and higher latency, the link became more stable, with an average measured bit-rate of 20 Mbps (fluctuating between 10 Mbps to 30 Mbps) and an average measured latency around 61 ms.

Then, with regard to roaming performance of the moving rover between the APs, during the transition between AP #1 and AP #2, the connection switching has been pretty smooth, with a temporary negligible latency drop to 40 ms. The same applied while moving from the AP #2 to AP #3, where the latency momentously increased to 50 ms. Instead, with regard to the last two hops, the transition did not happened smoothly, with a significant latency increment and several dropped frames. So as, on the average, a latency up to 500 ms has been noticed when transitioning between the AP #4 and AP #5, and vice-versa.

Finally, with regard to the battery run-time, the whole network has been powered through PoE battery banks, and this allowed to exploit the deployable WMN from 6 h up to 8 h, depending on the overall load of the network. This can be mentioned as being enough to perform multiple high-precision environmental mapping BVLOS missions in an harsh environment, also given the limited battery life of just 2 h of the rover and 25 min of the drone.

3.8 Final Considerations

The research activity described in this chapter showed two possible approaches to enable UAV-to-UAV and UAV-to-Ground connectivity according to two operational criteria: VLOS and BVLOS conditions. With regard to the first activity, given the VLOS condition between both *in-flight* UAVs and ground control stations, there is no need to rely on a third-party network, since the combination of high throughput IEEE 802.11s and long-range LoRa layers allows to perform different kind of missions involving several network topology and configurations of the layers—namely, based on the physical distances between the nodes, as well as the application requirements. However, with regard to situations in which (i) there is no possibilities to maintain a VLOS condition between UAVs, and (ii) the amount of involved drones does not allow the adoption of intermediate UAV as relays, then a third-party network is needed, especially for all those critical missions happening over a wide area and requiring high throughput and low latency links.

Then, the experimental results related to the BVLOS-oriented WMN and regarding the second activity detailed in this chapter, has been beneficial to highlight that, despite the limits observed with a large number of hops in the specific linear topology configuration adopted in the discussed experiment (inside a tunnel in a Sweden mine), the overall performance of the designed (and deployed) fully battery-powered WMN can be considered as good enough for granting high-throughput BVLOS swarming missions in critical environments. However, in order to allow robust performance and coverage over a wide area, a structured network, composed by several APs connected through a wired back-haul, is needed. Moreover, this also demonstrated how, while multiple industrial buildings and environments already have their own wireless Wi-Fi network covering most of the environment itself (indoor and outdoor), for large-scale BVLOS missions, the easiest and logic third-party network, besides Wi-Fi, might involve the use of cellular networks, being already supported by the MIG proposed in Chapter 2. However, since (as detailed in Chapter 1) existing cellular networks suffer some limitations if adopted for providing an aerial connectivity (especially at higher altitudes), then mitigation techniques might be useful, as investigated and proposed

in Chapter 4, which in detail is focused on the exploitation of cellular connectivity on UAV to enable critical BVLOS missions.

Chapter 4

UAV Cellular Connectivity Enhancement

This chapter discusses the use of the cellular connectivity for BVLOS flight missions of UAVs, where it is mandatory to rely on existing network to maintain a robust communication link between one (or multiple) UAVs and the control center (or the pilot). More in detail, the current limitations of existing cellular networks are initially analysed, discussing on literature works and focusing on some measurements carried out in real experimenal scenarios. Then, some possible solutions to overcome these constraints are evaluated by studying the literature and the research activities carried out on this topic.

Then, a novel approach to enhance cellular connectivity for UAVs applications through the use of multiple selective antennas is introduced and discussed, with the overall architecture and system design description, as well as the real implementation on a UAV and the integration of the system in the MIG presented in Chapter 2. Finally, the proposed system is validated according to a well-defined tests suite, whose results are then discussed aiming at highlighting possible future improvement of the system.

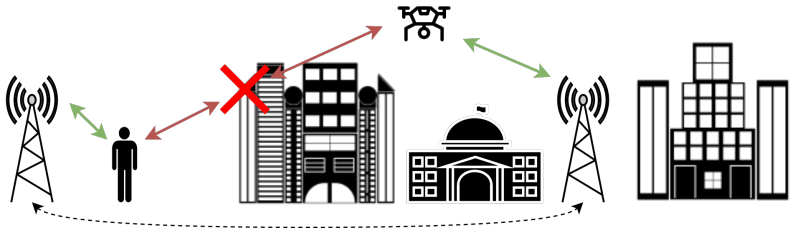


Figure 4.1: Generic cellular connectivity-enabled UAV BVLOS scenario.

4.1 Overview

Nowadays, given both regulations and technical limitations, the usage of UAVs has been mainly limited to VLOS applications or short range BVLOS flight mission, where the current existing commercial solutions can already provide a few hundreds meters of operating range. The manufacturer DJI, as an example, has already some advanced commercial UAVs using their proprietary communication protocols (OcuSync [29] protocol) still operating in 2.4 GHz and 5.8 GHz bands that is able to guarantee a several kilometers operational range in VLOS conditions and a few hundreds of meters in BVLOS scenarios. Although VLOS flights still have a huge market to conquer, most of the drone manufacturing companies are nowadays targeting an enhancement of the adoption of UAVs in BVLOS scenarios, in this way enabling complex missions and advanced services over large areas [3], like, as an example, goods delivery of building inspections.

While for indoor BVLOS missions the usage of traditional Wi-Fi mesh networks, as proved by the experiment reported in Chapter 3, can already provide a robust enough wireless link, when dealing with outdoor flight mission over wide areas where it is no possible to rely on VLOS communications links, different solutions have to be exploited. Commercial UAV manufactures already started to investigate the usage of existing cellular networks for BVLOS-like missions, thus leading to the development of high-end quad-copters exploiting the cellular technology [4] to extend their flight missions' operating range, similarly to the generic cellular connectivity-enabled UAV BVLOS representation shown in Figure 4.1.

Terrestrial cellular networks are the best logical candidate for enhanced BVLOS UAVs applications, given their widespread diffusion and reliable architecture connecting billions of devices across the world for at least a decade, thus confirming the overall quality and reliability of these networks. However, despite the wide diffusion and limited cost of the 4G LTE network adapters, together with their small footprint (making them easily to be integrated into aerial platforms), their usage on UAV is still limited (as highlighted in literature analyzed in Chapter 1) mainly as a backup solution for non essential data. More in detail, existing 4G LTE and 5G cellular networks are able to provide low latency and high data rate communication links, but they have been designed, deployed, and optimized for terrestrial usages, as most of the connected terminals (e.g., smartphones, cars, SIM-based devices, etc.) are located and moving on the ground or in the first tens meters of altitude.

In detail, among several technical aspects of these networks, their deployment and frequency reuse plans between nearby cells, whose general pictorial representation is shown in Figure 4.2, have been optimized to avoid inter-cells interference on ground-located connected terminals. In European countries commercial UAVs can typically fly at an altitude up to 120 m AGL according to the EASA regulations [21]. However, upon permission, it is possible to fly at higher altitudes, depending on the specific application which they are involved in. This implies that a cellular terminal located on a drone features a direct visibility with multiple BTSs of nearby cells, especially in geographically flat environments or in highly density populated regions, where a massive amount of pico- and femto-cells can be present in a very limited area, then leading to an intensive frequency reuse. The consequence is a strong inter-cell interference, since the system receives signals from the neighboring BTSs, that leads to a reasonable decrease in the radio signal quality metrics, in the end affecting the network's effective performance and stability, threatening the safety of critical BVLOS missions.

This interference effects, which have been reported through *in-flights* cellular network signals measurements carried out by several entities mentioned in literature analysis of Chapter 1, have also been confirmed by the *in-flight* measurement carried out with the aerial MIG developed along this doctoral thesis, whose impact on the 4G

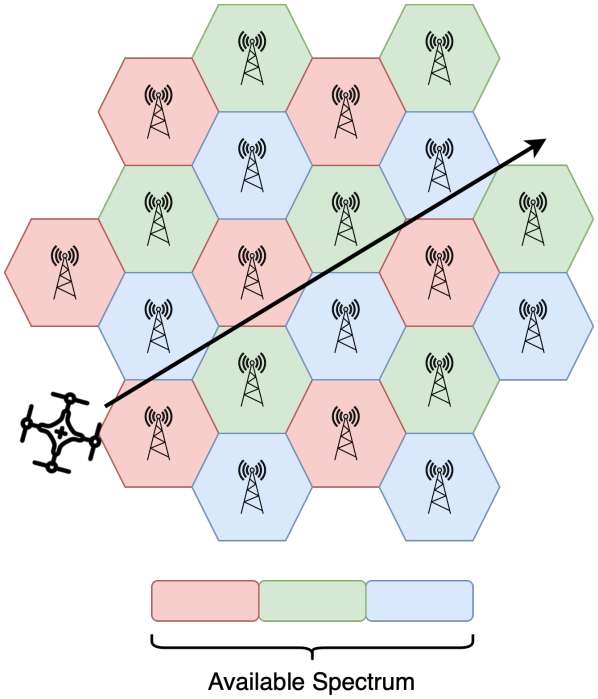


Figure 4.2: Typical frequency re-use plan for a standard cellular network.

LTE radio signal metrics are shown in Section 4.2 and have been the starting point for the development of a novel system based on the usage of selective antennas to enhance cellular connectivity for UAV.

To speed up and improve the integration of UAV cellular connectivity, given the feasibility of adopting selective antennas on a drone as well as the possibility to exploit existing cellular networks without modifying the radio equipment's mounted on thousands of cell towers, the research activity presented in this doctoral thesis focused on the design and development of a novel prototypical architecture, denoted as *Smart Selective Antennas System* (SSAS), composed by four selective antennas mounted on the drone and controlled by a proper algorithm, aiming at enhancing the UAV's connectivity reliability with the existing 4G LTE networks in BVLOS scenarios.

The remainder of this chapter is organized as follows. In Section 4.2, the re-

sults of the preliminary 4G LTE cellular connectivity radio signal measurements are presented, confirming the issues reported by the literature. Section 4.3 presents the proposed system, discussing its architecture and technical details. Section 4.4 illustrates the testing environment and methodologies exploited to validate the SSAS' performance, while experimental evaluations of the results are reported in Section 4.5. Finally, improvements regarding the proposed system and future research directions are discussed in Section 4.6, while final remarks are provided in Section 4.7.

4.2 Preliminary 4G LTE In-Flight Measurements

The aforementioned interference issue has been experimentally measured while testing the aerial MIG implementation discussed in Chapter 2 and the obtained results have been confirming the problem reported by [5,6,46,47]. More in detail, the data gathered using traditional omni-directional 4G LTE antennas during the flights in a semi-rural area in the north of Italy have been preliminary characterized in relation of the flight altitude of the UAV, in order to evaluate the impact at different flight height spacing from the ground level up to 120 m AGL. Four 4G LTE signal quality indexes have been considered, namely (i) the Signal to Interference plus Noise Ratio (SINR), (ii) the Reference Signal Received Quality (RSRQ), (iii) the Reference Signals Received Power (RSRP) and (iv) the RSSI, whose values are shown in Figure 4.3 and are relevant for the 4G LTE networks evaluations.

- According to [156], in 4G LTE networks, the RSSI (dimension: [mW]) is a parameter providing information related to the received wide band power (measured in all the symbols), including thermal noise as well as the noise generated in the receiver, everything within the measurement bandwidth. Therefore, the RSSI measures the average power of the signal received by the cellular modem and includes the power from the network's connected BTS signal, as well as other interfering signals, such as the ones arriving from nearby cells. The RSSI can be expressed as

$$\text{RSSI} = S_{\text{tot}} + I_{\text{tot}} + N_{\text{tot}} \tag{4.1}$$

where: S_{tot} (dimension: [mW]) is the useful received signal power (dimension: [mW]) of the connected cell measured over the 12 Resource Elements (RE) subcarriers defined by the 4G standard [156]; I_{tot} (dimension: [mW]) and N_{tot} (dimension: [mW]) are the interference and thermal noise powers, respectively.

- The RSRP (dimension: [mW]) is another key parameter useful to verify the 4G LTE connectivity quality and, according to [156], it indicates the useful received signal power level incoming from the connected cell tower averaged over all the used LTE REs within the designated measurement frequency bandwidth. The RSRP can be expressed as

$$\text{RSRP} = \frac{1}{N_{\text{RE}}} \sum_{i=1}^{N_{\text{RE}}} P_{r,i} \tag{4.2}$$

where: N_{RE} is the number of usable REs within the measurement frequency bandwidth, and $P_{r,i}$ (dimension: [mW]) is the power contribution of the i-th RE.

- The RSRQ (adimensional) takes into account both the strength of the reference signals of the connected BTS as well as the level of noise and interference components. It is used to assess the overall quality of the received signals, therefore allowing the modem to decide which cell to connect with. According to [156], it can be derived from the RSSI and RSRP as

$$\text{RSRQ} = \frac{N_{\text{PRB}} \cdot \text{RSRP}}{\text{RSSI}} \tag{4.3}$$

where N_{PRB} is the number of Physical Resource Blocks (PRBs).

- The SINR (adimensional) measures the ratio of the useful signal's power to the combined power of interference and background noise, thus quantifying how well the desired signal stands out from the unwanted signals and noise. It is used by the modem to calculate the Channel Quality Indicator (CQI) and therefore determine the throughput of the communication link. The SINR can

be expressed as

$$\mathrm{SINR} = \frac{S_{\mathrm{tot}}}{I_{\mathrm{tot}} + N_{\mathrm{tot}}} \tag{4.4}$$

where S_{tot}, I_{tot} and N_{tot} reflect the RSSI components detailed in Eq. (4.1). However, since S_{tot} depends on the number of used subcarriers of the measurement bandwidth, it is possible to derive it as

$$S_{\mathrm{tot}} = \frac{N_{\mathrm{RE}}}{N_{\mathrm{PRB_{used}}}} \cdot 12 \cdot N_{\mathrm{PRB}} \cdot \mathrm{RSRP} \tag{4.5}$$

where $N_{\mathrm{RE}}/N_{\mathrm{PRB_{used}}}$ indicates the load of the subcarriers and is equal to 1 when all the subcarries are used. Therefore:

$$S_{\mathrm{tot}} = 12 \cdot N_{\mathrm{PRB}} \cdot \mathrm{RSRP}\,. \tag{4.6}$$

By substituting Eq. (4.6) in Eq. (4.4), it is possible to obtain

$$\mathrm{SINR} = \frac{12 \cdot N_{\mathrm{PRB}} \cdot \mathrm{RSRP}}{I_{\mathrm{tot}} + N_{\mathrm{tot}}}\,. \tag{4.7}$$

Then, by substituting Eq. (4.1) at the denominator, it is possible to rewrite the SINR as

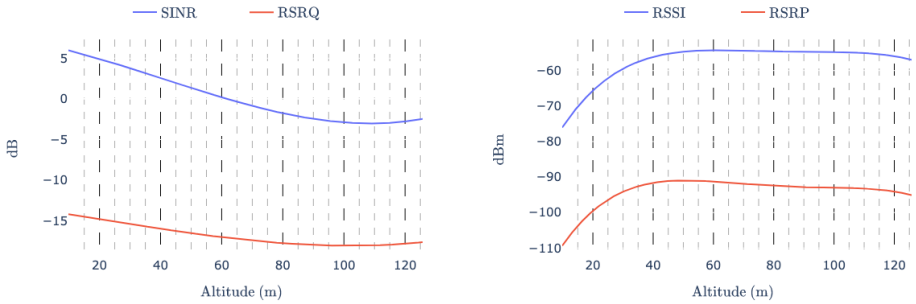
$$\mathrm{SINR} = \frac{12 \cdot N_{\mathrm{PRB}} \cdot \mathrm{RSRP}}{\mathrm{RSSI} - S_{\mathrm{tot}}}\,. \tag{4.8}$$

However, substituting again Eq. (4.3) and Eq. (4.5), it is finally possible to simplify the RSRP and N_{PRB} terms, thus obtaining

$$\mathrm{SINR} = \frac{1}{\frac{1}{12 \cdot \mathrm{RSRQ}} - \frac{N_{\mathrm{RE}}}{N_{\mathrm{PRB_{used}}}}}\,. \tag{4.9}$$

More in detail, looking at Figure 4.3(a), it can be noticed that by increasing the altitude, the SINR and RSRQ radio metrics decrease their values, thus resulting significant lower values that those at ground level. Is also important to notice that the SINR and RSRQ decrease their value while the RSSI and RSRP show a significant increase at higher altitudes, as shown in Figure 4.3(b).

The decrease of SINR and RSRQ, which are metrics that take into account the quality of the reference BTS signal with respect of the interference and noise is due



(a) (b)

Figure 4.3: Omni-directional antennas signal quality metrics different altitudes, in terms of (a) SINR and RSRQ, and (b) RSRP and RSSI.

to the direct visibility of the on-board cellular modem with multiple BTSs available nearby. The increase of the RSSI, is due to the fact that the VLOS between the *in-flight* drone and the nearby BTSs, allows the on-board modem to receive a much higher signal power with respect to the ground level, however such signal power also includes the interfering signal powers from the other BTSs, as well as of the connected one, thus explaining the significant increase. The same principle can be applied to the RSRP metric, which confirms the higher received power from the reference BTS, due to the VLOS signal propagation between the two entity of the communication link.

To further clarify the impact of inter-cell interference problem, in Figure 4.4 the Cumulative Distribution Function (CDF) of the four aforementioned 4G LTE radio signal quality indexes are reported, this tame taking into account five different UAV flight altitudes (namely: 20 m, 40 m, 60 m, 80 m, 100 m).

As can be seen, the more the UAV's height increases, the more the 4G LTE radio signal quality indexes values decrease, thus reaching significantly lower values than those observed at the lower level. This leads to a decrease of the network's effective performance and stability, in the end threatening the safety of BVLOS missions. Hence, given these remarks, to speed up the deployment of BVLOS drones-oriented

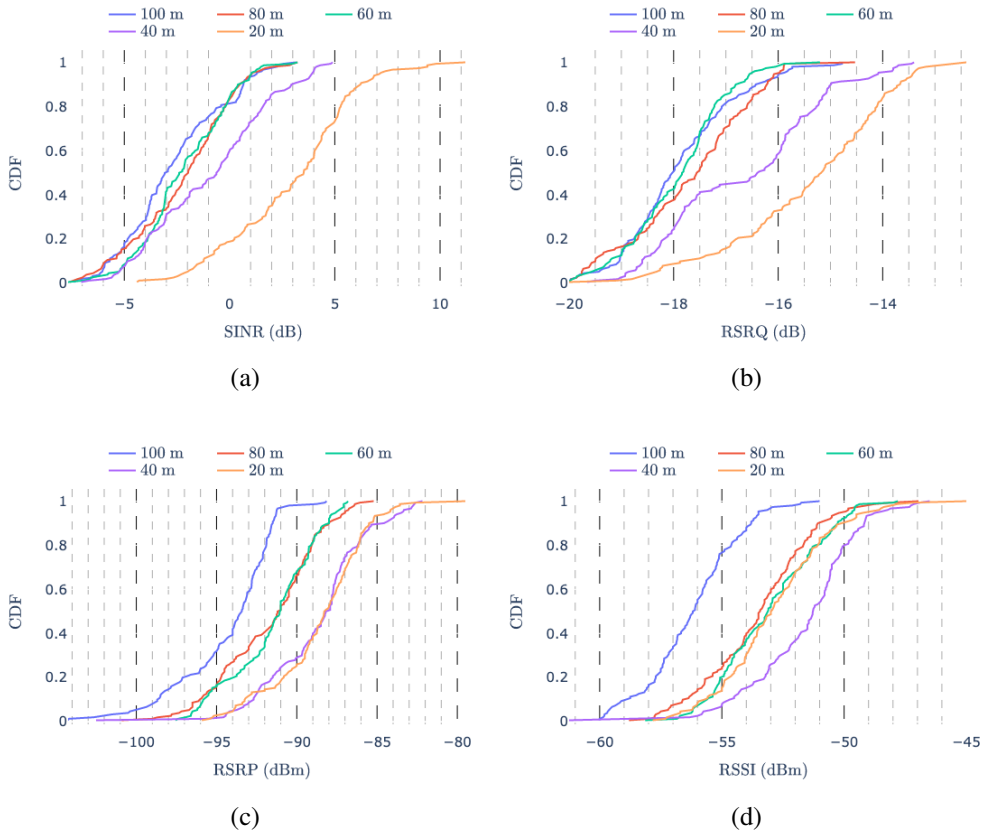


Figure 4.4: Omni-directional antennas CDF at different altitudes of: (a) SINR, (b) RSRQ, (c) RSRP and (d) RSSI.

applications and to provide a strong backup link to these devices until newer network generations (e.g., advanced 5G or even 6G) optimized for aerial connectivity will be available, it is imperative to find a solution to minimize this problem, aiming at easily and effectively use existing terrestrial networks.

4.3 System Architecture

The SSAS has been defined for enabling re-configurable directional communication, therefore it is composed by four main components: (i) four directional antennas; (ii) a switching network; (iii) the LTE cellular modem integrated in the aerial MIG mounted on the UAV; and (iv) the antennas control algorithm. The system architecture is described according the *hardware layer*, namely the antennas, the switching network and the aerial MIG integrated on the UAV, which are detailed in Subsection 4.3.1; and the *software layer*, composed by the control algorithms developed for the proposed system and illustrated in Subsection 4.3.2.

4.3.1 Hardware Layer

The prototype is composed by four 2×2 MIMO selective antennas with a beam directivity of about 90° mounted on a square support attached to the UAV platform, allowing to provide a 360° radio coverage on the horizontal plane. The antennas are connected to a four port switching network, which is in turn connected to the areal MIG's cellular modem. In particular, only the antenna pointing in the direction of the interested BTS is activated, with the aim to reduce inter-cell interference arriving from the surrounding BTSs in the other directions. Therefore the active antenna is chosen on the basis of drone's and cell towers' GNSS coordinates, obtained through the UAV's FC and via an *a-priori* access to open-data BTSs lists, respectively. A high level overview of the system is depicted in Figure 4.5.

More in detail, each antenna consists of 2 elements aligned horizontally at a 10 cm distance to enable 2×2 MIMO connectivity, therefore, including the 3D-printed frame, each antenna features a $21 \times 11 \times 2.4$ cm size with an overall weight of just 153 gr. The antennas are suitable for the LTE Band 3 ($1710 \div 1880$ MHz) and LTE Band 7 ($2500 \div 2690$ MHz) and they are connected to the switching board, which, being connected to the on-board aerial MIG through four input pins visible in Figure 4.6, allows the RPi4 to select the most suitable antenna through two digital control signals. More in detail, the switching board has 2 coaxial input connectors attached to the two AUX and MAIN antennas' connectors MIMO ports of the Sierra

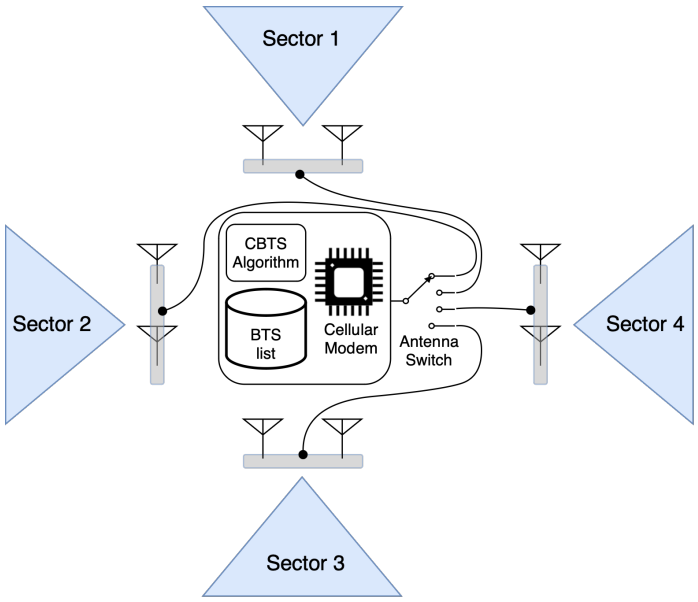


Figure 4.5: Block diagram of the SSAS, where the sectors are selected one at a time.

Wireless EM9191 [157] cellular modem (which has been used as replacement of the initial Huawei 4G LTE modem USB dongle) integrated in the MIG, while the 8 outputs, coupled two by two, are connected to each one of the four 2×2 MIMO selective antennas. A picture of the switch board with all the connection ports available is shown in Figure 4.6.

The whole system has been integrated into the custom-built UAV already used for the aerial MIG validation of Chapter 2, with a customized antennas support located in the bottom side of the UAV (between its legs, in order to lower the barycenter of the drone for an improved stability). The four 2×2 MIMO antennas have been fixed on a wood basement with a 45° shift with respect to the drone's head, aiming at removing the interference introduced by the carbon fiber legs of the UAV. Then, the aerial MIG powered by the UAV's battery, provides power to both the cellular modem (connected to the RPi4-based through a Techship MU201 [158] M2-to-USB3 adapter) as well as to the switching network. The complete SSAS system integrated on the UAV platform together with the aerial MIG is visible in Figure 4.7.

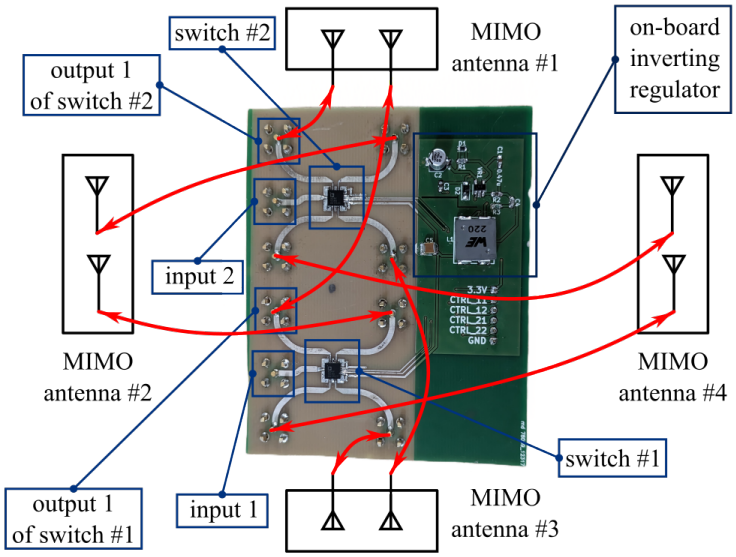


Figure 4.6: Switching network PCB.

Figure 4.7: UAV platform composed by the aerial MIG and the SSAS.

On the overall, the complete system—four antennas, switching network, aerial MIG, EM9191 cellular modem, and coaxial cables—achieves a final payload weight of about 1.1 kg.

4.3.2 Software Layer

Given the need to evaluate the performance gain of the proposed SSAS with respect to the omni-directional antennas provided by the Sierra Wireless EM9191 cellular modem manufacturer (having a gain between 3 dBi and 5.8 dBi in the bandwidth of interest), two different antennas selection algorithms have been defined and validated for *in-flight* data collection, in turn performed following the evaluation methodology detailed in Section 4.4 but exploiting different types of antennas' control criteria: (i) nearest BTS (NBTS) selection algorithm, detailed in Subsection 4.3.2 and (ii) connected BTS (CBTS) selection algorithm, discussed in Subsection 4.3.2.

Nearest BTS (NBTS) Antenna Selection Algorithm

The NBTS selection algorithm takes into account three main parameters: (i) the location of the *in-flight* UAV, based on its GNSS coordinates provided by the aerial MIG-connected FC; (ii) the flight direction of the UAV (also denoted as heading angle); and (iii) the location of the nearest BTS, represented by its GNSS coordinates. More in detail, as shown in Figure 4.8, once the SSAS obtains the telemetry data from the FC, it first loads the known BTSs' location database. Then, after a filtering operation based on the geographical distance between the drone and each BTS (based on their GNSS information), the UAV internally maintains only those BTSs in a 20 km nearby radius, in order to speed up the consequent look-up iterations. Hence, once selected the nearest BTS, the NBTS algorithm calculates the bearing angle between the UAV's GNSS coordinates and those of the selected nearest BTS. Finally, exploiting (i) the UAV's heading angle and (ii) the bearing angle of the nearest BTS, the NBTS algorithm selects the antenna (out of the four 2×2 MIMO antennas available on board) pointing toward the nearest BTS, while the remaining ones remain disabled. As a final remark, a check on the drone's heading and position is performed

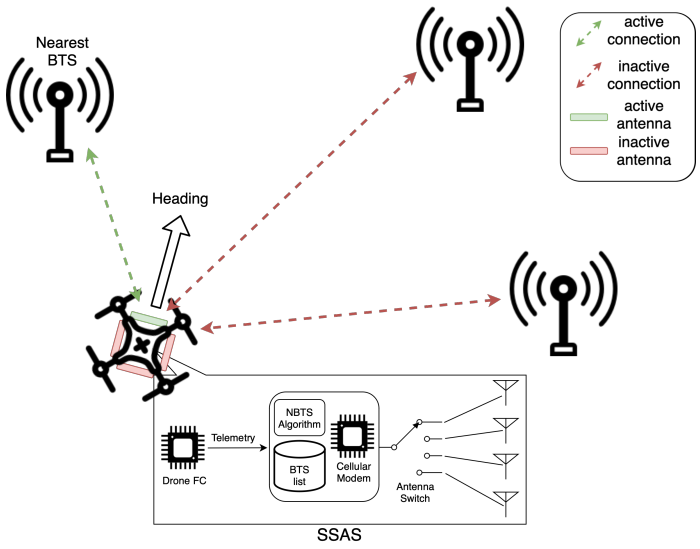


Figure 4.8: Scheme of NBTS algorithm

every 500 ms, to avoid a fast antenna switching caused by fast UAV's yaw movements affecting the system's stability.

Connected BTS (CBTS) Antenna Selection Algorithm

Similarly to the NBTS algorithm detailed in Subsection 4.3.2, the Connected BTS (CBTS) selection algorithm takes into account, as input parameters: (i) the GNSS location of the UAV; (ii) its heading angle; and (iii) the GNSS information of the BTS the LTE modem is connected to. As for the NBTS algorithm, an initial temporary BTS selection is performed to consider only those BTSs located in a 20 km nearby radius. Then, as shown in Figure 4.9, as soon as the initial cellular network scan is performed, the LTE evolved NodeB (eNB) is retrieved from the collected Cell IDs, with the SSAS trying to find the BTS with the same eNB in its internal temporary BTSs list:

- if the CBTS algorithm does not find a suitable BTS with the same eNB, then the NBTS algorithm kicks in for this iteration;

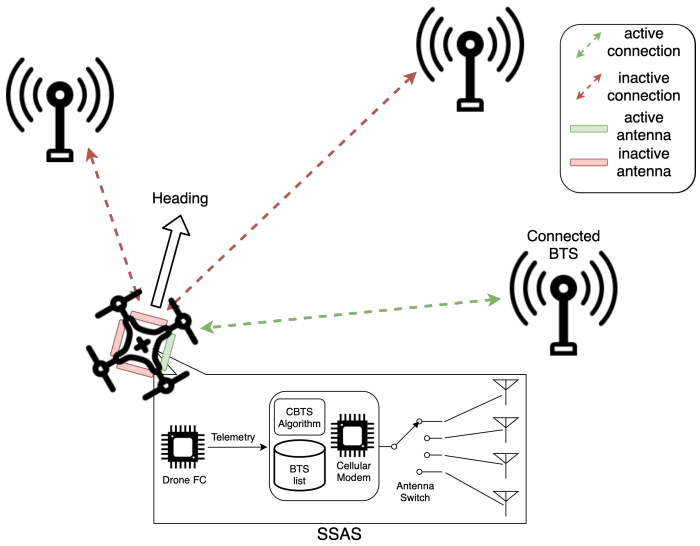


Figure 4.9: Scheme of CBTS algorithm

- if the CBTS algorithm finds the BTS with the same eNB, then the BTS's GNSS coordinates and data are retrieved, allowing to compute the geographical distance and the bearing angle with respect to the UAV position.

Once the target BTS has been identified, using the drone's heading angle and the computed bearing angle of the connected BTS, it is possible to select the proper 2×2 MIMO antenna pointing toward the BTS the on-board cellular modem is connected to, while the other antennas will be turned *off*. Finally, as for the NBTS algorithm, a check on the UAV's heading and position is performed every 500 ms.

4.4 Testing Methodology

In order to evaluate the performance of the proposed SSAS in comparison with those of the traditional omni-directional antennas provided by Sierra Wireless and used for the preliminary measurements presented in Section 4.2, a specific testing methodology has been opportunely defined taking into account several key aspects. In de-

tail, all the experimental measurements have been collected in a known environment with a predefined flight path pre-loaded into the UAV's FC, as illustrated in Subsection 4.4.1. Then, in order to collect comparable data across several flights and with different antennas and antennas control algorithms, the *on-board* cellular modem's parameters have been opportunely specified, with the aim to gather the essential radio quality metrics discussed in Subsection 4.4.2.

4.4.1 Testing Environment and Flight Path

The experimental flight tests have been carried out in the Po Valley near Sabbioneta, Mantova, Italy, in a rural area featuring no buildings or houses, in order to (i) comply with the EASA regulations and (ii) keep an adequate safety level for both the pilot and the drone during the tests. In particular, this is required since the Maximum Take Off Mass (MTOM) of the UAV (including the SSAS) is around 6 kg, thus not suitable for urban flights over crowded or populated area, where, instead, special permissions and additional safety measures are needed. Moreover, the testing environment features a completely flat terrain—no hills are still present—and allows a direct visibility of tens of BTSs even at a great distance (15+ km). This leads, as shown in the satellite view in Figure 4.10, to an environment suitable for verifying how the proposed SSAS behaves, given the large number of nearby cells available introducing inter-cell interference for the flying drone.

With regard to the flight path to be followed by the UAV during the experimental evaluations, it has been defined and designed targeting the following goals:

- allow repeatability in the experimental evaluations, in the way that the drone could easily fly by itself following this predefined and pre-loaded flight path;

- maximize the added value of the experimental evaluations, focusing on the goals of verifying how the NBTS and CBTS antennas control algorithms handle the alignment between the UAV's heading and the nearby BTSs, therefore on how switching *on* and *off* the selective and directional antennas pointing toward the specific BTS will be performed.

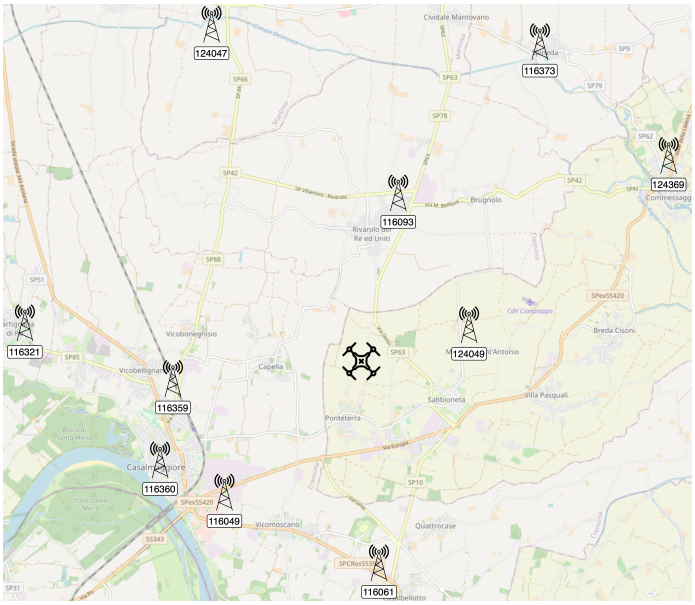


Figure 4.10: Map of the test environment with all the BTS's eNBs detected by the UAV

More in detail, the reference flight path requires the UAV to follow a circular path composed by multiple flight points. Then, during the flight, the drone always keeps its head aligned with the line tangent to the perimeter of the loop, performing a complete 360° yaw turn during the whole flight and allowing to verify if the proposed algorithms effectively turn *on* and *off* the antennas during the tests. Then, to further verify the behavior of the proposed SSAS at different flight heights, the flight path has been enhanced, requiring the UAV to perform multiple overlapped loops at different heights, ranging from 20 m AGL (the lowest one) up to 100 m AGL, with a 20 m height spacing between them. For the sake of completeness, the flight path composed by five overlapped ascending loops at different heights, with 150 measurement points for each height level, is shown in Figure 4.11.

Moreover, the following operating aspects have been considered during the flights:

- the UAV operational mode has been set to `AUTO` mode, thus leaving the drone

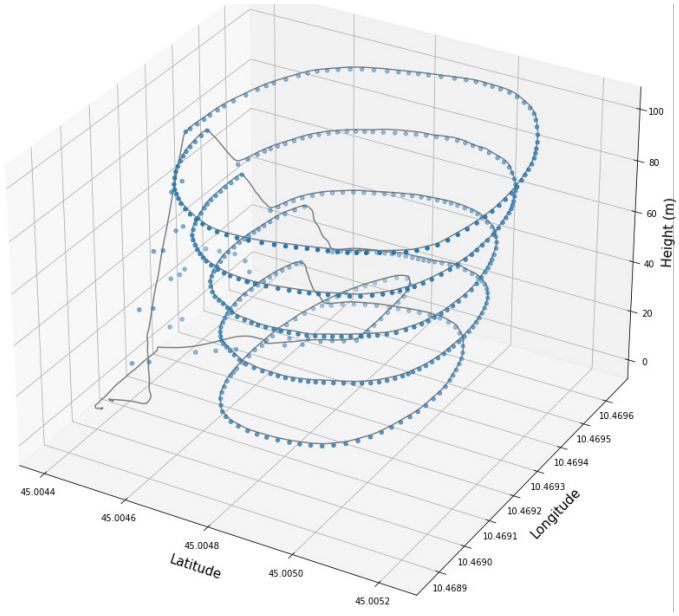


Figure 4.11: Flight path and measurement point exploited for the data collection and flight tests.

flying autonomously following the predefined way-points;

- the maximum horizontal flying speed has been set equal to 3 m/s, while the yaw speed has been set to `AUTO`;
- for safety reasons, ascending and descending speed has been set to a maximum value equal to 2 m/s.

During the flights, as it will be further explained in Subsection 4.4.2, the data related to LTE network signals quality indexes, BTSs' parameters, antennas' selection choice, UAV telemetry, and other FC data, have been stored and saved in the aerial MIG mounted on the drone, in order to create a complete dataset with multiple flights records to be evaluated at the end of the flights. The same flight path has also been used to collect all the aforementioned metrics using the stock omni-directional antennas provided with the modem, with the aim to experimentally compare ed evaluate

how they perform with respect of the proposed SSAS in terms of both signal quality and network stability.

4.4.2 Network Parameters and Gathered Data

In order to speed-up the development and evaluation of the SSAS, some network features have been limited due to technical reasons, as specified in the following. With regard to the cellular network compatibility, the first prototypical implementation of the SSAS is mainly suitable only for 4G LTE networks, since the used antennas are compatible only with the main LTE bands used by Italian cellular Internet Service Providers (ISPs), namely the LTE Band 3 and Band 7, in this way excluding others bands used in Italy, like the 800 MHz band (Band 20: $791 \div 862$ MHz). Moreover, the LTE Band 20 has been also omitted for the following reasons:

1. Band 20 is known to have limited throughput, since its useful bandwidth is limited to around 30 MHz, split between three operators (10 MHz each). This limits the maximum theoretical throughput to 75 Mbps, lower with respect to the 150 Mbps throughput achievable on Band 3 and Band 7.

2. Band 20 is mainly used in rural areas for indoor coverage, given the better penetration of the lower carrier frequency. However, since the SSAS mainly targets outdoor use cases, mostly in direct VLOS conditions between the 4G LTE cellular modem and the BTS, the Band 20 can be neglected in favor of the most suitable (and performing) Band 3 and Band 7.

On the practical side, the SSAS has been designed to satisfy the requirements for LTE Cat. 4 networks, so, as detailed in Subsection 4.3.1, it features only 2×2 MIMO channels, therefore only two antennas' connectors on the EM9191 modem has been exploited (namely, `MAIN` and `AUX` connectors), while the `MIMO1` and `MIMO2` connectors have been left unconnected. Moreover, to avoid bandwidth aggregation being not supported on all the BTSs, the Carrier Aggregation (CA) feature in the EM9191 modem has been disabled, in this way easing the tests repeatability. Then, in order to gain from more uniform and comparable data collected using both directional (those

part of the SSAS) and omni-directional (e.g., commercial) antennas, the LTE modem has been set to use only the bands 3 and 7, as well as it has been forced to use only 4G LTE networks, thus excluding the use of backward compatible networks (e.g., 2G/3G cellular networks) sharing some bands in the antennas' frequency operating range.

Despite the compatibility of the adopted EM9191 modem with sub-6GHz 5G networks, experimental evaluations exploiting new generation cellular networks have been omitted on the first SSAS prototype mainly for two reasons:

1. 5G networks have not been deployed yet in rural area where the tests have been performed;

2. the widely used 5G bands, namely Band N78 (3.7 GHz) and Band N28 (700 MHz), are outside of the operating frequency of the adopted antennas.

As anticipated in Subsection 4.4.1, several radio metrics data have been collected and stored inside JSON files during the experimental flights for a final results evaluation. In detail, the following network parameters indexes (defined by the 3GPP [156] and detail in Section 4.2) have been collected every 500 ms: (i) RSSI (mean value, `MAIN` antenna, `AUX` antenna); (ii) SINR; (iii) RSRP (mean value, `MAIN` antenna, `AUX` antenna); (iv) RSRQ; (v) LTE Band; (vi) Cell ID; (vii) eNB.

While the SINR, RSRP, RSRQ and RSSI have already been described in Section 4.2, the LTE Band represents the specific frequency range (regulated by authorities and used by mobile network operators) which 4G LTE signals are transmitted and received within; the Cell ID represents a unique identifier assigned to each cell in a mobile network end, allowing end devices to recognize and connect to specific cells, while, in the end, the eNB is the key component of 4G LTE BTSs connecting end terminal devices to the core network.

Finally, it is noteworthy to clarify that: (i) the aforementioned parameters are measured and obtained on the cellular modem located on the UAV, thus being related to the DL communication link (i.e., from the BTS to the UAV), since without the access to the eNB it is not possible to measure the same parameters on the BTS side, therefore not allowing to evaluate the UL (i.e., from the UAV to the BTS); (ii) data

have been associated with the UAV's telemetry data (e.g., GNSS coordinates, altitude information, and other essential data) collected in the same time frame.

4.5 Experimental Performance Evaluation

On the basis of the testing methodology detailed in Section 4.4, in the following an experimental performance evaluation of the proposed SSAS is presented and discussed, especially in terms of the network parameters considered in Subsection 4.4.2.

More in detail, in order to better compare the obtained information, the same amount of flights have been performed (namely 6) for each system setup of interest, leading to 6 flights with the omni-directional antennas (denoted, for the sake of comparison, as OBTS), 6 flights with the SSAS running the NBTS algorithm as antennas control algorithm, and 5 flights with the SSAS running the CBTS algorithm as antennas control algorithm. After the flights, a data pre-processing and cleaning has been performed, considering as "valid" only those data associated with valid telemetry information (e.g., a number of satellites greater or equal to 6 satellites; valid flight altitude; same LTE band for all the radio data). Finally, the data have been clustered according to (i) their collection flight height, which is considered as a key parameter, since it determines the behavior of the cellular network and the overall link quality; and (ii) their GNSS coordinates, aiming at a point-to-point comparison between different experimental flights setups.

In order to evaluate how the antennas switching mechanisms and the flight height affect the LTE cellular network radio signal behaviour during the whole flights (e.g., signal stability and communication link quality), an analysis of the impact of radio and SSAS parameters along the flight path is discussed in Subsection 4.5.1. Then, an analysis of the radio parameters by clustering the collected data into the five tested flight altitudes is detailed in Subsection 4.5.2, while a comparison and evaluation of the impact of the number of eNB changes on the different tested setups and at the various flight heights is performed in Subsection 4.5.3.

4.5.1 Impact of the SSAS Control Algorithms

The first signal behavior analysis has been carried out on the whole flight path followed by the UAV, aiming at showing how the chosen radio parameters (namely: SINR, RSRQ, RSRP, RSSI, and eNB) evolve as a function of the flight height, and of the heading angle of the UAV.

The impact of the default omni-directional antennas connected to the EM9191 cellular modem located on board the UAV is shown in Figure 4.12. It is possible to observe that both SINR and RSRQ initially increase and then decrease with the altitude of the drone, while both RSRP and RSSI drastically increase as soon as the UAV flies a few meters AGL, thus confirming the data collected in the preliminary flights of Section 4.2. Then, it can be noticed that, during the flights, SINR, RSRQ, RSRP, and RSSI oscillate when the cellular modem connects to a different cell (as represented by the eNB curve). Moreover, while there seems to be a correlation between UAV flight altitude and radio signal quality parameters, no clear correlations can be found between the drone's heading angle and the eNB value: when the drone has the same heading (no matter of the flight height), the eNB values are often different. The same holds for the radio signal indexes, whose behaviour does not depend on the UAV heading angle, as expected by a system equipped with omni-directional antennas.

The experimental results obtained through the SSAS running the NBTS antennas selection algorithm are shown in Figure 4.13. It can be seen how the overall behaviour of SINR, RSRQ, RSRP, and RSSI is similar to that obtained with the omni-directional antennas (as a function of the UAV flight height), but with the values of the RSRP and RSSI being significantly higher with respect to those shown in Figure 4.12, while both SINR and RSRQ being marginally higher then the omni-directional antennas setup (as further discussed in Subsection 4.5.2). Moreover, exploiting (in Figure 4.13) the presence of the identifier of the chosen antenna (among the four available on the SSAS), and looking at both the chosen antenna and the heading angle, it can be noticed a clear correlation between them, thanks to the fact that the NBTS algorithm is keeping enabled the antenna aligned with the nearest BTS while the UAV performs its flight mission, slowly rotating on the yaw axis at each flight altitude. Therefore,

Figure 4.12: Omni-directional antennas LTE radio signal behaviour along the predefined flight path.

the selected antenna plot present a repetitive pattern correlated to the heading angle, thus confirm that the system is properly working. However, the plots also show that, despite directional antenna reduced beamwidth and the NBTS algorithm keeping enabled the proper antenna, there are several cell changes, as returned by the eNB curve. To this end, one might expect the system to be able to maintain the connection with the nearest cell without changing to others nearby cells; however, given the significantly higher signal gain of the directional antennas, sometimes the system connects to other cells which are in the same direction where the selected antenna is pointing to, but often farther away than the nearest one, thus explaining the lower radio parameters values (e.g., as visible at timestamps between 290 and 320). Moreover, if compared to the omni-directional antennas system, in Figure 4.13 some signal drops can be noticed (e.g., at timestamps 137, 182, 236, 311, 337, etc.) when the SSAS switches from an antenna to another one—this behaviour will be further discussed in Section 4.5. This will also explain the reason why, despite the average value being better on the SSAS, the absolute minimum values are worse on the SSAS with respect to the omni-directional system, as discussed in Subsection 4.5.2.

Finally, with regard to the evaluation of the proposed SSAS running the CBTS an-

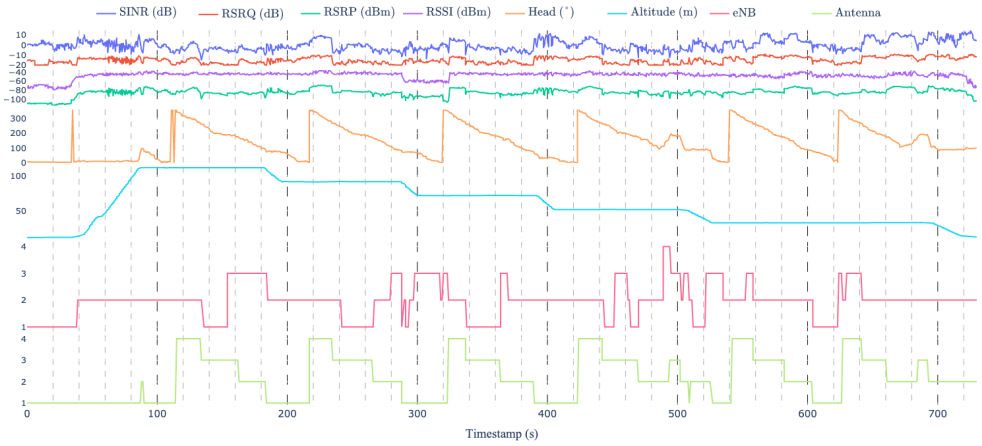


Figure 4.13: Directional antennas with NBTS algorithm LTE radio signal behaviour along the pre-defined flight path.

tennas control algorithm, the obtained experimental results are shown in Figure 4.14. In this case, it is possible to confirm the better RSRP and RSSI values with respect to the omni-directional setup, as well as the similar signal behavior associated with the flight height change seen in both Figure 4.13 and Figure 4.12. In fact, since the CBTS algorithm selects the antenna pointing toward the BTS the modem is connected with, there are no repetitive patterns and a clear correlation between the SSAS antenna identifier and UAV heading angle, while the same signal drops seen in Figure 4.13 happening at the same time the enabled antenna switch to another one can often be seen, thus confirming the main limit of the current SSAS implementation.

4.5.2 Impact of the Flight Altitude on Signal Quality

In order to better estimate the SSAS performance given the antennas control algorithms detailed in Subsection 4.3.2 and with respect to the omni-directional antennas-based solution, all the data gathered during the flights performed for each experimental setup have been merged in the same dataset (one dataset for each experimental setup), then removing the data collected before the test launch and during the altitude transitions to avoid to impact and skew the results. Finally, the obtained datasets

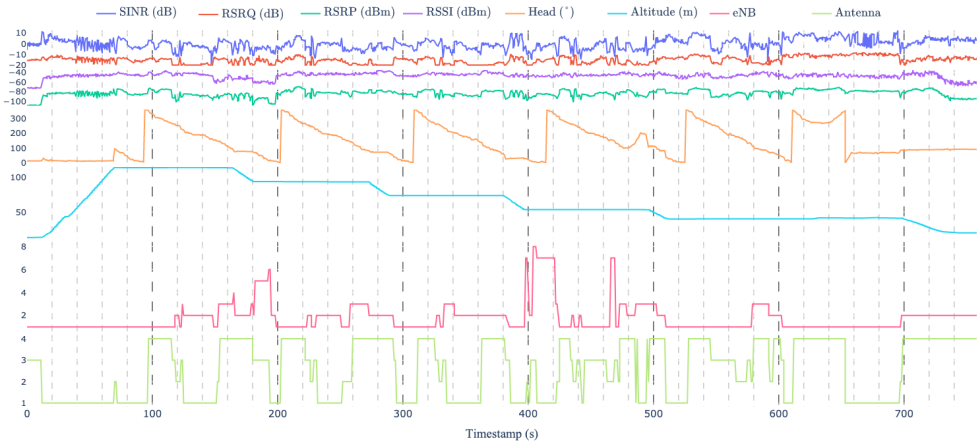


Figure 4.14: Directional antennas with CBTS algorithm LTE radio signal behaviour along the pre-defined flight path.

have been clustered *first* according to their heights (namely: 20 m, 40 m, 60 m, 80 m, 100 m) and *then* according to the GNSS coordinates of the measurement points. More in detail, as detailed in Subsection 4.4.1 and shown in Figure 4.11, 150 GNSS points have been identified and applied to the data clustering function, in this way enabling to perform (i) a results' comparison and analysis on the advantages and disadvantages of the proposed SSAS system with respect to the adoption of omni-directional antennas at different flight altitudes, and (ii) a *one-to-one* comparison between NBTS-, CBTS-, and OBTS-based antennas selection on the basis of the three experimental setups at the same height but also at the same GNSS point of the predefined flight path. Moreover, to ease the visualization and understanding of the gathered data, the CDF of SINR, RSRQ, RSRP and RSSI have been calculated at the five different flight heights. In the following, the main signal quality indexes among the considered experimental setups (NBTS, CBTS, OBTS) are investigated and discussed at 100 m, 80 m, 60 m, 40 m, and 20 m AGL flight heights.

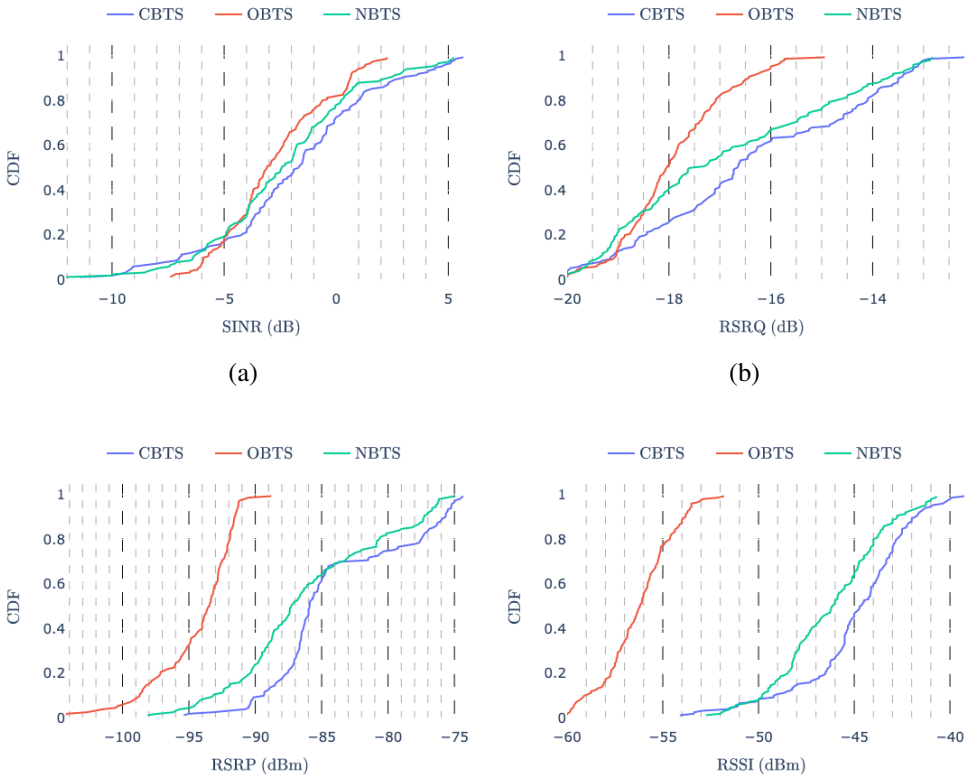


Figure 4.15: (a) CDF of the SINR among the considered experimental setups (NBTS, CBTS, OBTS) at a 100 m AGL flight height, (b) CDF of the RSRQ among the considered experimental setups (NBTS, CBTS, OBTS) at a 100 m AGL flight height, (c) CDF of the RSRP among the considered experimental setups (NBTS, CBTS, OBTS) at a 100 m AGL flight height and (d) CDF of the RSSI CBTS algorithm, NBTS algorithm and omni-directional antennas at 100 m.

Experimental Results at a 100 m AGL Flight Height

The CDFs of the SINR, RSRQ, RSRP and RSSI quality indexes are shown in Figure 4.15(a), Figure 4.15(b), Figure 4.15(c) and Figure 4.15(d), respectively.

For the sake or completeness, the minimum, maximum, and mean values for all

Table 4.1: SINR and RSRQ measurement at a 100 m AGL flight altitude through the different experimental setups.

	SINR [dB]			RSRQ [dB]		
	mean	max	min	mean	max	min
CBTS	−1.45	6.95	−11.45	−15.82	−12.35	−20.0
NBTS	−1.97	7.0	−9.85	−16.32	−10.0	−20.0
OBTS	−2.62	3.15	−7.16	−17.87	−14.78	−20.0

the parameters are listed in Table 4.1 and Table 4.2, whose values at a 100 m AGL flight height clearly show an average improvement of RSRP and RSSI when using the SSAS with both CBTS and NBTS algorithms (with respect to omni-directional antennas). In detail, both the average RSRP and RSSI are almost 10 dBm higher than the omni-directional antennas values. Regarding the average SINR, both the CBTS and NBTS manage to achieve better values, however, for the NBTS the improvement is limited to less than 0.7 dB, while the CBTS manage to achieve a 1.2 dB gain. The same applies to the RSRQ, with the CBTS achieving a gain of almost 2 dB, while the NBTS gain is just 1.5 dB. With regard to the SINR, it must be noticed that the minimum observed values while using the SSAS are lower than with the omni-directional antennas. This can be motivated by signal drops due to the antenna switching, underlined in Subsection 4.3.2 and further discussed in Subsection 4.5. On the overall, from a radio signal quality perspective, at a 100 m AGL flight height the best solution seems to be the adoption of a CBTS-controlled SSAS that, on average, returns better radio performance than if controlled through the NBTS algorithm.

Experimental Results at a 80 m AGL Flight Height

The CDFs of the SINR, RSRQ, RSRP and RSSI quality indexes are shown in Figure 4.16(a), Figure 4.16(b), Figure 4.16(c) and Figure 4.16(a), respectively.

Analyzing the obtained results, at a 80 m AGL flight height there is a clear advantage for the SSAS system regarding the RSRP and RSSI, whose values are (on average) 10 dBm and 6 dBm greater, respectively, with the NBTS and CBTS algo-

Table 4.2: RSSI and RSRP measurement at a 100 m AGL flight altitude through the different experimental setups.

	RSSI [dBm]			RSRP [dBm]		
	mean	max	min	mean	max	min
CBTS	−44.86	−39.38	−54.0	−84.2	−73.75	−95.38
NBTS	−46.06	−39.25	−52.88	−85.85	−74.0	−97.38
OBTS	−56.24	−52.25	−60.08	−94.33	−89.25	−105.25

Table 4.3: SINR and RSRQ measurement at an 80 m AGL flight height through the different experimental setups.

	SINR [dB]			RSRQ [dB]		
	mean	max	min	mean	max	min
CBTS	−2.88	5.87	−10.6	−16.93	−10.47	−20.0
NBTS	0.35	9.4	−9.3	−15.01	−10.25	−20.0
OBTS	−2.26	3.6	−7.68	−17.74	−15.0	−20.0

rithms, with respect to the use of omni-directional antennas. In this case, the average SINR and RSRQ improvement for the NBTS system is more than 2 dB higher with respect to the CBTS and omni-directional antennas values, as returned by the CDF shown in Figure 4.16(a). At a 80 m AGL flight altitude, the NBTS algorithm obtains the best performance gains, as shown in Table 4.3 and in Table 4.4, in which, for the sake or completeness, the minimum, maximum, and mean values of all the parameters are listed.

Experimental Results at a 60 m AGL Flight Height

The CDFs of the SINR, RSRQ, RSRP and RSSI quality indexes are shown in Figure 4.17(a), Figure 4.17(b), Figure 4.17(c) and Figure 4.17(d), respectively.

At a 60 m AGL flight height, the performance gains at the radio level reduce when comparing both the SSAS algorithms (namely, NBTS and CBTS) with the

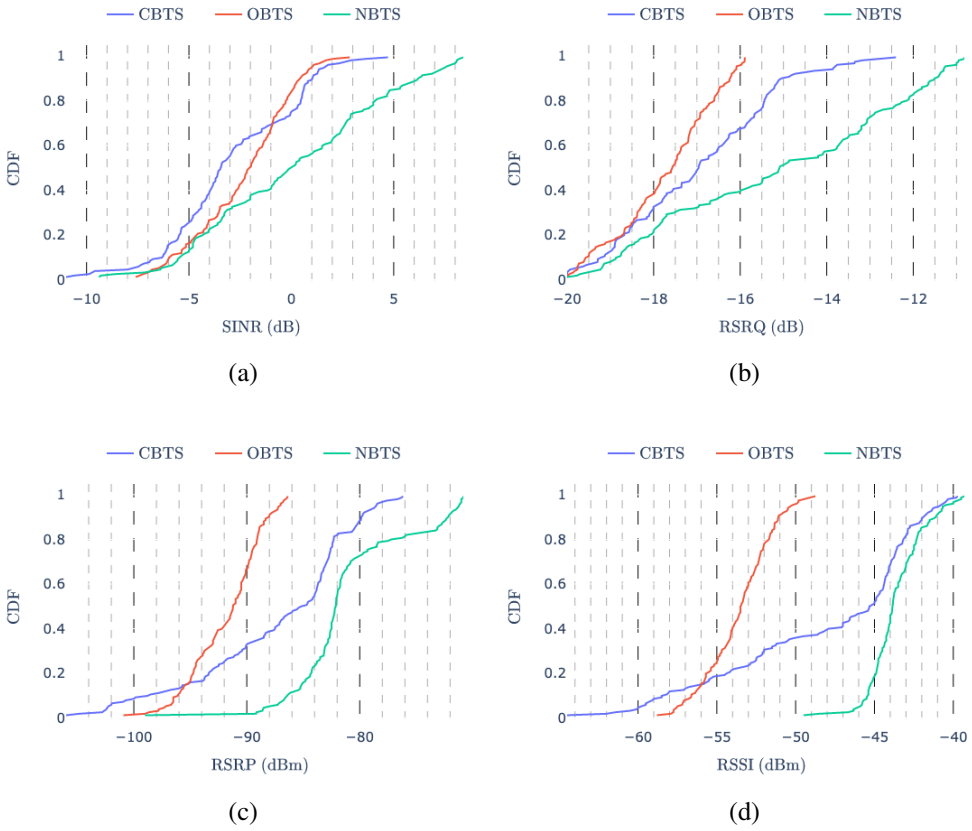


Figure 4.16: (a) CDF of the SINR among the considered experimental setups (NBTS, CBTS, OBTS) at a 80 m AGL flight height, (b) CDF of the RSRQ among the considered experimental setups (NBTS, CBTS, OBTS) at a 80 m AGL flight height, (c) CDF of the RSRP among the considered experimental setups (NBTS, CBTS, OBTS) at a 80 m AGL flight height and (d) CDF of the RSSI CBTS algorithm, NBTS algorithm and omni-directional antennas at 80 m.

omni-directional-based system. In detail, the overall gains of the RSRP and RSSI are within 9 dBm and 5 dBm in favor of the SSAS system, with the NBTS solution achieving the best performance. Therefore, in this case the best SSAS algorithm is the NBTS, which is more stable than the NBTS. Regarding SINR and RSRQ, the

Table 4.4: RSSI and RSRP measurement at an 80 m AGL flight height through the different experimental setups.

	RSSI [dBm]			RSRP [dBm]		
	mean	max	min	mean	max	min
CBTS	−47.96	−39.0	−64.5	−87.18	−71.0	−105.5
NBTS	−43.57	−38.75	−49.5	−80.77	−69.0	−99.0
OBTS	−53.5	−47.11	−58.8	−91.72	−84.67	−100.92

Table 4.5: SINR and RSRQ measurement at a 60 m AGL flight height through the different experimental setups.

	SINR [dB]			RSRQ [dB]		
	mean	max	min	mean	max	min
CBTS	−1.87	7.4	−7.14	−16.82	−12.1	−19.95
NBTS	−1.85	5.9	−13.2	−16.24	−12.15	−20.0
OBTS	−2.27	2.96	−7.58	−17.95	−15.86	−20.0

Table 4.6: RSSI and RSRP measurement at a 60 m AGL flight height through the different experimental setups.

	RSSI [dBm]			RSRP [dBm]		
	mean	max	min	mean	max	min
CBTS	−44.1	−38.0	−48.58	−83.09	−73.25	−91.4
NBTS	−47.55	−42.0	−55.58	−86.29	−74.5	−103.5
OBTS	−53.16	−47.8	−58.67	−91.62	−86.81	−97.42

SINR difference between the three setups is minimal, with a small 1.5 dB gain of the mean value of the RSRQ in favor of the NBTS, as confirmed by the CDF shown in Figure 4.17(b). For the sake or completeness, the minimum, maximum, and mean values for all the network parameters are listed in Table 4.5 and in Table 4.6.

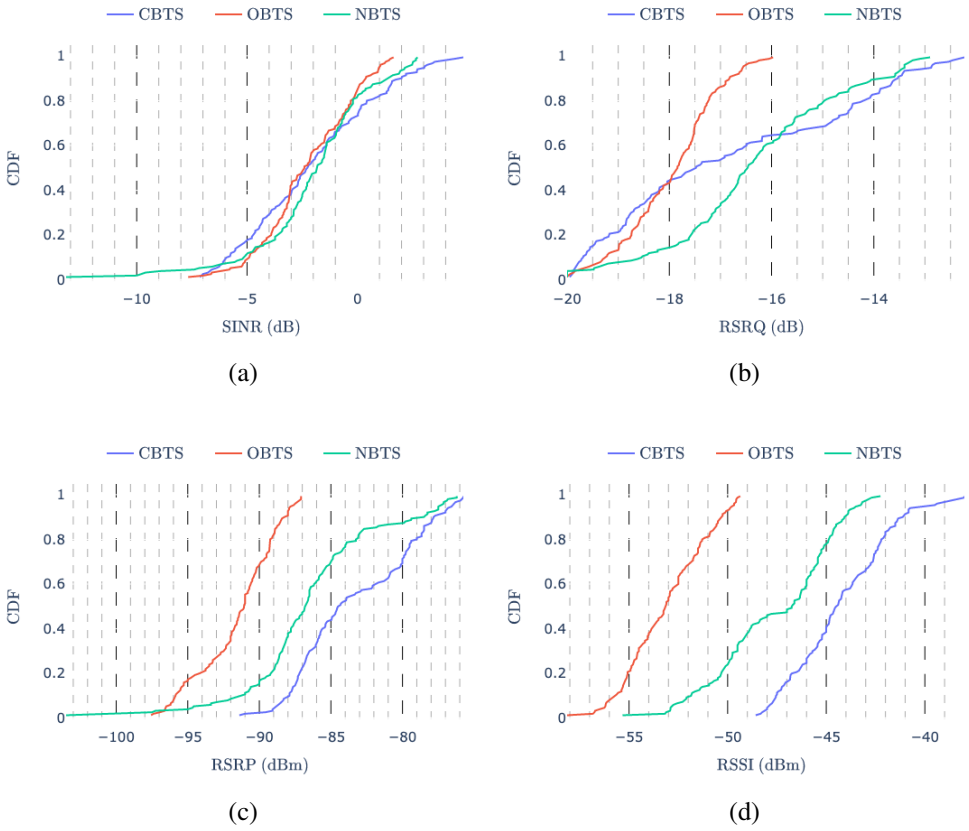


Figure 4.17: (a) CDF of the SINR among the considered experimental setups (NBTS, CBTS, OBTS) at a 60 m AGL flight height, (b) CDF of the RSRQ among the considered experimental setups (NBTS, CBTS, OBTS) at a 60 m AGL flight height, (c) CDF of the RSRP among the considered experimental setups (NBTS, CBTS, OBTS) at a 60 m AGL flight height and (d) CDF of the RSSI CBTS algorithm, NBTS algorithm and omni-directional antennas at 60 m.

Experimental Results at a 40 m AGL Flight Height

The CDFs of the SINR, RSRQ, RSRP and RSSI quality indexes are shown in Figure 4.18(a), Figure 4.18(b), Figure 4.18(c) and Figure 4.18(d), respectively.

Lowering the flight height to 40 m AGL, the performance gains of the SSAS fur-

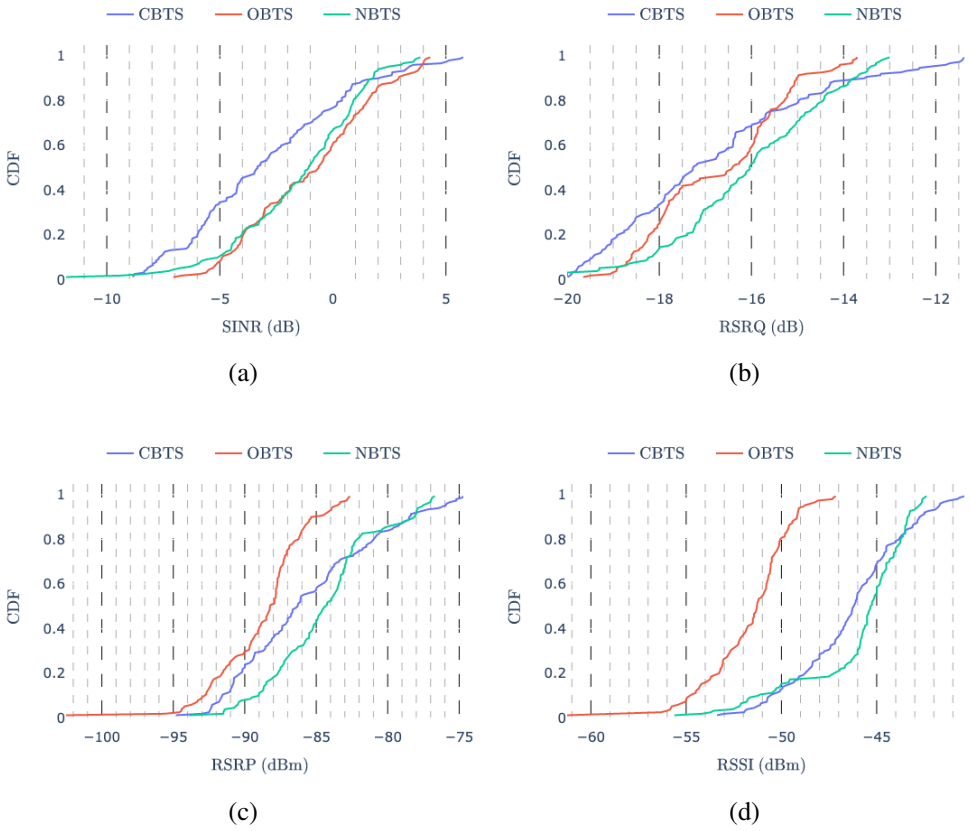


Figure 4.18: (a) CDF of the SINR among the considered experimental setups (NBTS, CBTS, OBTS) at a 40 m AGL flight height, (b) CDF of the RSRQ among the considered experimental setups (NBTS, CBTS, OBTS) at a 40 m AGL flight height, (c) CDF of the RSRP among the considered experimental setups (NBTS, CBTS, OBTS) at a 40 m AGL flight height and (d) CDF of the RSSI CBTS algorithm, NBTS algorithm and omni-directional antennas at 40 m.

ther reduces when comparing both the algorithms with the omni-directional-based system. In detail, the main advantages are related to RSRP and RSSI values, given the higher gains of the directional antennas of the SSAS, while the SINR is basically better with the omni-directional system. The same cannot hold for the RSRQ,

Table 4.7: SINR and RSRQ measurement at a 40 m AGL flight height through the different experimental setups.

	SINR [dB]			RSRQ [dB]		
	mean	max	min	mean	max	min
CBTS	−2.86	5.96	−8.93	−16.77	−11.2	−19.9
NBTS	−1.56	4.13	−12.2	−16.06	−12.92	−20.0
OBTS	−0.96	4.9	−7.05	−16.67	−13.4	−19.65

Table 4.8: RSSI and RSRP measurement at a 40 m AGL flight height through the different experimental setups.

	RSSI [dBm]			RSRP [dBm]		
	mean	max	min	mean	max	min
CBTS	−46.2	−38.0	−52.69	−85.58	−74.5	−94.83
NBTS	−46.12	−41.67	−57.12	−84.53	−76.7	−94.0
OBTS	−51.68	−46.5	−61.25	−88.75	−82.5	−102.5

which is slightly better on the SSAS controlled by the NBTS algorithm. The overall gains of the RSRP and RSSI are within 5 dBm and 3 dBm in favor of the SSAS system, with the NBTS solution outperforming the CBTS. For the sake of completeness, minimum, maximum, and mean values for all the networks parameters are listed in Table 4.7 and in Table 4.8.

Experimental Results at 20 m AGL Flight Height

The CDFs of SINR, RSRQ, RSRP and RSSI quality indexes are shown in Figure 4.19(a), Figure 4.19(b), Figure 4.19(c) and Figure 4.19(d).

Finally, at a 20 m AGL flight height, the behaviour of the SSAS is pretty similar to that experienced at 40 m AGL, with some relevant gains only for RSRP, RSSI and RSRQ values, while the SINR (shown in the CDF depicted in Figure 4.19(a)) does not return any advantage of the SSAS with respect of the omni-directional-based

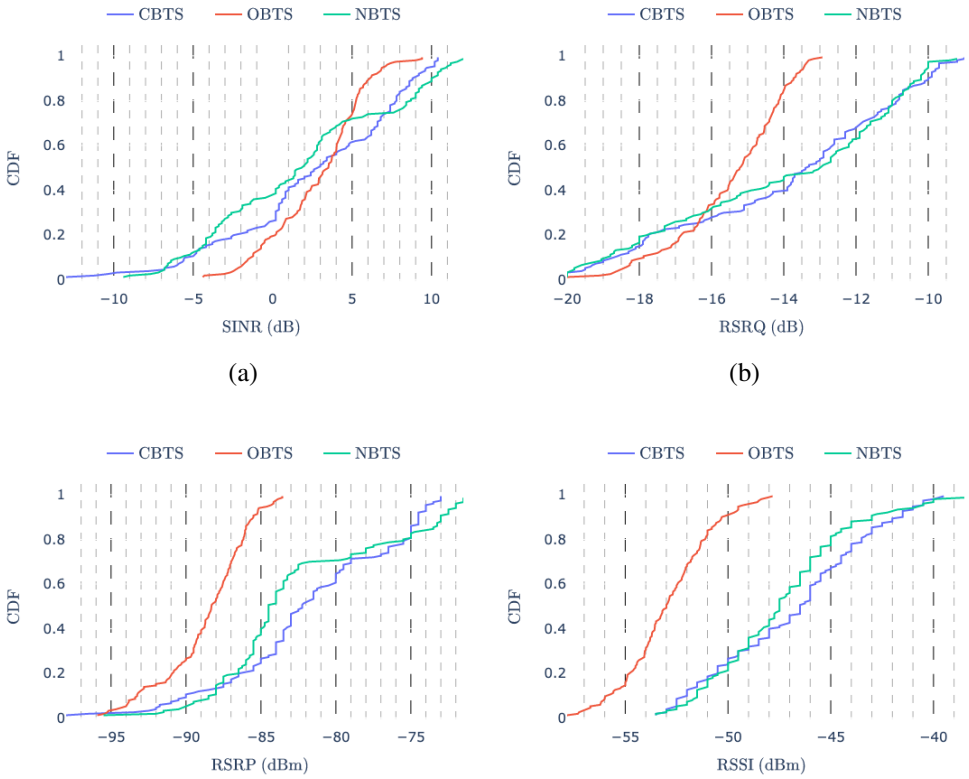


Figure 4.19: (a) CDF of the SINR among the considered experimental setups (NBTS, CBTS, OBTS) at a 20 m AGL flight height, (b) CDF of the RSRQ among the considered experimental setups (NBTS, CBTS, OBTS) at a 20 m AGL flight height, (c) CDF of the RSRP among the considered experimental setups (NBTS, CBTS, OBTS) at a 20 m AGL flight height and (d) CDF of the RSSI CBTS algorithm, NBTS algorithm and omni-directional antennas at 20 m.

system. The gains of the RSRP and RSSI are within 7 dBm and 5 dBm in favor of the SSAS system, with the CBTS solution achieving the best performance. The RSRQ shows similar values between CBTS and NBTS solutions, both achieving an average gain close to 1.5 dB with respect to the omni-directional solution. For the

Table 4.9: SINR and RSRQ measurement at a 20 m AGL flight height through the different experimental setups.

	SINR [dB]			RSRQ [dB]		
	mean	max	min	mean	max	min
CBTS	2.63	11.2	−13.0	−13.85	−8.9	−20.0
NBTS	1.94	12.0	−9.4	−13.99	−8.8	−20.0
OBTS	2.9	11.5	−4.4	−15.43	−11.95	−20.0

Table 4.10: RSSI and RSRP measurement at a 20 m AGL flight height through the different experimental setups.

	RSSI [dBm]			RSRP [dBm]		
	mean	max	min	mean	max	min
CBTS	−46.72	−38.5	−53.5	−81.66	−72.0	−98.0
NBTS	−47.52	−38.5	−54.0	−82.53	−71.0	−94.5
OBTS	−52.58	−45.0	−57.83	−88.3	−79.5	−95.92

sake of completeness, the minimum, maximum, and mean values of all the network parameters are listed in Table 4.9 and in Table 4.10.

4.5.3 Impact on the Amount of Cells Changes

The last experimental performance evaluation, whose results are shown in Table 4.11, has been performed investigating the amount of network cells changes experienced during the experimental flights—6 flights for each setup—at the different AGL flight altitudes.

According to the results listed in Table 4.11, the proposed SSAS system suffers less cells changes when the UAV is at 100 m and 80 m AGL with respect to the omni-directional system, with the NBTS algorithm obtaining the better results (especially at a 80 m AGL flight altitude). At lower flight altitudes (at 60 m), the performance of the three experimental setups result in being similar, with a small advantage for

Table 4.11: Amount of network cells changes experienced during 6 experimental flights at different AGL flight height, for the considered experimental setups.

	100 m	80 m	60 m	40 m	20 m
CBTS	36	34	42	63	47
NBTS	32	22	34	48	54
OBTS	49	44	38	31	27

the NBTS-based solution. Instead, at 40 m and 20 m AGL, the omni-directional antennas manage to achieve a relevant smaller number of cell changes with respect to the SSAS. In fact, these results reflect those related to the radio quality indexes discussed in Subsection 4.5.2, where the SSAS achieved better performance at higher flight altitudes when the drone is in VLOS condition with more BTSs, thus experiencing additional interference with omni-directional antennas that, instead, can be reduced by the proposed SSAS. On the other side, at lower flight heights, the omni-directional-based solution achieves better results, since more complex factors affect the performance of the cellular network (e.g., signal shadowing and reflections due to trees, buildings and other possible sources of noise).

4.6 Future Improvements

In order to improve the performance of the proposed SSAS and overcome the drawbacks of the first prototypical version, different improvements can be defined and applied to the system, on both hardware and antennas control algorithm layers. In the following, some possible future research topics aiming at enhancing reliability and performance of the SSAS for complex BVLOS applications are detailed.

4.6.1 Improved Switching Network

As highlighted by the experimental evaluation detailed in Section 4.5, one of the main limits of the current prototypical SSAS implementation refers to the antennas switching network, in detail allowing us to use only one 2×2 MIMO antenna at time. This

sometimes leads to significant signal drops when the SSAS *hard* switches from one antenna to another. Therefore, in a more advanced version of the system, the switching network might allow the use of multiple antennas at the same time, thus providing the ability to perform a *soft* switch between the antennas. As an example, in the case the SSAS system has a particular antenna enabled (i.e., the antenna #1) and, given the UAV heading rotation with respect to the target BTS, the SSAS has to switch to the contiguous antenna (i.e., the antenna #2), the improved version of the system might enable the antenna #2 a few seconds in advance before the need to switch to it, hence introducing a smoother transition without any signal drop. Moreover, this enhancement could also provide omni-directionality by enabling all the four antennas, for low altitude flights, where the current SSAS achieves weak performances if compared to the use of omni-directional antennas, therefore making the system more suitable to complete flight mission involving both high and low altitude flight operations.

4.6.2 Reduced SSAS Weight and Sizes

The currently adopted antennas are not suitable for all the UAVs, given their size and weight, as detailed in Subsection 4.3.1). However, targeting the possibility to integrate smaller drones with a more advanced deployment of the proposed SSAS system, thus benefiting of its improvements, then smaller and lighter antennas should be designed, as well as a more compact switching network. In the end, smaller components might benefit not only smaller drones, but also allow the integration of additional antennas on the UAV, thus opening to the possibility to more than 2×2 MIMO cellular communications (e.g., newer LTE equipment nowadays support up to 4x4 MIMO channels).

4.6.3 Support for 5G Cellular Networks

The current prototypical antennas are suitable for some the LTE bands adopted in Italy. However, the SSAS would allow, through the use of different antennas, the connectivity thought different bands and even the usage of the latest 5G NR cellular networks. Therefore, an enhanced version of the SSAS might feature newer

antennas able to support newer LTE and 5G bands, in this way opening the SSAS to next-generation cellular networks. As a consequence, the same might hold for others wireless protocols (e.g., Wi-Fi) that could benefit from a re-configurable antennas system such as the defined SSAS, especially for specific applications where cellular networks may not be involved.

4.6.4 Enhanced Antennas Control Algorithms

Finally, on the software side, newer and advanced antennas control algorithms might be designed and integrated in the SSAS, aiming at further optimizing and improving the connectivity performance of the system. As an example, a more robust control algorithm could exploit ML-based mechanisms to understand which BTS selection criteria is the most suitable according to various parameters, such as the flight altitude, the distance between the UAV and the BTSs, and, in a more complex algorithm, the physical modelling of signal propagation of the different bands, thus allowing a more robust usage of the SSAS according to the different application scenario.

4.7 Final Considerations

Looking at the results highlighted in Subsections 4.5.1–4.5.3, it can be concluded that, as expected, the proposed SSAS provides a signal quality improvement only at the higher flight altitudes (from 80 AGL to above flight altitudes) of the tested heights, where the cellular modem located on board the flying UAV has a higher probability to reach VLOS conditions with several BTSs using the same frequency (thanks to the frequency re-usage scheme of cellular networks shwon in Figure 4.2). Instead, at lower altitudes, the benefit of the SSAS is negligible, as below 60 m AGL the omni-directional antennas can achieve similar (and even better) results, especially at a 20 m AGL flight height, where the VLOS condition between the drone and the BTS is no more guaranteed due to taller obstacles (i.e., trees, buildings, etc.). This might cause problems in understanding which is the best serving BTS without knowing the cellular network planning scheme, thus omni-directional antennas are more suitable for operating application requiring cellular-enabled UAVs flying at smaller

flight altitudes. This also explains why, at these low level of altitudes (below 20 m AGL), the best performing SSAS algorithm is the CBTS. However, given the presence of obstacles, thus the effects of signal shadowing and reflections, not always the direct straight line between the UAV and the BTS position represents the real signal path, therefore afflicting the reliability and performance of the CBTS-controlled SSAS with respect to the omni-directional antennas.

Looking at the overall SSAS performance, the 2×2 MIMO directional antennas design of the proposed SSAS is able to provide significantly better RSRP and RSSI gains at every flight altitude, with respect to the omni-directional antennas, and their directivity combined with the proper control algorithm can improve the SINR and RSRQ at high altitudes. However, as discussed in Subsection 4.5.1, the switching board part of the proposed SSAS, combined with the non-overlapping beamwidth of the four antennas, sometimes could introduce signal quality drops when the SSAS switches from one antenna to another one, thus lowering the overall gained performance. Despite this drawback, which might be solved in a more advanced switching system discussed in Section 4.6, the SSAS can still decrease the overall number of cells changes along the designated flight path, especially at high flight altitude (where the system obtains the best results).

It is interesting to observe that, at most altitudes, the NBTS algorithm achieves better results than the CBTS. More in detail, at 100 m and 20 m AGL flight heights, the best performance are obtained by the CBTS algorithm applied to the SSAS, while in all the other cases, the NBTS algorithm outperforms the CBTS. This can be explained by the fact that, at different altitudes, there might be different cells signal arriving from nearby BTSs, as proved by [159]. Also, according to the FSPL signal propagation scheme, it is clear that in VLOS condition the NBTS strategy can offer better results, while at very low altitudes, where the VLOS is no more guarantee, the CBTS strategy is better than the NBTS.

Finally, it can be concluded that, although the system is not perfect and has room for several improvements, it can already increase the overall cellular connectivity adoption for UAVs without changing the existing cellular networks, therefore allowing to temporary use the actual 4G and even 5G networks while waiting the optimiza-

tion the next generation of cellular networks will bring with the aim to enable aerial connectivity for UAVs.

Chapter 5

Radio-based Localization

While the first part of this doctoral thesis discussed various approaches useful to provide UAV-to-X connectivity, in this way enabling the use of several wireless communication technologies and protocols in heterogeneous scenarios, this chapter (and Chapter 6) focuses on localization approaches in GNSS-denied contexts, in detail relying on different radio localization techniques and aiming at comparing various solutions with different accuracies. Moreover, the complexities and overall costs of these solutions are carefully taken into account, as they represent a relevant factor to be considered when dealing with the deployment of localization systems in large environments.

More in detail, in the first part of this chapter, the adoption of a COTS UWB *ready-to-use* solution applied to a small constrained UAV (for indoor localization) is investigated, aiming at evaluating the accuracy of the system for real-time localization in indoor environments. Then, looking for (i) a cost reduction (in terms of infrastructure) and (ii) the possibility to real-time localize a mobile UAV over a larger area, the adoption of Wi-Fi for localization purposes is investigated, highlighting viable Wi-Fi-based localization approaches, as well as their advantages and disadvantages with respect to other solutions, such as COTS UWB systems.

5.1 Overview

The radio technology applied to the topic of the localization has always been investigated under different facets since, given the widespread adoption of wireless connectivity, the possibility of exploiting already-available infrastructures for low-cost asset tracking purposes has always been associated with a demanding market [160]. To this end, efforts have been invested in the development of low complexity and affordable localization solutions, especially involving the utilization of existing Wi-Fi infrastructures (widespread in several industrial and public environments). However, while the literature overview carried out in Chapter 1 showed some results already suitable for IoT applications, where a position error on the order of several meters could be considered sufficient to roughly understand where an IoT device is located (e.g., inside a warehouse), the same cannot be applied to mobile UAVs or robotic platforms that, to properly operate, need a position estimation error close to half-a-meter, especially in critical BVLOS environments.

In order to achieve this goal, the *first part* of this chapter initially investigates the adoption of the UWB technology on a small and lightweight quad-copter (with a MTOM under 250 gr), suitable for indoor BVLOS flight missions in narrow and small environments. The detail regarding the testbed and implementation of the system on the UAV are discussed and presented, followed by an experimental evaluation involving real-time localization of the flying drone in a well-known environment, with several deployed UWB anchors. Then, after a few considerations about advantages and disadvantages of the UWB technology, an investigation on alternative localization techniques involving Wi-Fi is carried out, to extend the operational area where the UAV position can be estimated, thus significantly reducing the overall cost (with respect to UWB-based solutions).

In the *second part* of this chapter, Wi-Fi-based localization is considered for (i) traditional RSSI-based localization and (ii) relatively newer FTM-based localization techniques for a real mobile UAV target, in the end aiming at developing and providing a backup localization solution suitable to provide an approximate real-time position of the drone flying in a GNSS-denied environment. This will be beneficial to

help the pilot in remotely controlling the UAV flying in BVLOS conditions to understand in which part of the pre-defined flying path the UAV is operating. In order to reach this goal, a Wi-Fi-based localization architecture is designed and implemented on a real UAV to validate its performance and understand if the localization accuracy achievable with Wi-Fi FTM localization solutions, reported in the literature described in Chapter 1, could be achieved also in real-time on a mobile UAV, considering also the adoption of more affordable hardware operating on the less reliable 2.4 GHz ESP32S3-based APs.

The remainder of this chapter is organized as follows. In Section 5.2, the adoption of COTS UWB-based technologies for UAV localization in narrow environments is investigated experimentally, aiming at confirming or neglecting the performance achieved by the literature discussed in Chapter 1. In detail, the implemented system architecture, as well as the obtained experimental results, are discussed in Subsection 5.2.1 and Subsection 5.2.2, respectively. Section 5.3 investigates the adoption of Wi-Fi RSSI and FTM for the purpose of real-time UAV localization: Subsection 5.3.3 presents the principle behind the radio-based localization and illustrate the adopted algorithms; in Subsection 5.3.4 the adopted filtering strategy, exploited to reduce the effects of measurement and RF noise, is illustrated; in Subsection 5.3.5, the developed system is presented, discussing both the architecture and technical details, as well as its integration on a real UAV platform; Subsection 5.3.6 showcases the testing environment and the methodologies exploited to validate the localization system performance. Finally, improvements regarding the proposed system and future research directions are discussed in Section 5.3.7, while final remarks are provided in Section 5.3.8.

5.2 UWB-based Localization

UWB technology is a highly efficient wireless technology compliant with the IEEE 802.15.4-2011 standard [161] (at PHY and MAC layers), characterized by low power consumption and suitable to enable short-range data transmission and exceptionally precise localization. In detail, UWB can achieve data rates of several tens of Mbit/s

within a range of approximately 15 m and has the ability to provide centimeter-level location information between transmitting and a receiving units, at a typical distance between 10 m and 15 m. In fact, the precision of UWB in positioning and direction-finding overcomes that of more common technologies like Bluetooth and Wi-Fi, thus making it attractive for all those application involving high-accuracy localization requirements.

UWB operates across a broad frequency spectrum, ranging from 6 GHz to 8.5 GHz, with a channel bandwidth $B \geq 500$ MHz. Moreover, it relies on a impulse-based radio technology, emitting transient impulses of extremely short duration (on the order of ns). This ultra-wide frequency spectrum enables rapid data transmission via pulsed signals. Furthermore, the technology's broad bandwidth contributes to a low power spectral density, thereby minimizing interference with other technologies sharing the same frequency range. Finally, regulatory authorities (such as the ETSI [134]) have established a stringent limit of -41 dBm/MHz for spectral density, ensuring that UWB complies with established standards.

The advantageous combination of low power spectral density and pulse-based transmission of UWB technology not only reduces the potential for interference with coexisting technologies, but also enhances the security and reliability of UWB transmissions applied to the localization tasks, since the short duration of the UWB pulses reduces the likelihood of signal overlap in scenarios involving multiple reflections. In fact, in cases of signal reflections, the short pulse duration minimizes the possibility of the original pulse to overlap with its reflection. Moreover, this property enhances the robustness of UWB systems when faced with the challenging multi-path problem, provided that there exists a clear LOS communication path between transmitter and receiver. In fact, the LOS condition is pivotal in ensuring the resilience of UWB systems against multi-path effects, as it significantly diminishes the probability of conflicting signal interactions and facilitates reliable and accurate data transmission.

Given these characteristics, UWB systems are well known for their capacity to achieve remarkably accurate range resolution. In this context, range resolution is the system's ability to distinguish between two separate points (or objects) in space, based on their distances, and is determined by the pulse duration and bandwidth of the

UWB signal. In detail, the range resolution (denoted as *rr*) can be estimated as [162]

$$rr \approx \frac{v}{2B} \tag{5.1}$$

where: *rr* represents the attainable range resolution (dimension: [m]); *v* denotes the velocity of the signal (dimension: [m/s]), which can be approximated to the speed of light ($\approx 3 \cdot 10^8$ [m/s]); $B \geq 500$ MHz is the system's bandwidth (dimension: [Hz]).

The key factor contributing to high range resolution in UWB is the extremely short duration of its pulses, which allows to precisely measure the time it takes for a signal to travel to a target and back, denoted as ToF or Time of Arrival (ToA). Considering $B \geq 500$ MHz, this means a minimum $rr = 0.30$ m, while it is possible to reach $rr \approx 0.11$ m with $B = 1300$ MHz through some commercial UWB systems, in the end making UWB extremely advantageous in narrow indoor environments positioning applications. More in detail, the concept behind the ToF measurement adopted by UWB-based localization solutions relies on two separate approaches: (i) Two-Way Ranging (TWR) or (ii) Time Difference of Arrival (TDoA).

More in detail, as shown in Figure 5.1, the TWR approach relies on the computing of the distance between the transmitter and receiver by measuring the ToF of a UWB pulse. The process is initiated by the one of the two nodes, denoted as *tag*, which begins the measurement by dispatching a *Poll* message to the known address of the recipient, denoted as *anchor*. This time instant is known as Time of Sending Poll (TSP), denoted as $t_{p:tx}$. Upon receiving the *Poll* message, the anchor registers the Time of Reception of Poll (TRP, denoted as $t_{p:rx}$) and, subsequently, it responds with a reply message within a pre-defined delay (equal to $\tau_{p:rx:tx}$) at the *Poll* message, including both $t_{p:rx}$ and the Time of Sending Response (TSR, denoted as $t_{r:tx}$). At this point, the tag, after the reception of the *Response* message, records the Time of Response Reception (TRR, denoted as $t_{r:rx}$). Finally, it calculates τ_{tof} as follows:

$$\tau_{tof} = \frac{(t_{r:rx} - t_{p:tx}) - (t_{r:tx} - t_{p:rx})}{2} \cdot (1 - \varepsilon_{i,r}) \tag{5.2}$$

where $\varepsilon_{i,r}$ is a correction factor to compensate the clock drift on the two devices happening during the process. Then, knowing the speed of light, once τ_{tof} has been

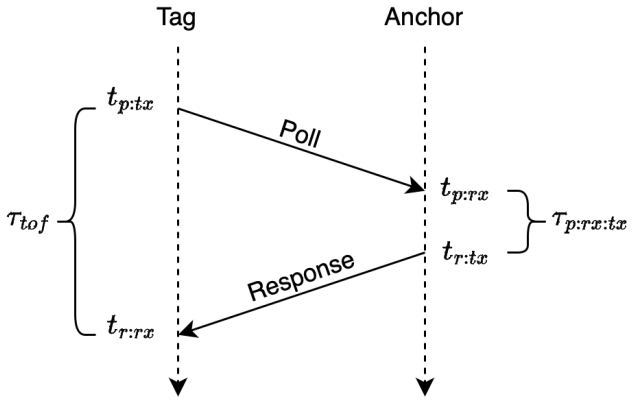


Figure 5.1: TWR scheme adopted in UWB localization.

calculated, it is possible to determine the distance d between the tag and the anchor as

$$d = \tau_{tof} \cdot v. \tag{5.3}$$

Once at least four τ_{tof} measurements from four different anchors $i \in \{1,\ldots,4\}$ are received, the estimated position can then computed by a multi-lateration algorithm (detailed in Subsection 5.3.3).

TDoA relies on a different approach, due to the fact that, in order to operate effectively, a precise synchronization among all the deployed anchors and tags is crucial, in this way ensuring that they share a common clock. In detail, the TDoA-based mechanism requires that the tag transmits signals at regular intervals through short *Blink* messages. These messages are received and processed by all anchors within its communication range. Then, regardless of their synchronization status, these anchors subsequently transmit all the message reception timestamps to a central server entity, denoted as Real-Time Location System (RTLS), in charge of accurately computing the position of the tag. More in detail, the RTLS server relies on timestamps received from at least four anchors sharing the same clock reference. Denoting the positions of the anchors as (x_i, y_i), where $i = \{1,2,\ldots,n\}$, and the tag's position (x,y), the

TDoA-based estimate for all possible anchor pairs can be expressed as

$$\text{TDoA}_{ij} = t_i - t_j, \quad \text{for } i,j = 1,2,\ldots,n, \text{ where } i \neq j \tag{5.4}$$

where: TDoA_{ij} is the TDoA between anchor i and anchor j; t_i (t_j) corresponds to the time instant at which the blink signal from the tag reaches anchor i (j); n is the total number of anchors. These TDoA values represent the time differences between all possible pairs of anchors and are used to determine the position of the tag using multi-lateration techniques, regardless of the number of anchors involved in the system.

Therefore, a key difference between these two localization techniques relies on the fact that TWR is less complex to implement (not requiring a clock synchronization between anchors, as instead required by TDoA). Also, TWR allows both a bi-directional communication between the two nodes and sharing the τ_{tof} measurement on both the devices, hence allowing *on-board* and *third-party* localization of the tag. However, given the longer processing time required by τ_{tof} and the need to repeat this measurement for at least four different anchors, this solution can be considered as (i) less effective for applications involving both a large number of tag and anchors, (ii) less energy efficient, given the large number of message exchanged, and (iii) less suitable for real-time localization application requiring extremely high update rates. On the other side, TDoA allows to track a higher amount of tags at the same time, as well as to reduce the power consumption, given the limited amount of exchanged messages, but requiring a much more complex and higher infrastructure cost, due to the synchronization requirement.

Looking at available COTS UWB solutions, the most effective development kits are the ones provided by Decawave [69] that, with a minimal effort infrastructure setup, can easily allow to adopt UWB technology for real-time localization purposes, as will be discussed in detail in Subsection 5.2.1.

5.2.1 System Architecture

In order to implement a COTS-based UWB localization solution on a sub-250 gr quad-copter, proper components have been adopted. More in detail, the UWB infrastructure, composed by 8 DWM1001 Decawave UWB anchors operating at 6.5 GHz

(a) (b)

Figure 5.2: Hardware used for the UWB localization test: (a) a single Decawave DWM1001 UWB anchor; and (b) quad-copter loading on board a DWM1001 UWB tag (connected to a RPi3).

and with a bandwidth $B = 500$ MHz, have been deployed in the environment and synchronized through the Android mobile application provided by Decawave. This mobile application also allows to setup the anchors' relative positions within the environment, thus prompting all the information needed to perform a UWB tag localization. For the sake of completeness, a Decawave DWM1001 UWB anchor is shown in Figure 5.2(a), while Figure 5.2(b) highlights the UWB tag (still based on a DWM1001) embedded on a RPi3 mounted on the GEPRC CineLog25 drone frame-based quad-copter [163], also providing the power to the SBC through the drone's battery. In the end, this system deployment features an overall weight under 250 gr—this is considered as a critical threshold, as defined by the EASA [21], to allow flight missions close to people.

Moreover, this setup allows several life hours to the deployed anchors, while the

UAV, considering both the power needed by the four engines and by the RPi3, has a flight time of 5 min, given the small size of the on-board 650 mAh LiPo battery capacity.

In order to localize the UAV position on the (x, y) horizontal plane, the UWB anchors have been deployed on the walls in the environment at a 2 m height, while the UAV has been manually flown at an approximately 1.5 m height. Since the goal of this experiment was to just verify and prove the reliability of a UWB COTS-based solution applied to real-time UAV localization in an indoor narrow environment, from an architectural point of view the system relies on the built-in multi-lateration proprietary model and filtering algorithms provided by Decawave [69], whose localization outputs are accessible through proper APIs allowing to retrieve the final position estimation of the *in-flight* UAV. More in detail, the adopted localization strategy has been the Reverse TDoA, based on the same working principle of the TDoA approach detailed in Subsection 5.2. However, instead of having the tag send the *Blink* messages and the anchors listen for them, the roles are inverted, thus having the deployed anchors sending several *Blink* messages according to a shared and synchronised clock received by the tag node embodied on the UAV, which then performs the multi-lateration and position estimation. For the sake of clarity, this is the same working principle adopted by GNSS systems [164].

Finally, this approach, combined together with conservative low energy consumption parameters set on UWB nodes, allows to obtain an accurate position estimation every second, by gathering the estimated position (thanks to the Decawave's APIs) through a Python script running on the RPi3, which in turn publishes the position to a remote cloud service through the RPi3's built-in Wi-Fi connectivity. Then, the cloud service is responsible for visualizing the updated positions on a properly designed map of the environment, at the end allowing to identify the position of the UAV in the environment, as discussed in Subsection 5.2.2.

5.2.2 Localization Results

Since in the test environment, which has an overall dimension of 3×20 m, no optical tracking system is available, the UAV has been flying according to the following

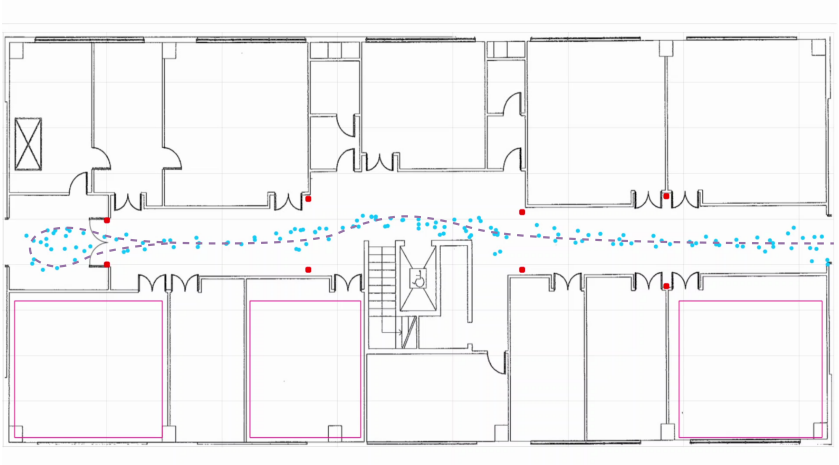


Figure 5.3: Experimental results of the UWB localization application on a sub-250 gr quad-copter. Red dots: represent the anchors positions; cyan dots: represent the estimated real-time position of the UAV; purple dashed line: is the ground truth trajectory followed by the UAV during the experimental flight.

well predefined flight path denoted by the purple dashed line shown in Figure 5.3: (i) the UAV start its flight from the middle of the door located on the right side of the building, (ii) flies through the corridor at a speed between 1 m/s and 4 m/s (avoiding the elevator's room), and (iii) finally, after performing a *U-turn* at the end of the building close to the door located in the left, goes back to the starting point. In this way, by observing the path followed by the UAV and the estimated position points, denoted by cyan points in Figure 5.3, it is possible to roughly determine the accuracy of the system in the tested environment. According to the results shown in Figure 5.3, it can be confirmed the good match between UWB-based estimated position of the UAV with respect to the ground truth trajectory, with 70% of the estimated positions within a 30 cm positioning error, thus confirming the *rr* of the adopted system operating with $B = 500$ MHz.

The only positions of the experimental path where some position estimated with a 70 cm error took place are (i) at beginning of the flight (at the starting point) and (ii) at

the end of the corridor, where the UAV executes the *U-turn*. This is due to the fact that these parts of the path were not covered effectively by UWB anchors, thus making the position estimate biased by the NLOS ToF estimation of the other UWB anchors which are far away. Another part of the path where a 50 cm estimation position error can be noted is close to the elevator's room, where some point have been marked within the wall. Even in this case, the poor position estimation performance can be attributed to the NLOS condition between nearby UWB anchors, that (due to signal reflections) introduces a range estimation error.

5.2.3 Final Considerations

The results discussed in Subsection 5.2.2 confirm the effectiveness of COTS-based UWB localization solutions which, without any in-deep knowledge of radio-based localization technology and algorithms, allow to easily deploy several anchors in an environment and integrate the UWB tag on different kind of targets to be tracked (including, obviusly, UAVs). The proposed solutions lead to very accurate position estimation, especially in the presence of LOS between the anchors and the tag, where the localization error can be lower than the advertised 30 cm. Moreover, it must be noted that in the reference testbed setup, the position of the tag is estimated only once per second. By increasing the power consumption or by adopting a TDoA approach, according to with the position can be estimated in the cloud, it would be possible to further increase the position estimate's update rate, up to 10 Hz, which might be useful for more critical BVLOS missions or for a filtering processing (e.g., advanced filtering) of the estimated positions.

The experimental results also confirm two main limitations of the proposed solution: (i) the need to deploy a certain amount of UWB anchors in the environment, in order to guarantee LOS conditions on the majority of the cases, and (ii) the relevant complexity and cost needed to deploy and keep several anchors synchronised in large environments. Those reasons, especially with regard to the complexity, pushed the research direction followed in this doctoral thesis toward the investigation of different radio-based localization technologies, mainly based on the cheap and widespread Wi-Fi, whose detail are discussed starting from Section 5.3.

5.3 Wi-Fi-based Localization

With regard to Wi-Fi-based localization techniques, there are several possible ways to exploit existing deployed Wi-Fi infrastructures for positioning purposes. In particular, two possible Wi-Fi localization techniques can be defined: (i) *passive* nd (ii) *active*.

Passive localization techniques include the most common Wi-Fi fingerprinting, as well as RSSI-based localization approaches. More in detail, they are defined as being passive as the target does not need to exchange data with the existing infrastructure, i.e., listening for incoming signals is sufficient, thus allowing to passively exploit the available APs deployed in the environment. At the opposite, *active* localization techniques involve both target and APs, as in a ToF-based approach.

5.3.1 Passive Localization

Wi-Fi Fingerprinting Localization

Wi-Fi fingerprinting localization is a clever technique that exploits the Wi-Fi signals broadcasted by APs present in an environment. More in detail, it relies on the unique behavior of Wi-Fi signals of interacting with the surrounding space, then causing variations in the signal strength, phase and quality. Therefore, the way Wi-Fi signals propagate through an indoor space can be considered similar to a "signature," with this RF signature changing through the environment and possibly being exploited for localization. This might happen by creating (through several in-field measurements) a heat map associated with a database containing RF data, such as APs' MAC addresses and their RSSIs in different positions, which are then used as unique *fingerprint* for each measurement location. Obviously, this approach is effective if the environment does not change.

Once both the heat map and the corresponding measurements database for the whole area of interest are available (denoted as *offline phase*), they can be used for the *online phase*, where the target device to be localized performs, at each position, a Wi-Fi scan of the environment, thus obtaining the RF data (i.e., the radio fingerprint) at each position. Through the use of the gathered data, it can then search within the

built database which is the location corresponding to the input *fingerprint*. Then, once (and *if*) a match is found, the system uses different mathematical techniques to estimate the approximate device's location, which, according to the density of the map built during the *offline phase* as well as the adopted algorithms (often relying on ML techniques), can lead to a position error of a few meters [165]. While this technique is heavily used on smartphones for indoor localization through shared heat maps [166], the accuracy of Wi-Fi fingerprinting depends on the signal variability and environmental changes. This implies that (i) if the layout changes, (ii) if APs are added or removed, as well as (iii) if there is interference incoming from other wireless devices, the system accuracy can be heavily affected.

Wi-Fi RSSI-based Localization

A simpler approach, not requiring to *a-priori* know the position of the APs deployed in the environment, is based on the use of the gathered RSSI measurements form each AP. In this case, the RSSI, which is a measure of the received RF power obtained by the radio hardware, is collected from the signals transmitted by multiple APs and is used to estimate the distance between the target to be localized and at least three APs (for bi-dimensional localization). More in the detail, the log-distance path loss model [167] relates the RSSI, defined as P_i, to the distance d_i as (in logarithmic scale)

$$P_i = P_0 + 10 \cdot \gamma \cdot \log_{10}\left(\frac{d}{d_0}\right) + X_g \tag{5.5}$$

where: P_i is the received signal power (dimension: [dBm]); P_0 is the measured received signal power (dimension: [dBm]) at a reference distance d_0 (dimension: [m]); γ is the path loss exponent (adimensional), a constant value typically between 2 and 4 depending on the environment; d_i is the distance between the AP and target node (dimension: [m]); and $X_g \sim \mathcal{N}(0, \sigma_g)$ is a normal distribution with zero mean, representing flat fading, used to model the signal envelope fluctuations and corresponding gains in received power. Therefore, by measuring P_0, γ and X_g in the target environment at d_0 (typically being $d_0 = 1$ m), it is possible to invert Eq. (5.5) to estimate d_i

from the measured RSSI P_i as follows:

$$d_i = 10^{-\left(\frac{P_i - P_0 - X_g}{10 \cdot \gamma}\right)}. \tag{5.6}$$

Once at least three ranges are estimated from different APs, it is possible to apply a multi-lateration algorithm and estimate the position of the target in the environment. However, while this approach has a very low implementation cost, since almost any Wi-Fi client can gather the RSSI from nearby APs, it is also known for its low accuracy, especially on medium- and long-range distances, where the log-distance path loss model, together with the fluctuations of the RSSI measurements, make very complicated to estimate the distance between the AP and the target, especially on mobile targets. This is even more true since only a few varying RSSI samples can be measured, thus hindering to properly filter the environmental noise, as shown in Figure 5.4.

Moreover, RSSI-based localization is also dependent on the antennas used by the device and the APs, since they can significantly affect the RSSI values based on their gains and directivities. Therefore, a proper value of P_0 must be measured and defined for each device. Also, radio signal propagates in different ways according to the environment, since obstacles like walls, humans, and other material, can significantly attenuate the signal propagation in the environment, making the distance estimation from the log-distance model inaccurate.

5.3.2 Active Localization

Wi-Fi ToF-based Localization

The most common active localization technique is based on ToF measurement, which has been officially introduced in the Wi-Fi IEEE 802.11mc enrichment standardized in June 2016, under the name of FTM [168]. Several COTS Wi-Fi chipset already support this enrichment, hence allowing to exploit it for meter-level localization purposes. More in detail, FTM utilizes Round-Trip Time (RTT) measurements to determine distances between the Wi-Fi client and APs. Then, as for the UWB technology discussed in Section 5.2, FTM exploits the TWR, thus not requiring any clock syn-

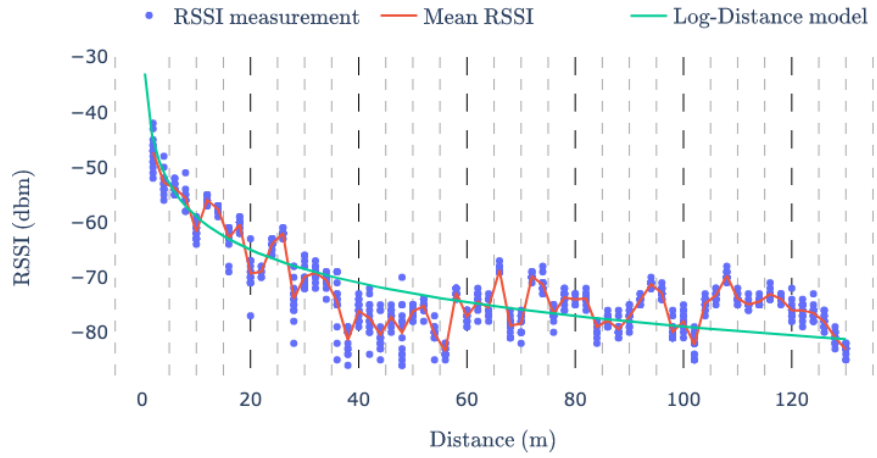


Figure 5.4: Experimental measurement of the Wi-Fi RSSI at different distances. Blue dots: are the measured values at different distances; red line: is the average RSSI value for each measurement point; green line: is the theoretical log-distance path loss model curve.

chronization between the two parties, since the RTT is computed by sending several messages between the *initiator* (i.e., the Wi-Fi client) and the *responder* (i.e., the AP). However, unlike the UWB technology, which, given the extremely short RF pulses (and ultra large bandwidth), only one TWR is sufficient to estimate the RTT between two parties, Wi-Fi FTM requires the exchange of several messages in order to determine a reliable average value $\overline{RTT}$ and, thus, the distance between the two parties.

As shown in Figure 5.5, after the *initiator* performs a Wi-Fi scan of the surrounding environment, the process begins with the initiator sending a *FTM Request* frame to the target *responder*, which immediately replies with an *Acknowledgment (ACK)* frame to the *initiator*. At this point, according to the defined parameters (namely, the number of *FTM* frames to exchange), a series of *n FTM* and *ACK* frames, defined as *Burst*, are exchanged between the two parties. More in detail, for the i-th RTT with $i \in \{1,\dots,4\}$ these frames contain the Time of Departures (ToDs) $t_1^{(i)}$ and $t_3^{(i)}$, as well as the Time or Arrivals (ToAs) $t_2^{(i)}$ and $t_4^{(i)}$ of the previous exchanged frames, in this way allowing the *responder* to compute the RTT_i and send it back to the *initiator* as

$$RTT_i = (t_4^{(i)} - t_1^{(i)}) - (t_3^{(i)} - t_2^{(i)}) \tag{5.7}$$

where: $t_1^{(i)}$ is the ToD at which the *FTM* frame has been sent by the *responder*; $t_2^{(i)}$ is the ToA at which the *FTM* frame has been received by the *initiator*; $t_3^{(i)}$ is the ToD of the *ACK* frame; $t_4^{(i)}$ is the ToA at which the ACK has been received by the *responder*. Since during a *Burst n* RTT measurements $\{RTT_i\}_{i=1}^n$ are exchanged, the *initiator* computes the average RTT as

$$\overline{RTT} = \frac{1}{n} \cdot \sum_{i=1}^{n} RTT_i \,. \tag{5.8}$$

Therefore, it is finally possible to compute the τ_{tof} as

$$\tau_{tof} = \frac{\overline{RTT}}{2} \,. \tag{5.9}$$

Consequently, the *initiator* can compute the distance d from the *responder* as

$$d = \tau_{tof} \cdot v + \varepsilon_{cal} \tag{5.10}$$

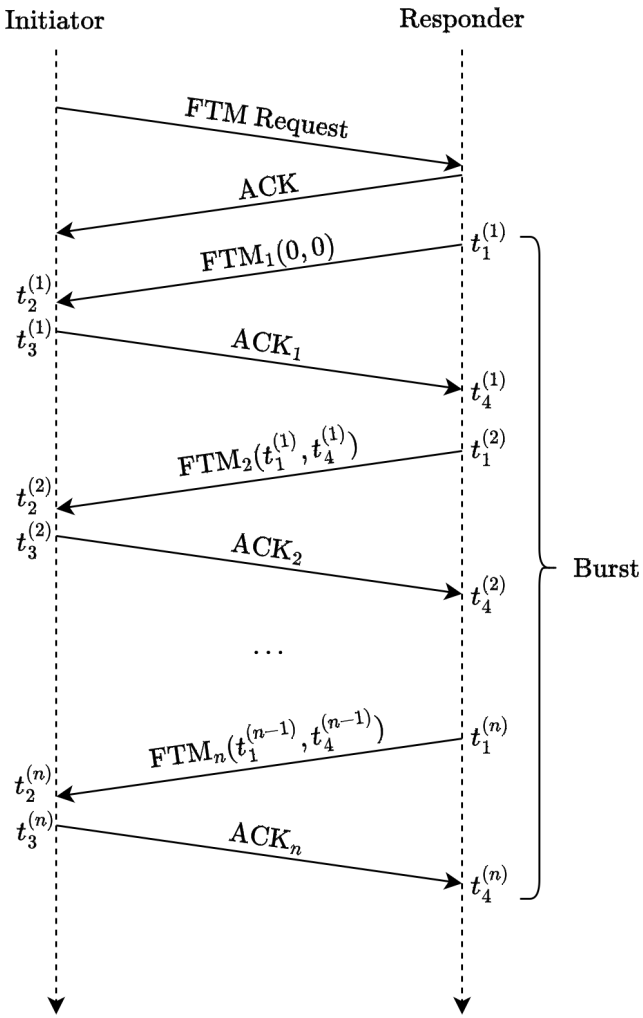


Figure 5.5: FTM frames exchange sequence to determine the RTT between the *initiator* and the *responder*.

where: v is the speed of light (dimension: [m/s]); ε_{cal} is an offset (dimension: [m]) used to calibrate the delays introduced by the specific device. Once at least three distances from known APs are gathered, performing the multi-lateration allows to estimate the position of device.

The advantages of Wi-Fi FTM are mainly related to (i) its low implementation cost, since the most recent APs and device already support it, and (ii) its higher precision than traditional RSSI-based localization solutions, since it is less affected by the presence of obstacles and signal attenuation and is not affected by the antennas' directivity. However, even this solution has some drawbacks, especially if compared with the UWB technology. First of all, given the narrower bandwidth (with respect of UWB)—in the 2.4 GHz band, is 20 MHz or 40 MHz, while in the 5 GHz band, it can reach 160 MHz—the expected rr is one order of magnitude higher (i.e., worse) than that with the UWB technology. More in detail, for a 2.4 GHz $B = 40$ MHz system, the smallest rr, which can be computed thanks to Eq. (5.1), is approximately equal to 3.75 m, which become 7.5 m for $B = 20$ MHz systems. In the 5 GHz $B = 80$ MHz Wi-Fi systems, it is possible to achieve a rr equal to 1.88 m, further reaching $rr = 0.94$ m with $B = 160$ MHz, as confirmed in [169]. Moreover, since the protocol works on different devices, different values of ε_{cal} must be taken into account for each device, limitating its interoperability only to well-known devices. Finally, unlike UWB, Wi-Fi FTM can be heavily affected by signal reflections in multi-path scenarios, but guaranteeing a wider operational range.

In the following, both Wi-Fi RSSI and FTM have been evaluated applied to a UAV for real-time localization purposes, in order to verify and estimate how better the FTM-based localization solution can perform with respect to a RSSI-based solution on a flying drone in a large operational area, all adopting the cheapest COTS FTM hardware available on the market, namely the ESP32S3 [170], operating at 2.4 GHz with a 40 MHz bandwidth.

5.3.3 Multi-Lateration

Once the range estimations between the target node and at least 3 APs or anchors are gathered (through passive or active approaches), it is possible to feed these data

together with the known positions of the APs into a geometrical multi-lateration algorithm, aiming at calculating the estimated position of the target device[1]. More in detail, since the multi-lateration algorithm [171] is based on the intersection between multiple spheres centered in the (known) APs' position coordinates and with radii equal to the estimated distances, in order to estimate the device position in a bi-dimensional environment, at least three APs are needed—while in a three-dimensional space, four APs are needed to identify a possible unique solution.

Given a list of known APs with coordinates $(x_i, y_i, z_i), i \in \{1, 2, \ldots, n\}$, together with the corresponding distances $d_i, i \in \{1, 2, \ldots, n\}$ between the device to be localized and the i-th AP, estimated according to one of the methods detailed and discussed in Subsection 5.3.2 and Subsection 5.3.1, it is possible to analytically define, at each measurement point, a system of equations, where each equation represents the squared distance from each AP:

$$\begin{cases} (P_x - x_1)^2 + (P_y - y_1)^2 + (P_z - z_1)^2 = d_1^2 \\ (P_x - x_2)^2 + (P_y - y_2)^2 + (P_z - z_2)^2 = d_2^2 \\ \quad \vdots \\ (P_x - x_n)^2 + (P_y - y_n)^2 + (P_z - z_n)^2 = d_n^2 \end{cases} \tag{5.11}$$

where P_x, P_y, and P_z represent the coordinates of the position of the device to be localized in the three-dimensional space.

The system of equations in Eq. (5.11) can be solved adopting an optimization algorithm trying to minimize the error. More in detail, in the system developed and described in this doctoral thesis, the Least Squares (LS) optimization algorithm [172] has been adopted to find the best fitting coordinates of the point P which minimize the sum of squared residuals of Eq. (5.11). Hence, it is possible to define the following objective function from Eq. (5.11):

$$C_{(P_x, P_y, P_z)} = \sum_{i=1}^{n} [(P_x - x_i)^2 + (P_y - y_i)^2 + (P_z - z_i)^2 - d_i^2]^2 \tag{5.12}$$

[1] Other non-geometric algorithms could be applied. However, this research direction is not investigated to keep the computational complexity low.

where the goal of a LS approach is to find the best fitting coordinates (P_x, P_y, P_z) minimizing the residuals of the function.

More in detail, the residual $r_i, i \in \{1, 2, \ldots, n\}$ correspond to the differences between the left-hand side and right-hand side of each equation in Eq. (5.11):

$$r_i = (P_x - x_i)^2 + (P_y - y_i)^2 + (P_z - z_i)^2 - d_i^2 \,. \tag{5.13}$$

Therefore, Eq.(5.12) can be expressed, in terms of residuals, as

$$C_{(P_x,P_y,P_z)} = \sum_{i=1}^{n} r_i^2 \,. \tag{5.14}$$

In order to minimize $C_{(P_x,P_y,P_z)}$, it is necessary to find its gradient with respect to the variables (P_x, P_y, P_z). Taking into account that the gradient is the following vector of partial derivatives:

$$\nabla C_{(P_x,P_y,P_z)} = \left[\frac{\partial C_{(P_x,P_y,P_z)}}{\partial P_x}, \frac{\partial C_{(P_x,P_y,P_z)}}{\partial P_y}, \frac{\partial C_{(P_x,P_y,P_z)}}{\partial P_z} \right] \tag{5.15}$$

where

$$\begin{aligned} \frac{\partial C_{(P_x,P_y,P_z)}}{\partial P_x} &= 2 \sum_{i=1}^{n} r_i \frac{\partial r_i}{\partial P_x} \\ \frac{\partial C_{(P_x,P_y,P_z)}}{\partial P_y} &= 2 \sum_{i=1}^{n} r_i \frac{\partial r_i}{\partial P_y} \\ \frac{\partial C_{(P_x,P_y,P_z)}}{\partial P_z} &= 2 \sum_{i=1}^{n} r_i \frac{\partial r_i}{\partial P_z} \,. \end{aligned} \tag{5.16}$$

Since

$$\begin{aligned} \frac{\partial r_i}{\partial P_x} &= 2(P_x - x_i) \\ \frac{\partial r_i}{\partial P_y} &= 2(P_y - y_i) \\ \frac{\partial r_i}{\partial P_z} &= 2(P_z - z_i) \end{aligned} \tag{5.17}$$

by combining Eq. (5.16) and Eq. (5.17), it is possible to obtain the following expression for the gradient of the objective function:

$$\nabla C_{(P_x,P_y,P_z)} = \left[2 \sum_{i=1}^{n} r_i \cdot 2(P_x - x_i), 2 \sum_{i=1}^{n} r_i \cdot 2(P_y - y_i), 2 \sum_{i=1}^{n} r_i \cdot 2(P_z - z_i) \right] . \tag{5.18}$$

Finally, to find the optimal solution (under the assumption that the objective function is convex), the gradient must be set to zero:

$$\nabla C_{(P_x,P_y,P_z)} = \mathbf{0}\,. \tag{5.19}$$

Therefore, it is possible to solve the system of equations adopting the Least Squares with Multiple Right Hand Sides (LSMR) [173] algorithm, that iteratively updates the values of (P_x, P_y, P_z) using the gradient of the cost function and the learning rate α to minimize the objective function:

$$\begin{aligned} P_x &= P_x - \alpha \cdot \nabla C_{(P_x,P_y,P_z)} \\ P_y &= P_y - \alpha \cdot \nabla C_{(P_x,P_y,P_z)} \\ P_z &= P_z - \alpha \cdot \nabla C_{(P_x,P_y,P_z)}\,. \end{aligned} \tag{5.20}$$

This iterative optimization process continues until the cost function converges to a minimum, providing the optimal (P_x, P_y, P_z) values for localization.

To further optimize the process to find the best fitting coordinates, it is possible to apply boundaries or constraints to the LS optimization problem, to ensure that the estimated parameters lie within a specific range or satisfy certain conditions. For instance, in Wi-Fi-based localization of a UAVs, it might be useful to apply a constraint to P_Z, as it cannot be negative. Moreover, it is also possible to rely on the UAV on-board barometer to estimate the flight altitude (i.e., P_z), which is already computed and provided by the UAV's FC. Therefore, P_Z can be assumed as known, thus simplifying Eq. (5.16) as

$$\begin{aligned} \frac{\partial C_{(P_x,P_y)}}{\partial P_x} &= 2\sum_{i=1}^{n} r_i \frac{\partial r_i}{\partial P_x} \\ \frac{\partial C_{(P_x,P_y)}}{\partial P_y} &= 2\sum_{i=1}^{n} r_i \frac{\partial r_i}{\partial P_y} \end{aligned} \tag{5.21}$$

where

$$\begin{aligned} \frac{\partial r_i}{\partial P_x} &= 2(P_x - x_i) \\ \frac{\partial r_i}{\partial P_y} &= 2(P_y - y_i) \end{aligned} \tag{5.22}$$

thus leading to the following simplified expression of the gradient of the objective function:

$$\nabla C_{(P_x,P_y)} = \left[2\sum_{i=1}^{n} r_i \cdot 2(P_x - x_i), 2\sum_{i=1}^{n} r_i \cdot 2(P_y - y_i) \right] . \tag{5.23}$$

5.3.4 Kalman Filter

At this point, in order to enhance the Wi-Fi-based position estimation accuracy, it might be useful (and might be considered a good practice) to apply some filtering techniques to the ranges estimated according to the RSSI or FTM methods described before, thus allowing to remove outlier measurements due to the environmental noise affecting the Wi-Fi signals reaching the receiver of the device to be localized. Among various possible filtering approaches, the most common and adopted filtering solution is the Kalman Filter (KF) [174]. In detail, the KF is a recursive mathematical filter used for estimating and predicting the state of linear and non-linear systems affected by noisy measurement inputs. This is why KFs are widely used in heterogeneous fields, including control systems, signal processing, localization, and even finance.

In detail, KFs have been initially proposed for linear systems, since they require low computational capabilities with respect to more complex approaches. However, in the last years, an Extended-KF (EKF) [175] has been proposed to be applied to non-linear systems, also thanks to the increased computing capabilities of mobile devices. Both KF and EKF iteratively combine information from previous estimates and current measurements to provide an accurate and optimal estimate of the state of a system, even in the presence of noisy measurements and uncertainty, thus allowing to significantly reduce the measurement errors.

Hence, when dealing with a KF, the system dynamics are crucial for the filtering process, thus determining the overall performance of the filter. Therefore, it is necessary to define the transition model and observation model that describe how the state of a dynamic system evolves over time and how it relates to measurements, respectively.

The state transition model of a KF can be generalized as [176]

$$\mathbf{x}_t = \mathbf{F}_t \cdot \mathbf{x}_{t-1} + \mathbf{B}_t \cdot \mathbf{u}_t + \mathbf{w}_t \tag{5.24}$$

where: $\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q}_t)$ represents the process noise, described by a normal distribution with a zero mean and covariance matrix $\mathbf{Q}_t$; $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$ are the current and previous estimated system state vectors at time t and $t-1$, respectively; $\mathbf{F}_t$ is the state transition matrix at time t; $\mathbf{u}_t$ is the control input while $\mathbf{B}_t$ the control input matrix, both at time t.

Instead, the observation model can be expressed as [176]

$$\mathbf{z}_t = \mathbf{H}_t \cdot \mathbf{x}_t + \mathbf{v}_t \tag{5.25}$$

where: $\mathbf{v}_t \sim \mathcal{N}(0, \mathbf{R}_t)$ is the observation noise, described by a normal distribution with a zero mean and covariance matrix $\mathbf{R}_t$; $\mathbf{z}_t$ is the observation (or measurement) vector; $\mathbf{H}_t$ is the observation matrix that describes the relationship between the state information vector $\mathbf{x}_t$ and the observation vector $\mathbf{z}_t$.

In order to properly estimate the new state of the system, the KF relies on two phases [177]: (i) *State Predict* and (ii) *State Update*. In particular, in the *State Predict* phase the new state is predicted using the defined state transition model, while the *State Update* phase refines the state estimate by incorporating the measurements using the observation model. This iterative process allows the KF to estimate the state of a linear dynamic system in the presence of noise and uncertainties.

The *State Predict* phase, which predicts the *a-priori* state information vector $\hat{\mathbf{x}}_{t|t-1}$ using the information from the previous *a-posteriori* state information vector $\mathbf{x}_{t-1|t-1}$, is defined as [176]

$$\hat{\mathbf{x}}_{t|t-1} = \mathbf{F}_t \cdot \mathbf{x}_{t-1|t-1} + \mathbf{B}_t \cdot \mathbf{u}_t \tag{5.26}$$

where: $\mathbf{u}_t$ is the system control input vector; $\mathbf{F}_t$ is the state transition matrix; $\mathbf{B}_t$ is the control input matrix.

The estimated *a-propri* covariance matrix $\hat{\mathbf{P}}_{t|t-1}$ can be obtained as [176]

$$\hat{\mathbf{P}}_{t|t-1} = \mathbf{F}_t \cdot \mathbf{P}_{t-1|t-1} \cdot \mathbf{F}^{\mathsf{T}} + \mathbf{Q}_t \tag{5.27}$$

where: $\mathbf{P}_{t-1|t-1}$ is the previous state *a-posteriori* covariance matrix; $\mathbf{Q}_t$ denotes the covariance matrix of process noise.

The *State Update* phase corrects the *a-propri* state information through the use of the observation vector. In detail, in this phase the innovation vector $\widetilde{\mathbf{y}}_t$, representing the difference between the observed measurement and the predicted measurement based on the current state estimate, is defined as [176]

$$\widetilde{\mathbf{y}}_t = \mathbf{z}_t - \mathbf{H}_t \cdot \hat{\mathbf{x}}_{t|t-1} \,. \tag{5.28}$$

The covariance matrix $\mathbf{S}_t$ can be defined as [176]

$$\mathbf{S}_t = \mathbf{H}_t \cdot \hat{\mathbf{P}}_{t|t-1} \cdot \mathbf{H}_t^{\mathsf{T}} + \mathbf{R}_t \,. \tag{5.29}$$

At this point, the updated *a-posteriori* state vector $\hat{\mathbf{x}}_{t|t}$ can be calculated as [176]

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t \cdot \widetilde{\mathbf{y}}_t \tag{5.30}$$

where $\mathbf{K}_t$ is the *optimal* Kalman gain, which can be obtained as [176]

$$\mathbf{K}_t = \hat{\mathbf{P}}_{t|t-1} \cdot \mathbf{H}_t^{\mathsf{T}} \cdot \mathbf{S}_t^{-1} \,. \tag{5.31}$$

Finally, the updated *a-posteriori* state covariance matrix $\mathbf{P}_{t|t}$ can be derived as [176]

$$\mathbf{P}_{t|t} = (\mathbf{I} - \mathbf{K}_t \cdot \mathbf{H}_t) \cdot \hat{\mathbf{P}}_{t|t-1} \,. \tag{5.32}$$

Once the KF has been set, the aforementioned parameters (namely: $\mathbf{P}_0$, $\mathbf{R}_t$, $\mathbf{Q}_t$, $\mathbf{H}_t$, $\mathbf{u}_t$, $\mathbf{F}_t$) must be defined according to their filter application. In the developed Wi-Fi-based localization system architecture (detailed in Subsection 5.3.5) the KFs are applied in two different parts of the system, namely: (i) at the reception of the RSSI or FTM measurement form each AP, and (ii) to further filter the estimated RSSI- or FTM-based positions computed by the multi-lateration algorithm described in Subsection 5.3.3. Moreover, different KFs with their parameters must be defined accordingly. The reason behind the double KF implementation in the developed system is detailed in the following.

The first KF, denoted as pre-filter RSSI-KF, is applied to the RSSI measurement arriving from the Wi-Fi scanner of the device to be localized, with the aim the remove possible outliers measurement. More in detail, in the case of the RSSI-based localization system, a RSSI-KF$_{AP_i}$ is assigned to each known AP_i, thus allowing to filter the

measured RSSI according to the previous measurement of the same AP_i. This means that, in the experimental setup of the developed system with 6 APs, 6 RSSI-KF$_{AP_i}$ (one for each AP) are initialized by the developed localization algorithm. Therefore, in the RSSI-KF$_{AP_i}$, the estimated $\mathbf{x}_t^{(\mathrm{AP}_i)}$ given by Eq. (5.24) are defined as

$$\mathbf{x}_t^{(\mathrm{AP}_i)} = \hat{RSSI}_t^{(\mathrm{AP}_i)} \tag{5.33}$$

while the observed $\mathbf{z}_t^{(\mathrm{AP}_i)}$ of Eq. (5.25) are defined as

$$\mathbf{z}_t^{(\mathrm{AP}_i)} = RSSI_t^{(\mathrm{AP}_i)}. \tag{5.34}$$

Since the estimated vector has only one element, all the aforementioned matrices have one element. Moreover, all $\{\text{RSSI-KF}_{AP_i}\}_{i=1}^{6}$ share the same initial covariance matrix $\mathbf{P}_0 = \mathbf{1}$, the same observation noise $\mathbf{R}_t = 3 \cdot \mathbf{I}$, and the same process noise $\mathbf{Q}_t = 0.2 \cdot \mathbf{I}$. Those values have been chosen according to *in-field* experimental measurements of the RSSI in both static and mobile conditions of the device to be localized, as further detailed in Subsection 5.3.6. When the RSSI-KF$_{AP_i}$ are initialized, the initial estimated states $\mathbf{x}_0^{(\mathrm{AP}_i)}$ are set equal to the first observations $\mathbf{z}_0^{(\mathrm{AP}_i)}$. Since in the developed system there is no control input signal $\mathbf{u}_t$, this term can be neglected, thus also removing $\mathbf{B}_t$. Instead, $\mathbf{H}_t$ and $\mathbf{F}_t$ are equal to the identity matrix $\mathbf{I} = \mathbf{1}$. The computed average gain RSSI-KF$_{AP_i}$ with the aforementioned parameters is equal to 0.227.

The same approach (initializing a filter for each $AP_i, i \in \{1, \ldots, 6\}$ is applied to the FTM. In detail, a FTM-KF$_{AP_i}$ is assigned to each known AP_i, in this way allowing to filter the measured FTM according to the previous measurement of the same AP_i. In the FTM-KF$_{AP_i}$, the estimated $\mathbf{x}_t^{(\mathrm{AP}_i)}$ is defined as

$$\mathbf{x}_t^{(\mathrm{AP}_i)} = \hat{FTM}_t^{(\mathrm{AP}_i)} \tag{5.35}$$

while the observed $\mathbf{z}_t^{(\mathrm{AP}_i)}$ can be expressed as

$$\mathbf{z}_t^{(\mathrm{AP}_i)} = FTM_t^{(\mathrm{AP}_i)}. \tag{5.36}$$

All $\{\text{FTM-KF}_{AP_i}\}_{i=1}^{6}$ share the same initial covariance matrix $\mathbf{P}_0 = \mathbf{1}$, the same observation noise $\mathbf{R}_t = 2 \cdot \mathbf{I}$, and the same process noise $\mathbf{Q}_t = 0.2 \cdot \mathbf{I}$. Similarly to

the previous case, those values have been chosen according to *in-field* experimental measurements of the FTM in both static and mobile conditions of the device to be localized. Then, as for RSSI-KF$_{AP_i}$ filters, the initial estimated states $\mathbf{x}_0^{(\mathrm{AP}_i)}$ are set equal to the first observation $\mathbf{z}_0^{(\mathrm{AP}_i)}$ to initialize the filters. Even in this case, the terms $\mathbf{u}_t$ and $\mathbf{B}_t$ are also neglected, while $\mathbf{H}_t$ and $\mathbf{F}_t$ are equal to the identity matrix $\mathbf{I} = \mathbf{1}$. Finally, the computed average gain FTM-KF$_{AP_i}$ with the aforementioned parameters is equal to 0.271.

Instead, the second KFs, denoted as multi-lateration filters POS-KF, are applied to the estimated RSSI $(P_x^{RSSI}, P_y^{RSSI}, P_z^{BARO})$ and FTM $(P_x^{FTM}, P_y^{FTM}, P_z^{BARO})$ positions computed by the multi-lateration algorithm. Therefore, the estimated state vector $\mathbf{x}_t^{RSSI}$ of the POS-KF^{RSSI} applied to the RSSI-based position, is defined as

$$\mathbf{x}_t^{RSSI} = (\hat{P}_x^{RSSI}, \hat{P}_y^{RSSI}, \hat{P}_z^{BARO})_t^\intercal \tag{5.37}$$

while the observed $\mathbf{z}_t^{RSSI}$ is defined as

$$\mathbf{z}_t^{RSSI} = (P_x^{RSSI}, P_y^{RSSI}, P_z^{BARO})_t^\intercal . \tag{5.38}$$

The estimated state vector $\mathbf{x}_t^{FTM}$ of the POS-KF^{FTM} applied to the FTM-based position is defined as

$$\mathbf{x}_t^{FTM} = (\hat{P}_x^{FTM}, \hat{P}_y^{FTM}, \hat{P}_z^{BARO})_t^\intercal \tag{5.39}$$

while the observed $\mathbf{z}_t^{FTM}$ can be expressed as

$$\mathbf{z}_t^{FTM} = (P_x^{FTM}, P_y^{FTM}, P_z^{BARO})_t^\intercal . \tag{5.40}$$

In both cases, the estimated vector has dimension equal to 3×1 and, therefore, all matrices have dimensions 3×3. The initial covariance matrix $\mathbf{P}_0$ is defined as

$$\mathbf{P}_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.41}$$

while the observation noise $\mathbf{R}_t$ is defined as

$$\mathbf{R}_t = \begin{bmatrix} 0.8 & 0 & 0 \\ 0 & 0.8 & 0 \\ 0 & 0 & 0.8 \end{bmatrix} \tag{5.42}$$

and the process noise $\mathbf{Q}_t$ as

$$\mathbf{Q}_t = \begin{bmatrix} 0.2 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.2 \end{bmatrix}. \tag{5.43}$$

Hence, when filters are initialized, the initial estimated state $\mathbf{x}_0^{RSSI}$ and $\mathbf{x}_0^{FTM}$ are set equal to the first observations $\mathbf{z}_0^{RSSI}$ and $\mathbf{z}_0^{FTM}$, respectively. Again, since there is no control input signal, the terms $\mathbf{u}_t$ and $\mathbf{B}_t$ are neglected, while $\mathbf{H}_t$ and $\mathbf{F}_t$ are set equal to the identity matrix $\mathbf{I}$:

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.44}$$

Finally, the computed average gain POS-KF with the aforementioned parameters is equal to 0.131.

All the aforementioned KFs' parameters have been chosen according to experimental evaluations carried out in both static and mobile conditions, aiming at finding a good *trade-off* in the KF tuning process, suitable for both scenarios, as it will be further discussed in the following, also allowing to understand where the use of both the filters can be beneficial, as well as where only one of the identified KFs (the first or the second) can fit better.

5.3.5 System Architecture

The developed low cost real-time Wi-Fi-based architecture is composed of several elements. On the hardware side, the Wi-Fi RSSI and FTM measurements are gathered on the common platform (both for the APs as well as the target device to localize) based on the ESP32S3 SoC, embedded in the Lilygo T3S3 development board [178]. One Lilygo T3S3 has been integrated, within its own 3D-printed case, on the Tarot 650-based UAV platform, as shown in Figure 5.6(a), in turn connected to the on-board RPi4 SBC running Ubuntu 20.04, as well as ROS framework [8]. An overview of the designed system architecture is depicted in Figure 5.7. The final

(a) (b)

Figure 5.6: Equipment used for the Wi-Fi localization test: (a) the quad-copter with the on-board ESP32S3 module attached to the RPi4, and (b) one of the six ESP32S3 APs, within its own 3D-printed case, deployed in the test environment.

goal aims at properly gathering ground truth data arriving from the GNSS receiver attached to the UAV's FC, as well as the Wi-Fi positioning data of the designed system, thus allowing to properly evaluate the performance of the developed solution. Then, six Lilygos T3S3, together with their 3D-printed case, as shown in Figure 5.6(b), have been adopted as Wi-Fi APs and deployed in the environment.

More in detail, the ESP32S3 firmware has been customized according to the role of the Lilygo T3S3. In detail, on the boards used as Wi-Fi APs, the developed firmware has been properly designed to exploit the ESP32S3 SoC to work as a Wi-Fi FTM *responder* on the 2.4 GHz band (with $B = 40$ MHz). Instead, the development board integrated on the UAV and connected to the on-board RPi4 through the USB cable, has been designed to implement a Wi-Fi scanner node continuously scanning the available Wi-Fi APs. The goal of the UAV is to retrieve MAC address, SSID, and RSSI (with 1 dBm granularity) of each Wi-Fi AP, in turn performing as a FTM *initiator* with the FTM-enabled APs.

With regard to the FTM process (detailed in Section 5.3.2), in order to reduce the time needed to complete the computation of $\overline{RTT}$, the number of *FTM* frames to be exchanged for each *Burst* has been set equal to 8, with a total of only one *Burst*

exchanged, trying to minimize the impact of the movement of the UAV. Therefore, this leads to a time duration of the *Burst* equal to 32 ms that, considering the Wi-Fi scan time and the six APs used for the localization experiment, leads to a measured average iteration period $\theta_{cycle} = 522$ ms.

Then, the Wi-Fi scan results are parsed into a JSON string and sent through the USB-enabled serial port to the RPi4, with the output JSON string depending on the result of the Wi-Fi scan process:. If the scanned AP is an FTM *responder*, then the output JSON includes the FTM-related fields, as follows:

```
{"SSID": "FTM_1", "MAC": "f6:12:fa:5a:05:10",
"rtt_est": 346, "rtt_raw": 478 , "dist_est": 5200,
"num_frames": 15, "mean_rssi": -45.00}
```

where: `SSID` corresponds to the name of the Wi-Fi network; the `MAC` address is the unique physical address of the Wi-Fi network; `rtt_est` is the $\overline{RTT}$ between the *responder* and *initiator* (dimension: [ns]) taking into account the clock drift compensation; `rtt_raw` is the $\overline{RTT}_{RAW}$ between the *responder* and *initiator* (dimension: [ns]); `dist_est` is the average estimated distance $\overline{d}$ between the *responder* and *initiator* (dimension: [cm]), derived from the `rtt_est` and without the initial off-set correction; `num_frames` is the total number of exchanged frames (i.e., including both the *FTM* and *ACK* frames) during the FTM process; and, finally, `mean_rssi` is the average RSSI (dimension: [dBm]) measured over all the exchanged FTM frames.

If, instead, the scanned AP is not an FTM *responder*, then only `SSID`, `MAC` address and `RSSI` are included, as follows:

```
{"SSID": "Tenda", "MAC": "f2:5a:01:5a:12:f4",
"mean_rssi": -52.00}
```

At the UAV side, the RPi4 receives these JSON strings and, through a self-made Python script, separates the known APs (thus the ones used for the localization task, whose GNSS coordinates are known) from those which correspond to unknown APs. Once the known APs' list is built, the RSSI-KF$_{AP_i}, i \in \{1,2,\ldots,6\}$ is fed with the new observed $RSSI_{AP_i}$. The same is performed with the FTM-KF$_{AP_i}$, which is instead fed

with the $FTM_{AP_i}, i \in \{1,2,\ldots,6\}$ measurements. If this is the first interaction of the system or if a new AP is added to the list, the relative KFs are initialized, as described in Subsection 5.3.4. Once the input data are filtered, in case of the RSSI-based localization, the log-distance path loss model is applied to the filtered $RSSI_{AP_i}^{FILT}$, while for the FTM-based solution, the calibration offset ε_{cal} is removed from the filtered $FTM_{AP_i}^{FILT}$ measurement. In detail, for the log-distance path loss model, $\gamma = 3.65$ and $P_0 = -27$ dBm have been measured in the experimental environment at the reference distance $d_0 = 1$ m. Instead, for the FTM calibration, according to the experimental measurement, the offset calibration has been set to $\varepsilon_{cal} = -52.82$ m.

Then, the output of log-distance estimation block and FTM offset removal, namely $d_{AP_i}^{RSSI}$ and $d_{AP_i}^{FTM}$, are sent to the multi-lateration algorithm, together with the altitude (dimension: [m]), which corresponds to the P_z^{BARO} of the device to be localized and is retrieved from the UAV's FC (which estimates it through the internal barometer). Before applying the multi-lateration, which has been described in Section 5.3, the GNSS coordinates of the known APs are converted from the World Geodetic System 1984 (WGS84) [179] standard to the Earth-Centered, Earth-Fixed (ECEF) [180] coordinates, allowing to apply the multi-lateration algorithm together with the LS. Once both the RSSI-based $(P_x^{RSSI}, P_y^{RSSI}, P_z^{BARO})$ and FTM-based $(P_x^{FTM}, P_y^{FTM}, P_z^{BARO})$ coordinates have been calculated, they are fed into the POS-KF filters, namely one for the FTM-based localization and one for the RSSI-based localization.

Finally, the output of these latter KFs, namely the filtered RSSI-based estimated position $(\hat{P}_x^{RSSI}, \hat{P}_y^{RSSI}, \hat{P}_z^{BARO})$ and the filtered FTM-based estimated position $(\hat{P}_x^{FTM}, \hat{P}_y^{FTM}, \hat{P}_z^{BARO})$ coordinates, are converted from ECEF to WGS84 coordinates and become the final estimated positions (according to the RSSI or FTM) of the UAV. A high level overview of the system architecture is depicted in Figure 5.7.

In order to facilitate the performance evaluation, all the raw Wi-Fi scan data, as well as the computed RSSI and FTM coordinates, are published to a custom-made ROS topic, thus allowing to record the estimated position, together with the ground truth position incoming from the UAV's GNSS receiver, and ensuring the synchronization between all the data coming from the different sensors. This will aid the results evaluation and dissemination of Subsection 5.3.6.

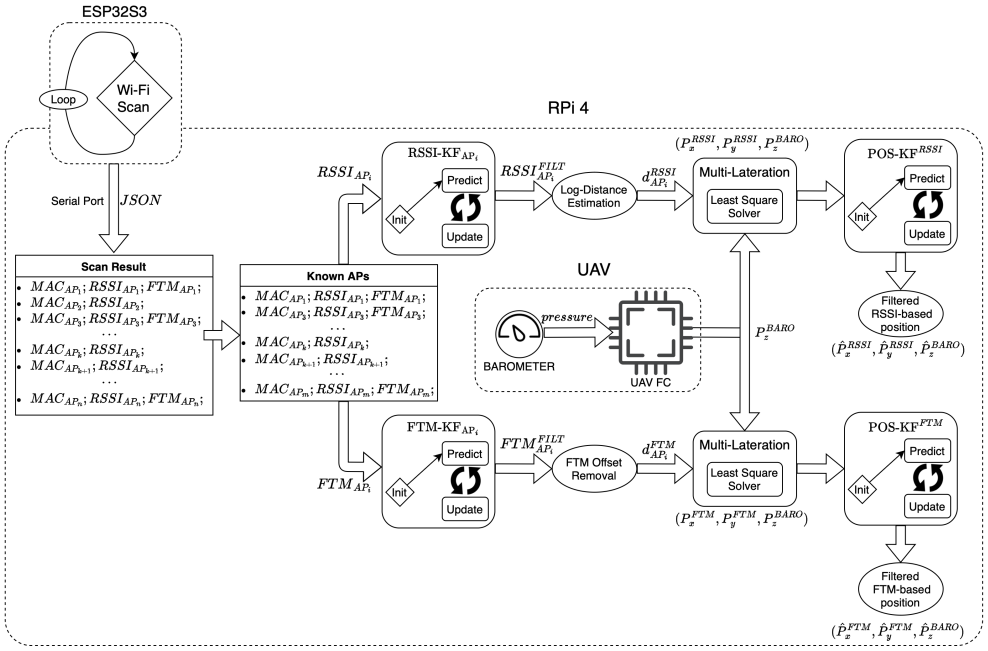


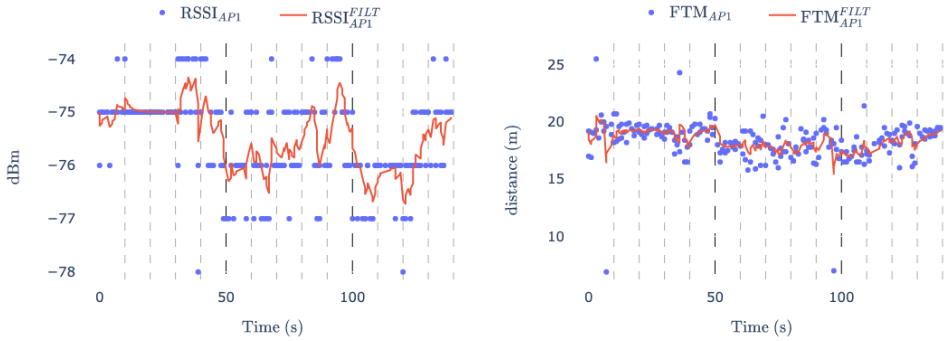
Figure 5.7: System architecture of the developed Wi-Fi-based localization solution.

5.3.6 Results

In order to evaluate the performance of the developed Wi-Fi-based localization system, a testbed has been defined in a yard (measuring 26×21 m) in a sub-urban area. More in detail, the battery-powered APs have been deployed on several tripods at the same altitude (equal to 150 cm) at the boundaries of the yard. Then, the GNSS coordinates of the APs have been gathered using the GNSS-receiver of the drone positioned on the tripod for approximately 5 min. This has been done for each AP's position, aiming at collecting their average GNSS coordinates (in terms of latitude and longitude)—averaging allows to remove the small measurable drift afflicting GNSS systems. In detail, the point cloud composed by the GNSS measurements collected in 5 min at each AP has a radius of approximately 1 m: the reference coordinates of the AP to be used for the multi-lateration algorithm correspond the average values of the latitude and longitude.

All APs have been deployed in LOS conditions with each other and with respect to the UAV. Given the sub-urban environment, a few Wi-Fi 2.4 GHz networks are available in the same area,: in order to mitigate the possible interference effects, the deployed Wi-Fi APs have been set to operate on unused Wi-Fi channels.

The experimental evaluation has been carried out according to two procedures: (i) a static localization test, where the UAV has been hovering for about 2 min in the same position at approximately 2 m AGL, with just a few fluctuations due to the correction applied by the FC to compensate wind and altitude drift; (ii) the UAV has been let flying in `AUTO` mode at 2 m AGL with an average horizontal speed of 1.3 m/s for 3 times through a pre-defined flight path with a square shape, with a size of approximately 16×11 m, within the experimental area bordered by the APs. These tests have been carried out with the aim to validate both the static real-time localization performance, in static and mobile conditions, in order to understand how accurate the proposed system can be in terms of real-time tracking of a mobile object.



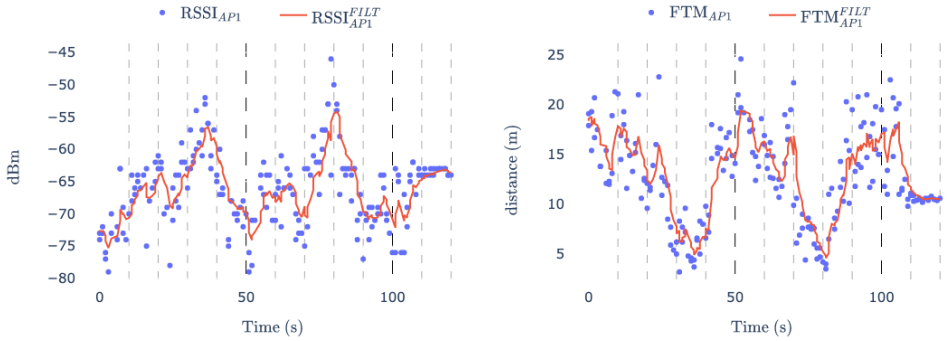
(a) (b)

Figure 5.8: Evaluation of the implemented KFs: (a) unfiltered and filtered AP_1 RSSI measurements with UAV in static hovering condition, and (b) unfiltered and filtered AP_1 FTM distance measurements with UAV in static hovering condition.

KF Evaluation

Before performing the in-flight tests, a preliminary evaluation of the implemented KFs adopted to mitigate the noise of the RSSI and FTM measurements has been carried out, aiming at evaluating the performance of the filters and properly setting the parameters discussed in Subsection 5.3.4.

In the static hovering experimental evaluation, carried out with respect to the AP_1, whose results are shown in Figure 5.8, the KF acts with an outlier removal function, filtering the measurement noise due to both environmental background noise as well as measurement error from the Wi-Fi scanner. In Figure 5.8(a), the unfiltered and filtered RSSI measurements with the UAV in static hovering condition from AP_1 are depicted, showing how the filter manages to remove some RSSI measurements at the boundaries of the observed RSSI's range, thus smoothing the RSSI values used for the log-distance estimation. In Figure 5.8(b), the unfiltered and filtered FTM-based distance measurements with the UAV in static hovering condition from AP_1 are shown, again highlighting how the KF manages to efficiently remove some FTM outliers, thus smoothing the FTM distance estimation.



(a) (b)

Figure 5.9: Evaluation of the implemented KFs: (a) unfiltered and filtered AP_1 RSSI measurements with UAV in mobile condition, and (b) unfiltered and filtered AP_1 FTM measurements with UAV in mobile condition.

Instead, the filtering capabilities (this time considering the mobile conditions of the UAV along the pre-defined flight path) are evaluated in Figure 5.9. As shown in Figure 5.9(a), the filtered RSSI measurements of the mobile UAV clarify how the KF manages to properly remove most of the outliers and smoothing the RSSI values. The same holds for the KF applied to the FTM measurements depicted in Figure 5.9(b), which confirms the good fit of the designed filter, suitable to remove some background noise without losing too much information, as it would happen with other filters (e.g., Exponential Mobile Average, EMA).

The achieved results also confirm that it is not possible to remove all the noise and properly smooth the input data, especially in a real-time filtering context, where the developed KFs managed to get a good balance in terms of filtering capabilities in both static and mobile conditions of the UAV, all with the same parameters defined in Subsection 5.3.4.

Positioning Error Evaluation

In the following, the localization accuracy of the proposed system has been evaluated in the two aforementioned scenarios (i.e., static hovering UAV and mobile UAV). However, in order to evaluate the impact of the implemented KFs and understand how much and when it is beneficial to use both, the localization results (in terms of position error with respect to the GNSS ground truth) have been evaluated in four different system setups for both the RSSI- and FTM-based localization solutions:

1. without any KF applied to the input RSSI and FTM data, denoted as *RAW RSSI position* and *RAW FTM position*, respectively;
2. with only the POS-KF applied, to the RAW RSSI and RAW FTM data, denoted as *RAW RSSI position with 2nd KF* and *RAW FTM position with 2nd KF*, respectively;
3. with only the RSSI-KF$_{AP_i}$ and FTM-KF$_{AP_i}$ filters applied to the raw input data before the multi-lateration, denoted as *RSSI position with 1st KF* and *FTM position with 1st KF*, respectively;
4. with both the double KFs applied at the raw input RSSI and FTM data as well as after the multi-lateration, denoted by *RSSI position with both KFs* and *FTM position with both KFs*, respectively.

Static UAV In the static hovering scenario, the *RAW RSSI position* solution achieved acceptable results, as shown in Figure 5.10(a) and further confirmed by the computed CDF, shown in Figure 5.10(c), where it can be noticed the beneficial effects of the *RAW RSSI position with 2nd KF* filtering solution, which reduces the overall positioning error, lowering the *RAW RSSI* mean position error from 2.58 m of the unfiltered method to 2.23 m, as listed in Table 5.1. The error reduction is higher for the 95th percentile position error, which reduces from the initial 4.61 m to 3.26 m of the *RAW RSSI with 2nd KF* solution.

With regard to the double filtering solution, Figure 5.10(b) shows a smaller radius of the point-cloud of the *RSSI position with both KFs* method near the true UAV

Table 5.1: Experimental evaluation of RSSI- and FTM-based localization with the UAV in static hovering condition with different filtering approaches.

Filtering Method	Mean Position Error (m)	95th Percentile Error (m)
RAW RSSI	2.58	4.61
RAW RSSI with 2nd KF	2.23	3.26
RSSI with 1st KF	2.54	4.20
RSSI with both KFs	2.21	3.13
RAW FTM	1.99	5.82
RAW FTM with 2nd KF	1.51	3.57
FTM with 1st KF	1.70	4.81
FTM with both KFs	1.41	3.22

GNSS coordinates with respect to the *RSSI position with 1st KF*, as also confirmed by the CDF shown in Figure 5.10(d), as well as by the values listed in Table 5.1. This holds for the mean positioning error for the double KFs solution, being equal to 2.21 m instead of the 2.54 m of the one with the pre-filtering approach. The results are further confirmed by the 95th percentile positioning error, with the *RSSI with both KFs* solution obtaining the best results among the four adopted RSSI filtering techniques, with a 95th percentile positioning error equal to 3.13 m.

Regarding the FTM-based solution, in the static hovering scenario the *RAW FTM position* solution achieved a more consistent and better result with respect to RSSI-based double KF solution, as listed in Table 5.1, with a mean positioning error of just 1.99 m. However, it is interesting to look at the higher 95th percentile error of this solution, equal to 5.82 m and, so far, the worst observed value for this metric. Comparing the *RAW FTM position* with the *RAW FTM position with 2nd KF* solution, the latter approach allows to lower the mean positioning error to 1.51 m, as well as the 95th percentile, which reduces to 3.57 m, as confirmed by the computed CDF shown in Figure 5.11(c) and further clarified in the map plot in Figure 5.11(a).

With regard of the double filtering solution, Figure 5.11(b) shows a better concentration of the filtered estimated positions near the true UAV GNSS position with

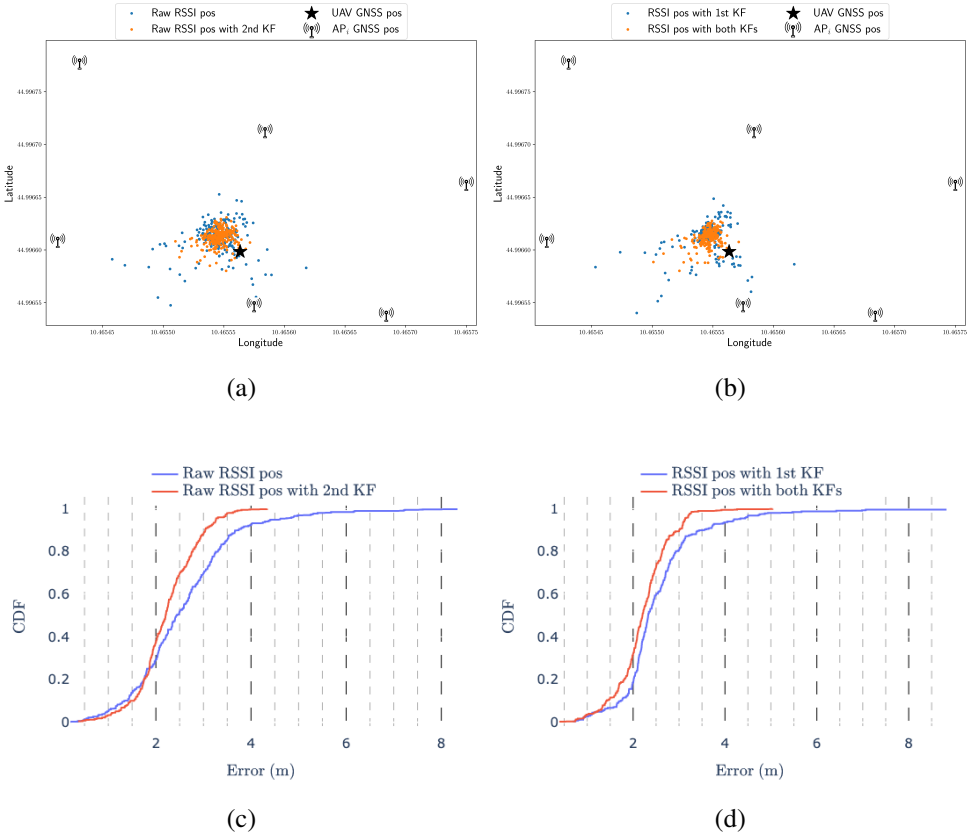


Figure 5.10: Evaluation of the implemented Wi-Fi-based localization system in the static hovering scenario with different filtering solutions: (a) *RAW RSSI position* and *RAW RSSI position with 2nd KF* experimental results; (b) *RSSI position with 1st KF* and *RSSI position with both KFs* experimental results; (c) CDF of *RAW RSSI position* and *RAW RSSI position with 2nd KF* experimental results; and (d) CDF of *RSSI position with 1st KF* and *RSSI position with both KFs* experimental results.

respect to Figure 5.11(a). This is confirmed by the CDF in Figure 5.11(d), as well as listed in Table 5.1, since the *FTM position with both KFs* solution achieves a mean positioning error of just 1.41 m, with a 95th percentile error equal to 3.22 m, thus achieving slightly better results than all the other aforementioned RSSI and FTM

filtering solutions. For the sake of comparison, the *RSSI position with 1st KF* only achieves a mean positioning error equal to 1.70 m, with the a 95th positioning error of 4.81 m, the second worst value for this metric in this static scenario. The obtained FTM results are significantly better than the expected *rr* for the Wi-Fi 2.4 GHz 40 MHz technology discussed in Subsection 5.3.2.

As expected, in the static scenario the FTM-based localization solution managed to achieve the best performance with or without filters with respect to RSSI-based solutions, with the double KF solutions able to further improve the localization accuracy of both the RSSI and FTM systems, but with the bigger gain on the FTM solution.

Mobile UAV With regard to the second experimental scenario, with the UAV flying along a square path, both RSSI- and FTM-based solutions achieve worse results with respect to the aforementioned static scenario. Starting with the RSSI-based approach, the *RAW RSSI position* solution significantly suffers from the movement of the UAV, as shown in Figure 5.12(a) and further confirmed by the values listed in Table 5.2. More in detail, the unfiltered localization solution achieves a mean positioning error equal to 5.49 m, with a 95th percentile error of 9.99 m. According to the CDF shown in Figure 5.12(c), it is possible to observe the beneficial effects of the *RAW RSSI position with 2nd KF* solution, which marginally reduces the mean error and 95th percentile error to 5.23 m and 9.06 m, respectively.

With regard to the *RSSI position with 1st KF* and *RSSI position with both KFs*, none of these approaches manages to improve the results of the *RAW RSSI with 2nd KF* solution, as confirmed by the mean error, in both cases closer to 5.35 m, reported by the comparison in Table 5.2 and by the CDF shown in Figure 5.12(b). In any case, in all the filtering approaches applied to the RSSI-based localization system, it is not clearly possible to understand (relying only on the RSSI-estimated coordinates) in which part of the pre-defined flight path the UAV is flying, hence making the use of this solution for localization purposes on a mobile target impractical.

Significantly better results are achieved by the FTM-based solution. As for the static hovering scenario, the *RAW FTM position* solution manages to obtain better

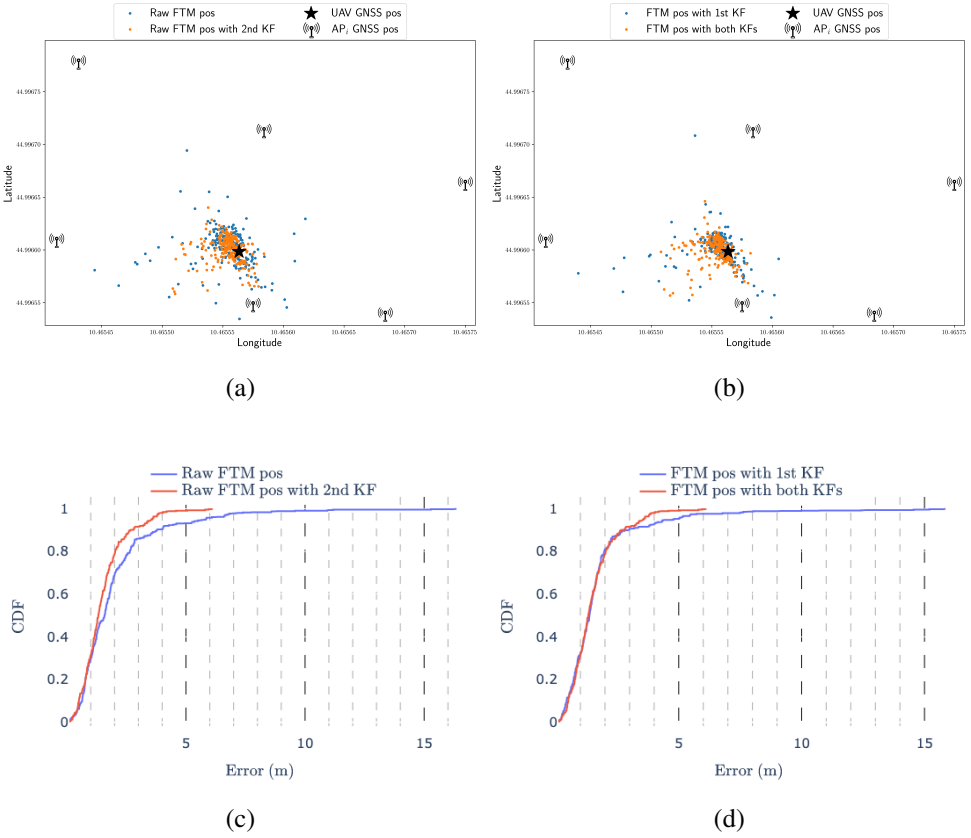


Figure 5.11: Evaluation of the implemented Wi-Fi-based localization system in the static hovering scenario with different filtering solutions: (a) *RAW FTM position* and *RAW FTM position with 2nd KF* experimental results; (b) *FTM position with 1st KF* and *FTM position with both KFs* experimental results; (c) CDF of *RAW FTM position* and *RAW FTM position with 2nd KF* experimental results;and (d) CDF of *FTM position with 1st KF* and *FTM position with both KFs* experimental results.

results with respect to all the RSSI-based solutions, as listed in Table 5.2. More in detail, without any filter, the FTM-based system achieves a mean positioning error of 3.76 m, with a 95th percentile error of 6.62 m. The unfiltered FTM-estimated positions are shown in Figure 5.13(a), together with the estimated positions computed

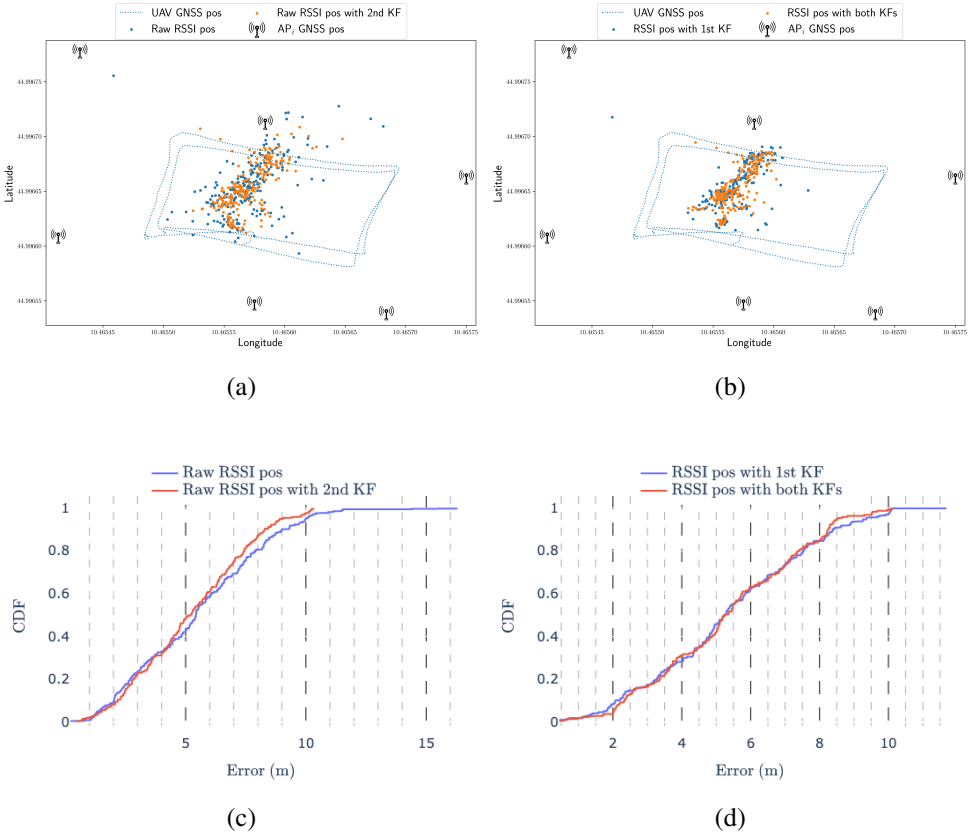


Figure 5.12: Evaluation of the implemented Wi-Fi-based localization system in the mobile scenario with different filtering solutions: (a) *RAW RSSI position* and *RAW RSSI position with 2nd KF* experimental results; (b) *RSSI position with 1st KF* and *RSSI position with both KFs* experimental results; (c) CDF of *RAW RSSI position* and *RAW RSSI position with 2nd KF* experimental results;and (d) CDF of *RSSI position with 1st KF* and *RSSI position with both KFs* experimental results.

adopting the *RAW FTM position with 2nd KF* solution. As visible in the CDF shown in Figure 5.13(c), the adoption of a KF after the multi-lateration algorithm, as in the *RAW FTM position with 2nd KF* approach, allows to significantly reduce the mean positioning error down to 3.32 m, with a 95th percentile error of 5.63 m.

Table 5.2: Experimental evaluation of RSSI- and FTM-based localization with the UAV in mobile condition with different filtering approaches.

Filtering Method	Mean Position Error (m)	95th Percentile Error (m)
RAW RSSI	5.49	9.99
RAW RSSI with 2nd KF	5.23	9.06
RSSI with 1st KF	5.37	9.35
RSSI with both KFs	5.34	8.52
RAW FTM	3.76	6.62
RAW FTM with 2nd KF	3.32	5.63
FTM with 1st KF	3.34	5.73
FTM with both KFs	3.30	5.70

With regard of the double filtering solution, Figure 5.13(b) shows a smaller number of outlier positions, as well as the estimated FTM-positions closer to the ground truth GNSS-based flight path followed by the UAV. This seems valid for both the *FTM position with 1st KF* and *FTM position with both KFs* solutions, as confirmed by the similar CDFs shown in Figure 5.13(d) and listed in Table 5.2. In fact, both pre-filtering and double filtering approaches manage to achieve a similar mean positioning error equal to 3.34 and 3.30 m, respectively, providing, at least analytically speaking, a marginal improving with respect to the *RAW FTM position with 2nd KF* solution. Similar 95th percentile positioning errors are also confirmed by the values listed in Table 5.2. In this case, the FTM-based localization results are aligned with the expected *rr* for the Wi-Fi 2.4 GHz 40 MHz technology discussed in Subsection 5.3.2.

With regard to the mobile scenario, despite the discussed positioning errors, it must be taken into account that the goal of the developed system is to provide an approximate knowledge of the position of the UAV, possibly flying in a BVLOS condition without a (or with a weak) GNSS signal reception. Therefore, behind the average positioning error (in terms of values), it is also relevant to evaluate the trajectories of the ground truth GNSS-based path with respect of the Wi-Fi estimated positions

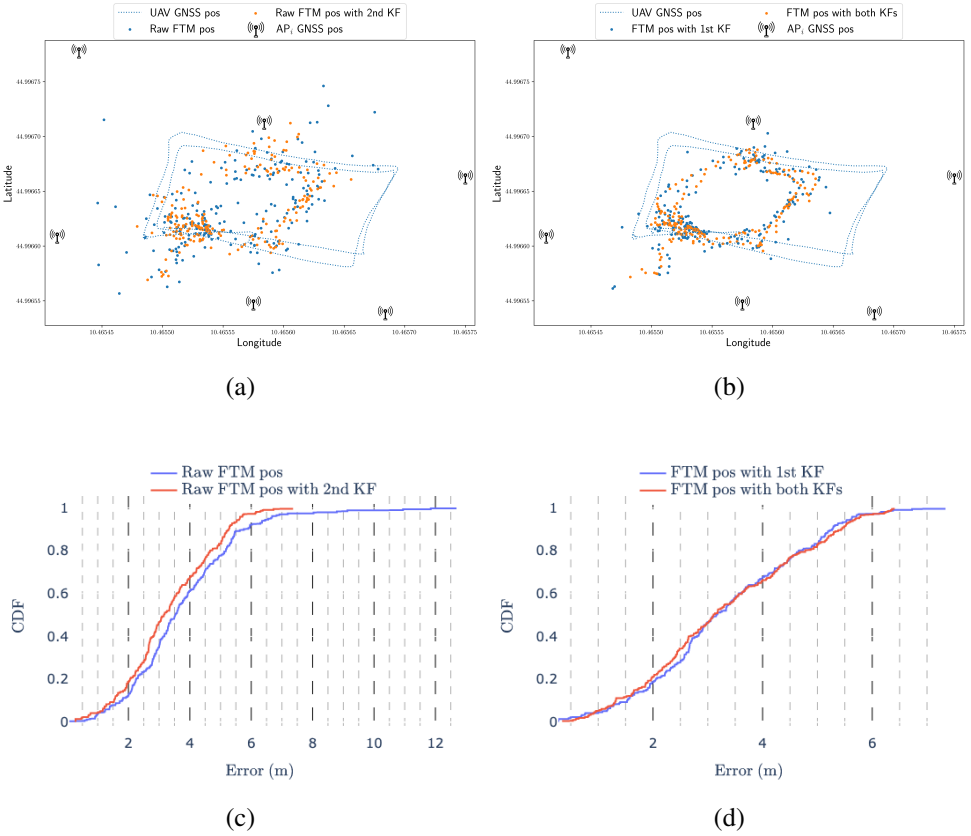


Figure 5.13: Evaluation of the implemented Wi-Fi-based localization system in the mobile scenario with different filtering solutions: (a) *RAW FTM position* and *RAW FTM position with 2nd KF* experimental results; (b) *FTM position with 1st KF* and *FTM position with both KFs* experimental results; (c) CDF of *RAW FTM position* and *RAW FTM position with 2nd KF* experimental results; and (d) CDF of *FTM position with 1st KF* and *FTM position with both KFs* experimental results.

of the proposed systems, in order to understand which approach can allow to better understand where the UAV is flying with respect to the its true position.

As shown in Figure 5.12(a), Figure 5.12(b), Figure 5.13(a), and Figure 5.13(b), related to the RSSI- and FTM-based positioning in the mobile scenario, respectively,

the first two images do not provide any reliable information to understand where the UAV is flying, while in the latter two images the estimated positions are distributed along the flight path. Therefore, the RSSI solutions are not suitable for real-time application on a UAV, at least with the proposed architecture and approach, while the FTM-based approach allows to estimate in real-time with significantly higher accuracy, even in mobile conditions. The best performance is achieved by the double filtering system: the estimated positions, shown in Figure 5.13(b), are mostly distributed on the ground truth flight path, indicating that the system is suitable to provide an approximate position estimate of the UAV in the case of the GNSS-outage, thus allowing the pilot to understand where the UAV is flying along the planned path.

Positioning Error vs UAV-APs Mean Distance

Another relevant metric to be used for evaluating the performance of the proposed Wi-Fi-based localization system is the relative positioning error with respect to the average target-AP distance, as proposed in [58].

According to this evaluation metric, in the *static scenario* the average UAV-AP distance, with the UAV hovering in a fixed position, is equal to $\overline{d}_{AP}^{STATIC} = 13.42$ m. Considering the RSSI-based localization system combined with the double KFs approach, which, according to Table 5.1, achieves a mean positioning error of 2.21 m, the relative position error (with regard to $\overline{d}_{AP}^{STATIC}$) is approximately equal to 16.67%. With regard to the FTM-based solution instead, given the mean positioning error of 1.41 m, the relative position error is approximately 10.65%.

Focusing on the *mobile scenario*, the mean distance is equal to $\overline{d}_{AP}^{MOBILE} = 14.01$ m. Therefore, according to Table 5.2, for the *RAW RSSI with 2nd KF* the relative position error (with regard to $\overline{d}_{AP}^{MOBILE}$) is approximately equal to 37.36%, while for the FTM-based solution with both KFs the relative position error is approximately 23.56%.

Update Frequency and Processing Times

Finally, since the system has been designed for real-time localization purposes, the measured positions update rate must be reported to further understand the capabilities

of the developed solution. As mentioned in Subsection 5.3.5, with six APs deployed in the test environment acting as FTM *responders*, the ESP32S3 can provide a list of Wi-Fi scan results with six FTM measurements and all the RSSIs of the nearby APs approximately every $\theta_{cycle} = 522$ ms.

More in detail, θ_{cycle} is defined as follows

$$\theta_{cycle} = \upsilon_{Wi-Fi} + n \cdot \nu \tag{5.45}$$

where: υ_{Wi-Fi} is the Wi-Fi scan time (dimension: [ms]), which is approximately equal to $\upsilon_{Wi-Fi} = 330$ ms; $n = 6$ is the number of APs; and ν is the FTM *Burst* duration (dimension: [ms]), which, given the parameters adopted in the proposed architecture, is equal to 32.

Therefore, considering the measured processing time introduced by the RPi4-handled filtering and multi-lateration, denoted as $\varkappa$ and approximately equal to 66 ms, the overall update period increases to approximately $\Theta_{cycle} = 588$ ms. This leads to a position update rate of $f_{pos} = 1/\Theta_{cycle} \approx 1.7$ Hz, which is sufficient for several real-time UAV applications.

5.3.7 Improvements

Several approaches might be implemented to further enhance the performance of the proposed Wi-Fi-based localization system. In detail, the following possible improvements have been identified for future research activities.

- Exploit the 5 GHz Wi-Fi band for RSSI-based distance estimation purposes, since the higher carrier frequency has different signal propagation ranges, typically smaller, thus providing a more granular measurement to aid the 2.4 GHz log-distance path loss model.
- Exploit other Wi-Fi-related signal quality metrics, such as the CSI, which can be used to estimate the AoA on Wi-Fi MIMO systems, increasing, however, both the cost and complexity of the system architecture.
- Utilization of 5 GHz 80 or 160 MHz Wi-Fi FTM equipments, in order to reduce the *rr* and achieve sub-meter level localization positioning error.

- Adoption of an EKF instead of a KF, as well as other possible different filters, with the aim to improve the filtering capabilities of the developed solution, thus improving the final localization accuracy.

- Adoption of a different algorithm for the localization task instead of the multi-lateration solution, such as, as an example, the Two-Step M-Estimator (TSME) [181], which might offer better noise removal capabilities, thus allowing to achieve a smaller positioning error with respect to the multi-lateration solution adopted in the developed solution.

- Evaluation of sensor fusion-oriented approaches to exploit both RSSI and FTM ranging solutions, aiming at developing LOS and NLOS channel estimation models suitable to improve the outliers removal filter and, therefore, enhance the range estimation, finally increasing the overall positioning accuracy.

5.3.8 Final Considerations

The developed and proposed real-time Wi-Fi-based system architecture discussed in Subsection 5.3.5 has been implemented and evaluated on a real UAV aiming at understanding which localization performance can be achieved with inexpensive IEEE 802.11mc 2.4 GHz COTS hardware, as well as adopting traditional and well-known filtering and ranging estimation techniques, discussed in Subsection 5.3.4 and Section 5.3.5. The obtained accuracy in mobile conditions with the FTM-based solution and presented in Subsection 5.3.6 is already suitable for environments with weak GNSS or GNSS-denied conditions, where the adoption of the UWB technology discussed in Section 5.2 might be too expensive (given the larger area) and non necessary (since sub-30 cm might be needed only on specific missions involving small UAVs).

Although this solution has limited use cases as temporary GNSS-backup solution to make aware the pilot where the UAV is flying with respect of the environment, the proposed further improvements identified in Subsection 5.3.7 might further enhance the performance of the developed system, thus opening new possible applications, all keeping the complexity and, therefore, the cost of the infrastructure low, since FTM

is being supported by most of the new released devices, even on the 5 GHz band, which should provide significantly better localization capabilities.

Finally, the study of the literature, as well as the data collected and evaluated during the design, implementation and experimental evaluation of the proposed Wi-Fi-based localization system, have been crucial to better understand the current limits and advantages of Wi-Fi for localization purposes, thus opening the exploitation of hybrid positioning solutions for UAVs and, more in general, robotics platforms operating in critical BVLOS environments, where, a backup and redundant solution are needed to ensure the success of the autonomous missions. This lead to the novel hybrid radio and LiDAR localization framework presented and discussed in Chapter 6, which fuses together the advantages of Wi-Fi technology with LiDAR-based positioning, providing a robust localization platform.

Chapter 6

Wi-Fi-LiDAR-based Localization

While Chapter 5 focuses on the use of different wireless protocols (namely, Wi-Fi and UWB) and different radio techniques aiming at performing an estimation of the distance between known anchors or APs (deployed in the environment) and the target node (mounted on the drone), so as being able to localize the UAV within the surrounding environment, this chapter presents a novel hybrid localization approach, aiming at developing a robust localization framework to be exploit on several robotics platform, as well as UAVs, for indoor localization in harsh environments.

In detail, the framework focuses on the combination of the Wi-Fi fingerprinting radio localization technique together with a SLAM algorithm processing the point cloud generated by the on-board LiDAR, in the end aiming at compensating the limitation of both technologies and developing a robust localization framework, able to operate without the need to know the exact position of the APs, unlike the systems proposed in Chapter 5.

6.1 Overview

Indoor localization has always been a crucial asset for a plethora of applications, spacing from the user tracking within the rooms of a building to the localization of several kind of robotics platforms, including compact-size UAVs, for applications in

GNSS-denied environments, where the only feasible way to localize a device consists in relying on different SLAM algorithms, mainly based on the use of LiDAR, inertial odometry, or visual-based SLAM, as well as on alternative radio localization systems. However, in the last decade, the use of the Wi-Fi has further spread, enriching the overall availability and coverage of the wireless networks, therefore making more feasible and easy to exploit such existing networks for different applications, like for localization purposes.

While the traditional multi-lateration approaches are suitable for all those application where the environment is well know, as well as the APs positions, there are several applications in which the knowledge of the environment is pretty limited and it is very complex (or almost impossible) to associate geo-spatial coordinates to the APs installed in the environment itself. Therefore, the research community has been investigating some alternative radio localization techniques, such as the Wi-Fi fingerprinting, where the observed wireless radio coverage of the environment (in terms of RSSI and detected MAC addresses) is associated to the coordinates where the scan has been performed, thus allowing to build a Wi-Fi heat map to be then used for localization purposes.

The novel framework described in this doctoral thesis associates Wi-Fi scanning data with the SLAM algorithm processing LiDAR and IMU data, thus allowing to:

- automatically build a Wi-Fi fingerprints heat map through a robotic platform enabled by a LiDAR-based SLAM algorithm;
- use the built Wi-Fi fingerprints map to aid the SLAM solution in environments characterized by repetitive patterns (e.g., a tunnel or a corridor);
- use the LiDAR SLAM to significantly improve the localization accuracy, overcoming the several meters error introduced by traditional pure Wi-Fi fingerprinting-based algorithms.

More in detail, this chapter focuses on the need to provide redundancy and enhance the reliability of localization systems, in detail by incorporating additional data arriving from different sensors sources, like, as in this case, an on-board Wi-Fi scanner, allowing to use as additional feature the data related to the RF domain, to be

exploited for localization purposes, thus allowing the system to benefit from complementary information sources.

Given these remarks, in Section 6.2 the current limitations of both LiDAR and Wi-Fi fingerprint are discussed, showing how the constraints of one technology caould be compensated by the strength of the other technology. Section 6.3 details the proposed framework architecture, discussing the framework's components, while Section 6.4 presents the test methodology and preliminary results of the novel proposed localization framework in different environments. Finally, Section 6.5 discusses some possible improvements to be applied to the framework, in order to further increase the overall accuracy, while in Section 6.6 final remarks are drawn.

6.2 SLAM and Wi-Fi Fingerprinting Limitations

The ability of a robotic platform to operate within a global map is essential for successful exploration and navigation missions. To this end, a global map provides robots with valuable information for various tasks, such as path planning, coordination of multiple robots, and localization of objects and survivors during Search and Rescue (SAR) missions. However, in these complex scenarios, localization algorithms can face temporary failures, as an example due to the presence of dust particles or drifting, that can cause these algorithms to temporarily lose accuracy and potentially affecting the ability of the robots to effectively carry out their mission objectives.

While SLAM algorithms applied to the point cloud collected by a LiDAR can already achieve a good level of accuracy in many environments, they are still affected by the loop-closure [182] issue and present poor performance in environments where dust and repetitive patterns in the surrounding are present, such as, for example, in a mining tunnel with walls made of rocks or in corridors with recurrent windows, doors or other fixed elements. While LiDAR-based systems offer advantages in terms of immunity to appearance changes and illumination with respect to traditional cameras, these sensors can also face difficulties in accurately capturing certain types of surfaces, such as transparent or reflective objects. Moreover, adverse weather conditions (such as rain, fog, or dust [183]) can affect their performance. These problems

can heavily affect autonomous mobile robotic missions, where the platform has to localize itself in a global map previously built aiming to complete the assigned task, since a failure of the localization estimation can lead to a failure in the mission, thus requiring an external human intervention.

So as, since Wi-Fi networks are nowadays deployed in almost every working environment (even in the most modern underground mine tunnels), a possible way to significantly reduce the risk of localization-related issues in autonomous missions might be further mitigated by enhancing LiDAR-based algorithms with the easy to integrate Wi-Fi fingerprinting techniques, thus allowing to integrate the point cloud maps with RF data as a *fourth dimension*. On the other hand, the usage of only Wi-Fi for localization cannot provide a strong enough localization accuracy suitable for robotics application, since most of the existing fingerprinting algorithms investigated in Chapter 1 can achieve only a few meters of accuracy, while RSSI- and FTM-based solutions, like the one developed and presented in Chapter 5, are limited to a positioning error of several meters.

6.3 System Architecture

In order to solve the limitations detailed in Section 6.2, the proposed hybrid localization framework, whose schematic representation is shown in Figure 6.1, can be divided into four main components: (i) LiDAR-based place recognition; (ii) Wi-Fi fingerprinting; (iii) best candidate selection; and (iv) ICP Point Cloud Registration.

With regard to the LiDAR-based place recognition, the framework approach is similar to the one proposed in [78], since the transformation to the image space keeps the computational effort low. More in detail, the preliminary step is to transform the incoming LiDAR scans $\mathscr{P}_t$ in depth images $\mathscr{I}_t$ through a bi-dimensional spherical projection. The images are then fed into the *Descriptor Extraction* module, that outputs the 2×64 vectors $\vec{q}$ and $\vec{w}$. In particular, $\vec{q}$ is an orientation-invariant vector encoding place-dependent information and primarily used for querying similar point clouds, while $\vec{w}$ is an orientation-specific vector, responsible for regressing the yaw discrepancy between two point clouds.

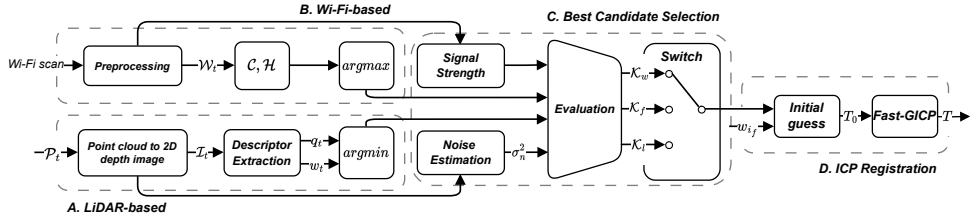


Figure 6.1: Overview of the proposed framework four main components. (A) LiDAR-based place recognition; (B) Wi-Fi fingerprinting; (C) best candidate selection; and (D) Iterative Closest Point (ICP) Point Cloud Registration.

So as, when new LiDAR observations are fed into the system, they are processed to extract the vector sets Q and W as follows:

$$Q = \{\vec{q} \in \mathbb{R}^{64}, k \in \mathbb{N} : \vec{q}_1, \vec{q}_2, \ldots, \vec{q}_k\} \tag{6.1}$$

$$W = \{\vec{w} \in \mathbb{R}^{64}, k \in \mathbb{N} : \vec{w}_1, \vec{w}_2, \ldots, \vec{w}_k\}\,. \tag{6.2}$$

Then, a query to the global map is done to recognize the effective place candidate. This process can be described as a minimization problem where a *k-d* tree is constructed with the vector set Q and is searched through with the current vector q_t to find the pair with the minimum distance in the q vectors space as

$$i = \arg\min_{i \in \mathbb{N}} f(Q, q_t)\,. \tag{6.3}$$

Moreover, according to the robotics and computer vision good practice, the framework keeps the *top-k* candidates, denoted as $\mathscr{K}_l = \{i_{l,1}, i_{l,2}, \ldots, i_{l,k}\}$, corresponding to the indexes of the most similar places.

The second component is the Wi-Fi fingerprinting system, whose integration details are discussed in Section 6.4. Hence, given the observations $\mathscr{W}$ and the current Wi-Fi scan $\mathscr{W}_t$, the goal is to find the pair with the strongest correlation. While a Wi-Fi scan contains multiple attributes, only the MAC addresses of the scanned APs, denoted as $A = \{a_0, a_1, \ldots, a_N\}$, and their corresponding RSSI values, denoted as $S = \{s_0, s_1, \ldots, s_N\}$ are currently used. Therefore, the Wi-Fi scan can be denoted as

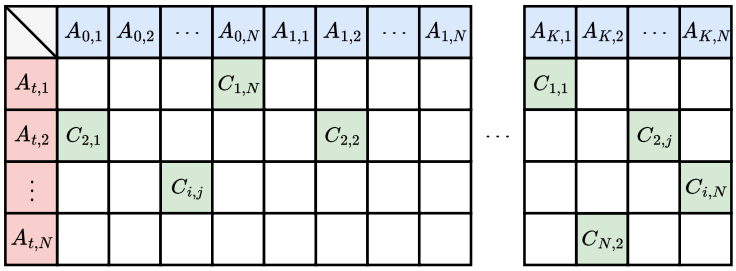


Figure 6.2: Example of correlation matrices $\mathscr{C}_{i,j}$, where the empty cells correspond to the case $A_{t,i} \neq A_{k,j}$, while the colored cells correspond to the case $A_{t,i} = A_{k,j}$.

$\mathscr{W}_t = \{A_t, S_t\}$. The correlation between the observations $\mathscr{W}$ and the current scan $\mathscr{W}_t$ is given by the matrix $\mathscr{C}_{N\times N\times K}$, where K is the total amount of observations in $\mathscr{W}$. An example of the correlation matrix is depicted in Figure 6.2.

The goal is to find the pair $(\mathscr{W}_k, \mathscr{W}_t)$ with the highest correlation by summing up each matrix $\mathscr{C}_{k,i,j}$, yielding the vector $\mathscr{H}_{1\times K}$, denoted as

$$\mathscr{H}_k = \sum_{i,j=0}^{i,j=N} \mathscr{C}_{k,i,j} \text{ for } k = 0,1,\ldots,K\,. \tag{6.4}$$

In the case of the Wi-Fi fingerprinting, the *top-k* candidates are the ones with the maximum correlation sum, so as

$$i = \arg\max_{i\in\mathbb{N}} \mathscr{H}\,. \tag{6.5}$$

Similarly to the LiDAR-based place recognition, the *top-k* candidates, denoted as $\mathscr{K}_w = \{i_{w,1}, i_{w,2}, \ldots, i_{w,k}\}$, are then kept.

At this point, the third component, namely the best candidate selection, kicks in, performing a quick evaluation of each candidate in order to add extra resiliency in the overall architecture. With regard to the LiDAR-based candidates, the presence of noise in the range images $\mathscr{I}_t, \mathscr{I}$ is estimated according to the technique proposed in [184]. In detail, in the case the noise variance $\sigma_n^2 \geq \sigma_{threshold}^2$, then the candidates are considered a liability and are discarded. Instead, for the Wi-Fi-based candidates, the correlation scores $C_{k,i,j}$ are evaluated., so that if the current RSSI values S_t are

considerably lower than the anchor values S_k, then the candidate should be disregarded due to low signal strength. As a final operating check, a switching mechanism has been deployed: in the case that the current LiDAR scan is too noisy, the candidate selection switches to Wi-Fi one only; vice-versa, if the Wi-Fi signal strength is too low, the hybrid system switches to the LiDAR candidates only. After the initial candidate evaluation step, it is assumed that the candidates with the highest likelihood $\mathscr{K}_f$ are the ones that exist in both sets, as follows:

$$\mathscr{K}_f = \begin{cases} \mathscr{K}_w & \sigma_n^2 \geq \sigma_{threshold}^2 \\ \mathscr{K}_l \cap \mathscr{K}_w & otherwise \\ \mathscr{K}_l & S_t \geq S_{threshold}\,. \end{cases} \tag{6.6}$$

The final component of the proposed framework, namely the ICP, it is responsible for aligning the gathered sets of point clouds to find their relative pose or transformation, a crucial step of SLAM algorithms. In detail, the Fast-GICP [185] point cloud algorithm has been adopted.

6.4 System Integration and Experimental Evaluation

The experimental evaluation of the proposed system has been divided into two phases: an *offline phase* and an *online phase*. In the *offline phase*, the environment is scanned with the platform equipped with both on-boad LiDAR and Wi-Fi scanners, gathering the point cloud of the environment together with the Wi-Fi data measurements. Then, these data are associated with the odometry calculated by the robotic platform and, at the end of this survey, the final map of the environment is built. Then, once the data have been gathered, the Wi-Fi fingerprints database is generated, associating each Wi-Fi scan with the position (x, y, z) in the built map. Instead, in the *online phase*, the re-localization framework, running on board the robotic platform, uses the LiDAR and Wi-Fi scanner as the input data to localize itself on the built map and to start the new mission. The proposed solution has been developed aiming at minimizing both weight and space required by the overall system, thus allowing the integration on several possible robotics platforms, spacing from wheeled rovers, legged robots

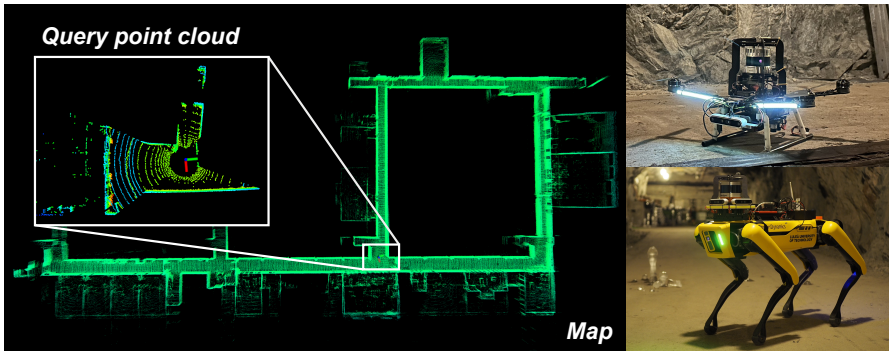


Figure 6.3: point cloud map generated during the experimental evaluation of the system, and showcasing some mobile robots equipped with the proposed solution.

(e.g., Boston Dynamics Spot) and even compact UAVs (e.g., Holybro X500 V2), as shown in Figure 6.3. In detail, the developed system architecture can be divided into two main hardware components, the IEEE 802.11 2.4 GHz Wi-Fi scanner and the 3D LiDAR scanner responsible for the localization through a SLAM algorithm, both detailed in the following.

6.4.1 2.4 GHz Wi-Fi Scanner and 3D LiDAR

In order to collect all the information regarding the Wi-Fi networks used to build the fingerprints' database, a suitable compact Wi-Fi scanner has been developed. Given the need to minimize the device footprint, on the hardware side, as Wi-Fi scanner compact-size development board based on the ESP32 SoC, which is able to provide both IEEE 802.11 Wi-Fi and Bluetooth connectivity on the 2.4 GHz RF band, has been used. Despite the Wi-Fi connectivity built-in on the reference robotics platforms, the use of an external (additional) adapter has been beneficial (i) to have a detachable solution deployable on different platforms, (ii) to avoid to impact the built-in Wi-Fi connectivity, generally used to stream the gathered data through the Wi-Fi mesh showcased in Chapter 3, and (iii) to minimize the measurement discrepancy introduced by the use of different antennas and Wi-Fi modems, thus allowing to collect cleaner and more reusable data for several experiments. The use of the ESP32 pro-

grammable micro-controller allows to fine tune the Wi-Fi networks scanning process. In fact, a complete scan of all the 2.4 GHz Wi-Fi channels—13 in Europe—requires a relevant amount of time to complete. The Wi-Fi networks scanning process consist in two types of scan: (i) *active scan* and (ii) *passive scan*.

The *active scan* is initiated by the Wi-Fi client device, which actively sends out a *probe request frame* on all the available channels. When nearby APs receive these frames, they respond with a *probe response frame* containing all the essential information regarding the AP, including its SSID, supported security protocols, and signal strength. Then, the active scanning is useful when a client wants to connect to a specific network or actively search for a network. However, it consumes additional power on the client device because it continuously transmits *probe request frames* and processes *probe responses*.

On the other end, a *passive scan* is initiated by APs that continuously broadcast *beacon management frames* at regular intervals, typically every 100 ms. These frames contain information about the network, including SSID, security settings, and supported data rates. Then, Wi-Fi client devices passively listen for these *beacon management frames* transmitted by nearby APs without sending any active requests, thus increasing the efficiency of the process on their side. However, passive scanning provides less control to clients when searching for specific networks. Clients can only detect and connect to networks whose frames they receive. Additionally, updates about available networks might not be as immediate because frames are usually broadcast at fixed intervals.

In summary, active scanning is proactive and gives clients control over network discovery, making it suitable when they need to connect to a specific network. Passive scanning is more energy-efficient and relies on APs continuously broadcasting *beacon management frames*, making it suitable when clients can be patient in discovering available networks. Most Wi-Fi devices use a combination of active and passive scanning to balance energy efficiency and network discovery.

Using the default ESP32 parameters, a Wi-Fi scan of all the 2.4 GHz channels requires approximately 2040 ms, achieving an update frequency of the Wi-Fi data approximately equal to 0.5 Hz (on average). In order to reduce the Wi-Fi scan time

and thus increase the update frequency, allowing to collect more data while moving the various robotic platforms and also to make the *offline phase* less tedious and faster, in the custom-made firmware developed for the ESP32 board, the Wi-Fi *active scan* time for the channels 1-11 has been reduced from the initial 120 ms to 85 ms, while the *passive scan* time for channels 12 and 13 has been reduced from 360 ms to 255 ms, thus reducing the average complete scan time to approximately 1445 ms and consequentially slightly increasing the update frequency of the Wi-Fi data to approximately 0.69 Hz (on average). Moreover, to collect more data, the detection of hidden SSID is enabled, allowing to detect both RSSIs and MAC addresses of the hidden networks. All the gathered data (namely: SSID, MAC address, RSSI and channel of each detected AP) from the Wi-Fi scanner are properly sent to the robotic platform companion computer through the USB serial port and, then, published to the custom ROS topic.

Regarding the LiDAR, two different options have been used in the two different test environments. In the *first* experiment, the chosen LiDAR scanner is the Velodyne Puck Hi-Res, running at 10 Hz frequency and featuring 16 channels with a 20° vertical Field Of View (FOV) and a 360° horizontal FOV. In the *second* experiment, the Ouster OS1-32 LiDAR, running at 10 Hz frequency and featuring 32 channels with a 45° vertical FOV and a 360° horizontal FOV has been used. Along with the integrated IMU unit, they are used to run DLO [186], a SLAM algorithm responsible for providing the odometry. For the 3D LiDAR-based place recognition, discussed in Section 6.3, a model similar to OREOS [78] has been trained, but with a different backbone model. In particular, instead of using a 3 layer Convolutional Neural Network (CNN), a ResNet18 is adopted in order to get a deeper and more sophisticated feature extraction, assuring a robust performance even with sparse LiDAR scans. The model has been trained using data collected from multiple runs in the underground corridors of Luleå University of Technology (LTU), Sweden, and a real-life underground mine facility located in Luleå, Sweden.

6.4.2 Experimental Evaluation

The experimental evaluations took place in two different environments with two different platforms. In the *first* scenario, the framework has been tested in the underground corridors of LTU with a Boston Dynamics Spot robot equipped with the aforementioned sensors, featuring an Intel NUC on-board computer with an Intel Core i5-10210U and 8 GB of RAM. This urban environment offers long, self-similar corridors with some parts containing glass, doors and various objects. Being part of the university, the corridors are populated with fixed Wi-Fi APs as well as temporary APs from various devices. During this experiment, the robot traversed approximately 500 m at a constant speed of 5 km/h.

The *second* experiment took place in a modern underground mining facility, being fully equipped with a Wi-Fi infrastructure composed of several APs. The same sensors have been mounted on top of a vehicle, which then navigated through the mine at various speeds (between 15 km/h and 20 km/h) for a total distance of approximately 1 km. In both these experiments, the re-localization performance are tested continuously as the robotic platform navigates the environment. The sampling speed of the LiDAR and the Wi-Fi scanners is determined by the slowest one, which in both cases is the Wi-Fi scanner. Therefore, the results presented below are from a continuous evaluation for every sampling step defined by the rate of the Wi-Fi scanner.

The results are shown in Figure 6.5, Figure 6.6, Figure 6.7, and Figure 6.4, which highlight the best candidates and the trajectory followed by the robotic platforms during the experiments, respectively. Moreover, for the sake of clarity, the results of the three experimental setups, namely the LTU data-set, the LTU data with added noise and underground mine, are shown in Table 6.1, Table 6.2 and Table 6.3, respectively. Moreover, the adopted metrics include (i) the recall score for the increasing number of nearest place candidates retrieved from the map, and (ii) the mean and standard deviation of the distance from the current pose to the one queried from the database. The latter metrics heavily relies on the recall score and on the density of the sampling space.

Focusing on the results with the *first* experiment at the urban indoor environment, shown in Figure 6.5 and Table 6.1, it can be observed how the LiDAR-based

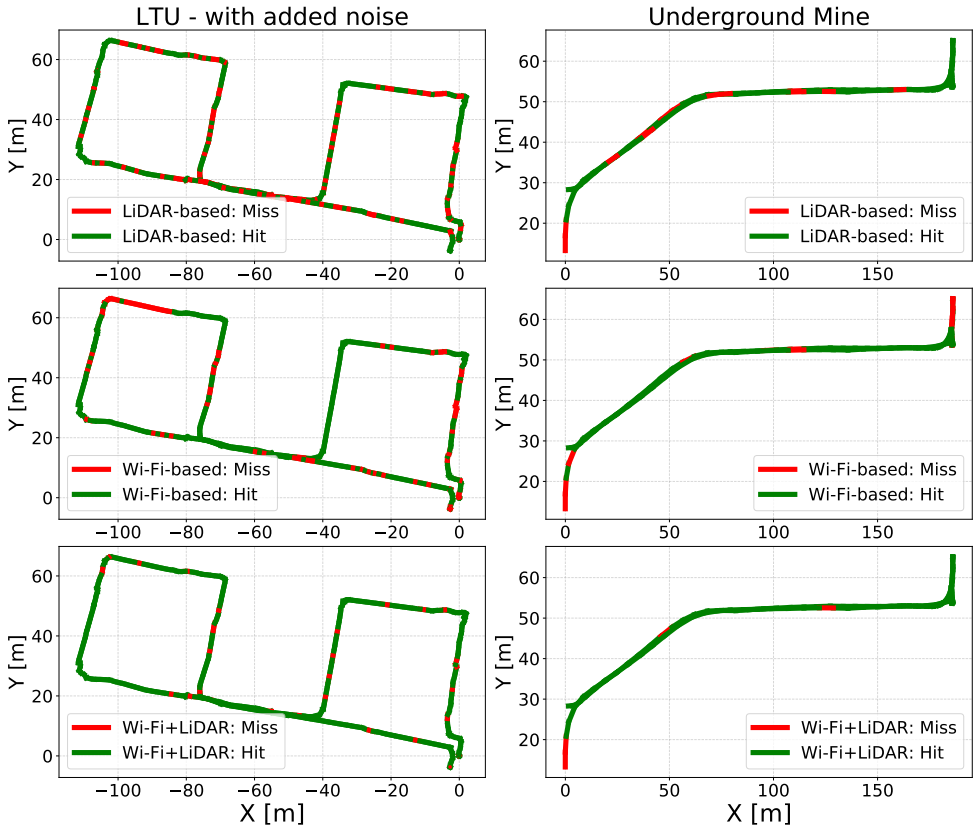


Figure 6.4: Trajectory for each experiment, based on the *top-1* candidate (green points represent the correct predictions, red points denote the incorrect ones).

method demonstrates a good performance in terms of recall percentage. On the other hand, the mean distance of the predicted pose, compared to the ground truth within the map, is significantly high, especially for the increasing number of *top-k* candidates. On the contrary, for the Wi-Fi fingerprinting, the mean distance is maintained low (around 3 m, increasing to approximately 5-10 m as expected for the less likely candidates). Then, the proposed switching solution is able to combine the best performance achieved by both approaches (namely, the good recall percentage of the LiDAR-based method and the lower mean distance error of the Wi-Fi fingerprint-

Table 6.1: LTU (w/o added noise) recall score, mean distance error and standard deviation for the *top-1* candidate, as seen on Figure 6.5.

	RECALL (%)	MEAN (m)	STD (m)
LiDAR-based	88.7	6.8	16.2
Wi-Fi-based	79.8	4.2	13.1
LiDAR+Wi-Fi	**96.9**	**3.3**	**9.5**

ing), finally resulting in a 97% recall and a mean error equal to 3.3 m for the best candidate selection. This performance gain is explained by the nature of the Wi-Fi fingerprinting, being able to narrow down the search space, compared to LiDAR, that is prune to perceptual aliases.

In tri-dimensional point clouds, descriptors are used to capture and represent the unique features of each point. However, it is possible for two descriptors to be similar despite if originating from different locations, in the case the underlying features of the point cloud, such as shape or texture, exhibit similarities. This similarity can occur because the descriptors are designed to extract and encode relevant information about the point cloud, allowing them to capture similar patterns or characteristics regardless of their spatial origin. The deterministic nature of the Wi-Fi fingerprinting overcomes this issue by removing the outlier candidates, and the combination of both yields the best result.

To further evaluate the performance of the framework, the performance of the LiDAR are degraded by adding Gaussian Noise at 40% of the samples with $\sigma_n^2 = 0.015$. As shown in Figure 6.6, the performance drop is approximately 20% for the recall percentage and the mean error distance from 7 m to 16 m. At the opposite, with the combined method the decrease is only 6%, demonstrating the ability to decide between the best candidates in the presence of disturbance.

The *second* experiment took place in a underground mining facility that contains long featureless tunnels with multiple drifts. As shown in Figure 6.7, the results of the switching solution further validate the performance of the framework. In detail, the inclusion of the Wi-Fi can narrow down the search space, and the resulting common

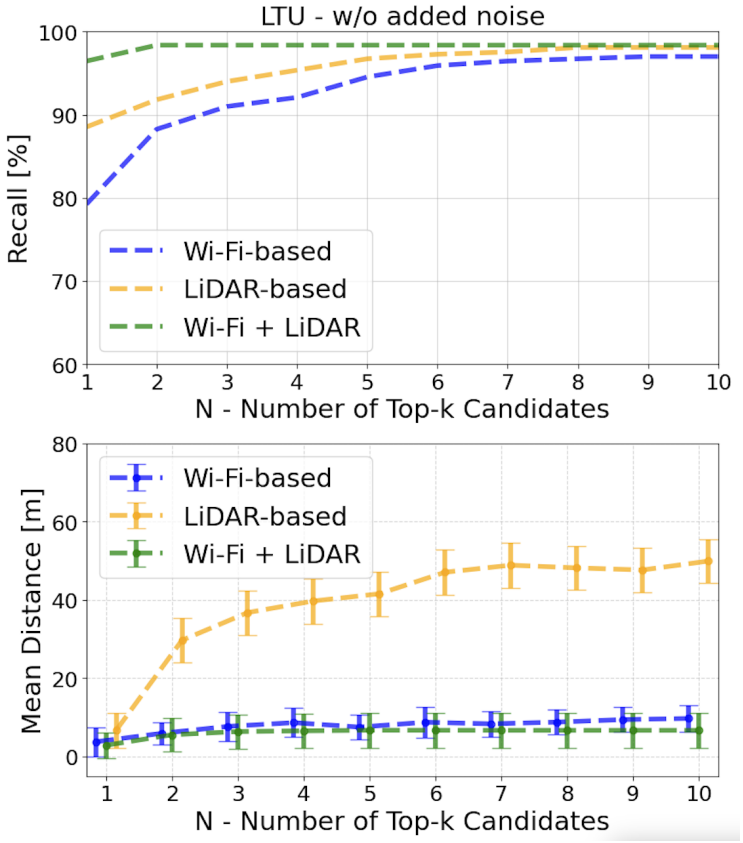


Figure 6.5: Experimental results in terms of place recognition for an increasing number N of nearest place candidates (top plot). A candidate is considered correct if it is within a 3 m radius from the corresponding pose in the database. Bottom plot demonstrate the mean distance of the predicted pose to the ground truth from the map, for each candidate, with the addition of the min and max deviations.

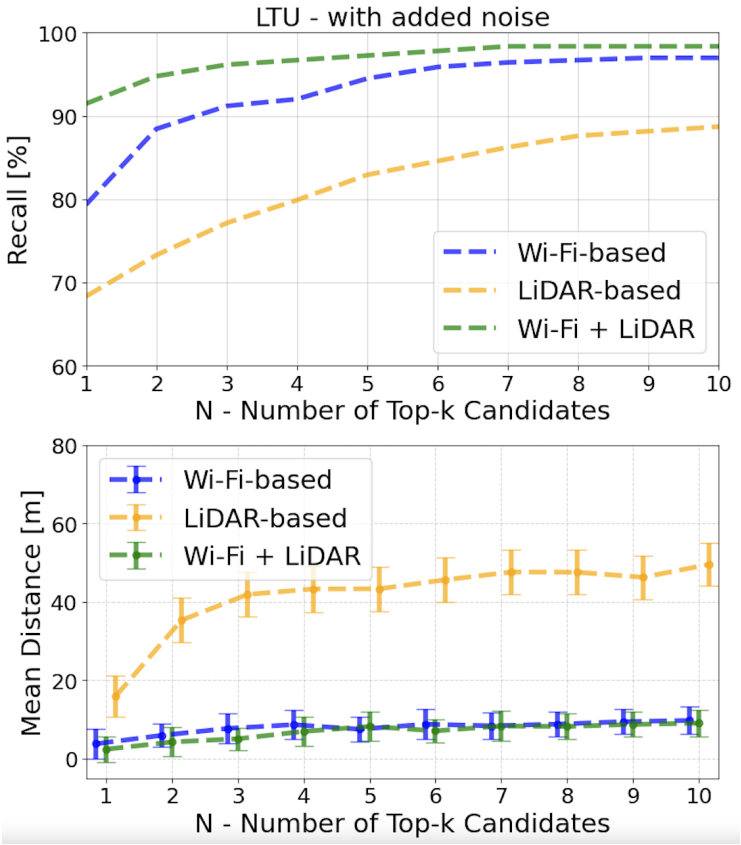


Figure 6.6: Experimental results in terms of place recognition for an increasing number N of nearest place candidates (top plot). A candidate is considered correct if it is within a 3 m radius from the corresponding pose in the database. Bottom plot demonstrate the mean distance of the predicted pose to the ground truth from the map, for each candidate, with the addition of the min and max deviations.

Table 6.2: LTU (with added noise) recall score, mean distance error and standard deviation for the *top-1* candidate, as seen on Figure 6.6.

	RECALL (%)	MEAN (m)	STD (m)
LiDAR-based	68.6	16.4	24.9
Wi-Fi-based	79.8	4.2	13.1
LiDAR+Wi-Fi	**91.4**	**3.8**	**10.2**

Table 6.3: Underground mine recall score, mean distance error and standard deviation for the *top-1* candidate, as seen on Figure 6.7.

	RECALL (%)	MEAN (m)	STD (m)
LiDAR-based	86.6	7.9	25.1
Wi-Fi-based	81.0	3.8	4.7
LiDAR+Wi-Fi	**93.8**	**3.4**	**11.6**

candidates yield a 93% recall score and a mean error distance approximately equal to 3.4 m. The high recall score for the LiDAR-based method is due to the higher resolution of the LiDAR scanner, which, in this case, was the OS1-32 and had a number of channels double with regard to the VLP16. Finally, as shown Figure 6.4, the red mismatched places are corrected, as long as at least one of the components predicted it right.

6.5 Future Improvements

There might be several options to improve the accuracy of the proposed framework. For example, in order to increase the amount of Wi-Fi data collected during the *offline* phase of the fingerprinting process, multiple scanners based on the same development board could be used, therefore improving both the update frequency of the RF data—thus allowing to the robotic platform to move faster—and to detect more networks, increasing the amount of features used then by the trained model. Moreover, over the last decade, the Wi-Fi technology has evolved with the aim to improve both the

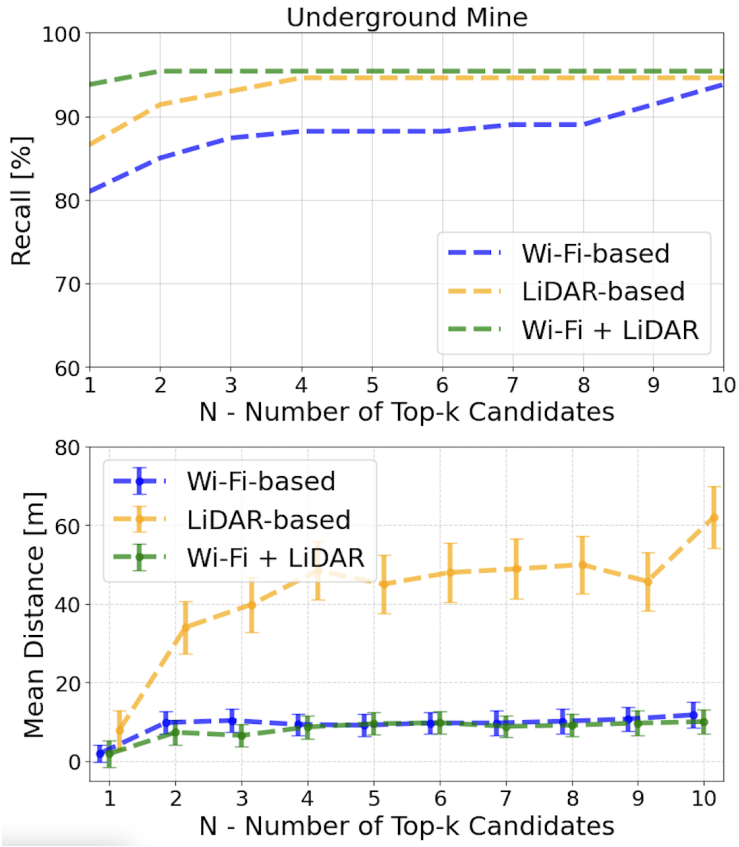


Figure 6.7: Experimental results in terms of place recognition for an increasing number N of nearest place candidates (top plot). A candidate is considered correct if it is within a 3 m radius from the corresponding pose in the database. Bottom plot demonstrate the mean distance of the predicted pose to the ground truth from the map, for each candidate, with the addition of the min and max deviations.

coverage and throughput of the networks. In order to achieve the latter result, newer Wi-Fi standards introduced the use of additional bands alongside the most common 2.4 GHz, with the aim to provide more bandwidth and thus significantly improve the throughput of the network, allowing to achieve data rates of several Gbps. More in detail, IEEE 802.11n, released in 2009 and known as Wi-Fi 4, introduced the use of the 5 GHz band. However, only with IEEE 802.11ac, released in 2014 and known as Wi-Fi 5, the 5 GHz band has become widely used. Furthermore, the last available IEEE 802.11ax, known as Wi-Fi 6, released in 2019 and extended to the Wi-Fi 6E in 2020, has introduced the new 6 GHz band to the Wi-Fi networks, further pushing the overall throughput capacity of such networks.

According to these advancements, in an enhanced version of the framework, a new Wi-Fi scanner able to scan these new Wi-Fi bands, especially the 5 GHz band, could be featured, thus allowing a more *granular* measurement of the RF spectrum, further aiding the 2.4 GHz fingerprinting solution. Besides Wi-Fi networks, even the new generation of cellular networks, like the 5G, are getting more and more diffuse, with the deployment of several new ground BTSs of increasingly smaller size becoming popular for indoor private 5G networks deployed for industrial applications in different kind of working environments. Given the different radio technologies used by the cellular networks and their adoption increase, an enhanced version of the proposed system could integrate the cellular network fingerprint to further enhance the accuracy of the system. Finally, more advanced ML techniques can be adopted [187,188], allowing a better usage of the gathered data to further increase the accuracy.

6.6 Final Considerations

The proposed hybrid localization framework combining LiDAR SLAM localization together with Wi-Fi fingerprinting showed some promising results, allowing to overcome some localization challenges of the current technologies. The preliminary experimental evaluations, as shown in Section 6.4, are promising, highlighting the framework's potential for localization applications in environments where pure

LiDAR solutions may encounter technological limitations. Thus, by exploiting the strengths of both LiDAR SLAM and Wi-Fi fingerprinting, the framework consistently achieves improved performance, especially in challenging environments, such as the urban indoor and underground mining scenarios used to validate the effectiveness of solution, where traditional methods fail or suffer several problems described in Section 6.2. The proposed framework holds promise for practical implementation in inspection and monitoring scenarios, empowering mobile robots with accurate global localization capabilities.

Chapter 7

Conclusion

This doctoral thesis aims at supporting the development of UAVs-related applications in multiple real scenarios and environments, improving and enabling novel communications and localization technologies, all adopting lightweight and affordable COTS components combined together with the novel developed software architectures and algorithms.

Two critical topics have been covered in this doctoral thesis: UAV-to-X connectivity and GNSS-denied localization techniques.

Regarding the *first* topic, in order to allow different information flows involving one or more UAVS, as well as other parties, like IoT nodes, Internet infrastructure, humans and ground located devices, a MIG has been properly designed and developed in Chapter 2, taking into account the required scalability and features needed to allow its applicability to different scenarios, spacing from traditional IoT applications, as well as more advanced and complex aerial IoT scenario, followed by multi-link connectivity for UAVs' related use cases. Moreover, the MIG has been investigated also in terms of performance capabilities, with the derivation of a novel Markov chain-based framework together with a Python-based simulator, presented in Section 2.3, suitable to verify and simulate the performance of the MIG under different workloads, thus allowing to predict the constraints of the proposed MIG in different scenarios without the need to perform a real testbed deployment, therefore

allowing to significantly speedup the development process of new use cases. Moreover, an extended version of the MIG, denoted as aerial MIG, has been developed to be integrated on a UAV platform.

The developed aerial MIG has been also the starting point of two further research activities related to UAVs' connectivity. In detail, based on the protocols integrated in the MIG, a hybrid opportunistic mesh networking based on a combination of LoRa, LoRAWAN, and IEEE 802.11s networks layers, allowing to enable complex UAV's swarming scenario operating in different condition and with different requirements, spacing from long-range, low throughput missions, as critical high throughput missions, operating in smaller areas. The defined hybrid mesh, detailed and discussed in Chapter 3, has been illustrated, describing the protocols rules and possible network topologies involving several components of a UAV swarm. Moreover, the experimental performance of a field deployable fully WMN have been investigated in a high throughput and low latency critical mission involving a UAV and a wheeled rover exchanging real-time point cloud data gathered from the on-board data, showing promising results for complex BVLOS missions, where a third-party network is needed to guarantee a reliable communication link. However, despite the achieved results, according to the use case and its related considerations, the need to rely on a more structured third-party network infrastructure emerged, opening to the investigation of how to integrate and use cellular connectivity on UAVs, which has been in deep discussed in Chapter 4.

Regarding cellular connectivity, after performing preliminary *in-flight* evaluation on the usage of 4G LTE network in the developed aerial MIG for UAV-to-X connectivity purposes, this doctoral thesis *first* investigated the behaviour of 4G cellular networks at different flight altitudes. *Then*, aiming at improving the cellular connectivity's reliability, the designed aerial MIG has been exploited as a starting point for the architectural design and implementation of the novel SSAS solution, discussed in Chapter 4. In detail, the system's integration, design and evaluation has been discussed in this doctoral thesis, in order to validate the benefit of such solution applied on a UAV to enable BVLOS missions where the reliability of the cellular communication link is crucial. The achieved results confirmed the overall functioning of the

SSAS, as well as opening to new research extensions of the proposed SSAS system.

Instead, the second part of this doctoral thesis investigated the critical aspect of real-time GNSS-denied localization techniques, mainly adopting radio-based solutions as well as LiDAR-based SLAM approaches. In detail, Chapter 5 presented the exceptional results related to the adoption of a COTS UWB-based localization system applied to a real UAV, followed by the architectural design of a low cost Wi-Fi-based solution suitable to provide an approximate position of the flying UAV within the environment. Despite the well-know limitations of Wi-Fi for localization purposes, the results achieved by this thesis combined with the low cost nature of the used equipment, confirmed the possibility to approximately localize in real-time the position of a flying drone, thus allowing the pilot to be ware of the UAV's position within a few meters positioning error, as a backup solution to more accurate UWB- and GNSS-based localization systems.

Moreover, in order to further push the GNSS-denied localization capabilities limits for UAVs, Chapter 6 presented a novel developed hybrid localization framework allowing to combine Wi-Fi fingerprinting together with LiDAR SLAM-based technology. The proposed framework managed to efficiently solve both the limitations of Wi-Fi and LiDAR applied to localization, combining the strengths of both solutions to compensate their relative weakness, managing to achieve relevant results in harsh environment, where a redundant and backup solution is needed to perform critical tasks.

Finally, behind the achieved results, this doctoral thesis also provided several insights regarding possible further research directions for each topic, with the aim additionally enrich and optimize the developed systems, thus allowing to exploit them in real commercial products as well as future prototypes and protocols.

List of Publications

International Journals

- Luca Davoli, Emanuele Pagliari, Gianluigi Ferrari, "Hybrid LoRa-IEEE 802.11s Opportunistic Mesh Networking for Flexible UAV Swarming", in *MDPI Drones, pp. 5–26*, April 2021.
  DOI:*10.3390/drones5020026*.

- Emanuele Pagliari, Luca Davoli, Gianluigi Ferrari, "Harnessing Communication Heterogeneity: Architectural Design, Analytical Modeling, and Performance Evaluation of an IoT Multi-Interface Gateway", in *IEEE Internet of Things Journal, pp. 1–1*, September 2023.
  DOI:*10.1109/JIOT.2023.3317672*.

- Emanuele Pagliari, Luca Davoli, Giordano Cicioni, Valentina Palazzi, Gianluigi Ferrari, "On UAV Terrestrial Connectivity Enhancement through Smart Selective Antennas", in *IOP Journal of Physics*.
  Accepted

- Emanuele Pagliari, Luca Davoli, Giordano Cicioni, Valentina Palazzi, Federico Alimenti, Luca Rosselli, Gianluigi Ferrari, "Smart Selective Antennas System (SSAS): Improving 4G LTE Connectivity for UAVs Using Directive Selective Antennas", in *IEEE Access*.
  DOI:*10.1109/ACCESS.2023.3347335*.

- Emanuele Pagliari, Luca Davoli, Gianluigi Ferrari, "Wi-Fi-based Real-Time UAV Localization: RSSI or FTM?", in *IEEE Transactions on Aerospace and Electronic Systems*.
  Submitted

International Conferences

- Emanuele Pagliari, Luca Davoli, Antonio Cilfone, Gianluigi Ferrari, "A Modular Multi-interface Gateway for Heterogeneous IoT Networking", in *2020 International Symposium on Advanced Electrical and Communication Technologies (ISAECT), pp. 1–6*, November 2020.
  DOI:*10.1109/ISAECT50560.2020.9523689*.

- Emanuele Pagliari, Luca Davoli, Giordano Cicioni, Valentina Palazzi, Gianluigi Ferrari, "On UAV Terrestrial Connectivity Enhancement through Smart Selective Antennas", in *Proceedings of the 13th EASN International Conference, pp. 1–6*, Salerno, Italy, September 2023.

- Nikolaos Stathoulopoulos, Emanuele Pagliari, Luca Davoli, George Nikolakopoulos, "Redundant and Loosely Coupled LiDAR-Wi-Fi Integration for Robust Global Localization in Autonomous Mobile Robotics", in *21st International Conference on Advanced Robotics (ICAR 2023), pp. 1–6*, Abu Dhabi, UAE, December 2023.
  Accepted

Bibliography

[1] Institute of Electrical and Electronics Engineers (IEEE). 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. *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*, pages 1–3534, Dec 2016. doi:10.1109/IEEESTD.2016.7786995.

[2] Directorate-General for Mobility and Transport. Drone Strategy: Creating a large-scale European drone market. *Transport News*, November 2022. https://t.ly/T1lck. Accessed on July 29, 2023.

[3] Grand View Research, Inc. *Commercial Drone Market Size, Share & Trends Analysis Report*. Grand View Research, Inc., 2023. https://t.ly/PyEjz. Accessed on July 29, 2023.

[4] Parrot. ANAFI Ai: The 4G robotic UAV, 2021. https://www.parrot.com/en/drones/anafi-ai. Accessed on July 20, 2023.

[5] Paul Guckian. LTE Unmanned Aircraft Systems Trial Report. Trial report, Qualcomm, May 2017. http://bit.ly/3Ftftmy. Accessed on May 12, 2023.

[6] Xingqin Lin, Richard Wiren, Sebastian Euler, Arvi Sadam, Helka-Liina Määttänen, Siva Muruganathan, Shiwei Gao, Y.-P. Eric Wang, Juhani Kauppi, Zhen-

hua Zou, and Vijaya Yajnanarayana. Mobile Network-Connected Drones: Field Trials, Simulations, and Design Insights. *IEEE Vehicular Technology Magazine*, 14(3):115–125, 2019. doi:10.1109/MVT.2019.2917363.

[7] ArduPilot. Pixhawk Cube Orange. https://t.ly/HjPZZ. Accessed on May 12, 2023.

[8] Open Robotics. Robot Operating System (ROS), 2007. http://www.ros.org/. Accessed on September 15, 2023.

[9] ExpressLRS Team. ExpressLRS - Open Source Long Range RC Control Link, 2021. https://expresslrs.org/. Accessed on September 15, 2023.

[10] Betaflight Community. Betaflight - Open Source Flight Controller Firmware, 2015. https://betaflight.com/. Accessed on September 15, 2023.

[11] Pixhawk Community. Pixhawk Autopilot - The World's Leading Open-Source Autopilot, 2011. https://pixhawk.org/. Accessed on September 15, 2023.

[12] J. Ding et al. IoT Connectivity Technologies and Applications: A Survey. *IEEE Access*, 8:67646–67673, 2020. doi:10.1109/ACCESS.2020.2985932.

[13] M. R. Palattella et al. Internet of Things in the 5G Era: Enablers, Architecture, and Business Models. *IEEE Journal on Selected Areas in Communications*, 34(3):510–527, 2016. doi:10.1109/JSAC.2016.2525418.

[14] K. Mekki et al. Overview of Cellular LPWAN Technologies for IoT Deployment: Sigfox, LoRaWAN, and NB-IoT. In *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 197–202, Athens, Greece, 2018. doi:10.1109/PERCOMW.2018.8480255.

[15] C. Chen, M. Lin, and C. Liu. Edge Computing Gateway of the Industrial Internet of Things Using Multiple Collaborative Microcontrollers. *IEEE Network*, 32(1):24–32, 2018. doi:10.1109/MNET.2018.1700146.

[16] Amir Ansari et al. Multi-interface wireless adapter and network bridge, 09 2007.

[17] B. da Silva Campos et al. Design and Construction of Wireless Sensor Network Gateway with IPv4/IPv6 Support. In *2011 IEEE International Conference on Communications (ICC)*, pages 1–5, Kyoto, Japan, 2011. doi:10.1109/icc.2011.5962848.

[18] Rajiv Ranjan et al. City Data Fusion: Sensor Data Fusion in the Internet of Things. *Int. J. Distrib. Syst. Technol.*, 7(1):15–36, 1 2016. doi:10.4018/IJDST.2016010102.

[19] B. Oniga, A. Munteanu, and V. Dadarlat. Open-source multi-protocol gateway for Internet of Things. In *2018 17th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, pages 1–6, Cluj-Napoca, Romania, 2018. doi:10.1109/ROEDUNET.2018.8514136.

[20] MAVLink. MAVLink Developer Guide. https://mavlink.io/. Accessed on May 12, 2023.

[21] European Union Aviation Safety Agency (EASA). https://www.easa.europa.eu/. Accessed on 1 April 2021.

[22] Fatemeh Jazayeri, Ali Shahidinejad, and Mostafa Ghobaei-Arani. A Latency-aware and Energy-efficient Computation Offloading in Mobile Fog Computing: A Hidden Markov Model-based Approach. *The Journal of Supercomputing*, 77(5):4887–4916, Oct 2021. doi:10.1007/s11227-020-03476-8.

[23] Hongyan Cui et al. Cloud Service Reliability Modelling and Optimal Task Scheduling. *IET Communications*, 11(2):161–167, Jan 2017. doi:10.1049/iet-com.2016.0417.

[24] Wenchen Zhou et al. Markov Approximation for Task Offloading and Computation Scaling in Mobile Edge Computing. *Mobile Information Systems*, 2019:1–12, Jan 2019. doi:10.1155/2019/8172698.

[25] Seyedsalar Sefati and Nima Jafari Navimipour. A QoS-Aware Service Composition Mechanism in the Internet of Things Using a Hidden-Markov-Model-Based Optimization Algorithm. *IEEE Internet of Things Journal*, 8(20):15620–15627, 2021. doi:10.1109/JIOT.2021.3074499.

[26] Bin Liu et al. Congestion-Optimal WiFi Offloading with User Mobility Management in Smart Communications. *Wireless Communications and Mobile Computing*, 2018:1–15, Aug 2018. doi:10.1155/2018/9297536.

[27] Areeg Samir and Claus Pahl. DLA: Detecting and Localizing Anomalies in Containerized Microservice Architectures Using Markov Models. In *2019 7th International Conference on Future Internet of Things and Cloud (FiCloud)*, pages 205–213, Istanbul, Turkey, 2019. doi:10.1109/FiCloud.2019.00036.

[28] Arun Kumar Sangaiah et al. Enforcing Position-Based Confidentiality With Machine Learning Paradigm Through Mobile Edge Computing in Real-Time Industrial Informatics. *IEEE Transactions on Industrial Informatics*, 15(7):4189–4196, 2019. doi:10.1109/TII.2019.2898174.

[29] What Is DJI OcuSync And How Does It Work?, 2019. https://tinyurl.com/5dva9ew3. Accessed on 29 December 2020.

[30] Specification for Spektrum Remote Receiver Interfacing, 2020. https://tinyurl.com/pddw634f. Accessed on 28 December 2020.

[31] FRSky Advanced Communication Control Elevated Spread Spectrum (ACCSS), 2020. https://bit.ly/3Sc5Q3g. Accessed on 28 December 2020.

[32] Futaba FASSTest, 2020. https://futabausa.com/protocols/. Accessed on 28 December 2020.

[33] Andri Radhmadhani, Isswandhana Radhika Richard, Andrea Giovani, and Riza Alaudin Syah. LoRaWAN as Secondary Telemetry Communication System for Drone Delivery. *IEEE International Conference on Internet of Things and Intelligence System (IOTAIS)*, Jan 2019. doi:10.1109/IOTAIS.2018.8600892.

[34] J.-M. Martinez-Caro and M.-D. Cano. IoT System Integrating Unmanned Aerial Vehicles and LoRa Technology: A Performance Evaluation Study. *Wireless Communications and Mobile Computing*, Nov 2019. doi:10.1155/2019/4307925.

[35] Marco Stellin, Sérgio Sabino, and António Grilo. LoRaWAN Networking in Mobile Scenarios Using a WiFi Mesh of UAV Gateways. *Electronics*, 9(4):573, Apr 2020. doi:10.3390/en11030573.

[36] Vishal Sharma, Ilsum You, Giovanni Pau, Mario Colotta, Jae Deok Lim, and Jeong Nyeo Kim. LoRaWAN-Based Energy-Efficient Surveillance by Drones for Intelligent Transportation Systems. *Energies*, 11:630, Mar 2018. doi:10.3390/electronics9040630.

[37] Omar A. Saraereh, Amer Alsaraira, Imran Khan, and Peerapong Uthansakul. Performance Evaluation of UAV-Enabled LoRa Networks for Disaster Management Applications. *Sensors*, Apr 2020. doi:10.3390/s20082396.

[38] K. G. Panda, S. Das, D. Sen, and W. Arif. Design and Deployment of UAV-Aided Post-Disaster Emergency Network. *IEEE Access*, 7:102985–102999, 2019. doi:10.1109/ACCESS.2019.2931539.

[39] S. Morgenthaler, T. Braun, Z. Zhao, T. Staub, and M. Anwander. UAVNet: A Mobile Wireless Mesh Network Using Unmanned Aerial Vehicles. In *2012 IEEE Globecom Workshops*, pages 1603–1608, Dec 2012. doi:10.1109/GLOCOMW.2012.6477825.

[40] Oscar Bautista, Kemal Akkaya, and A. Selcuk Uluagac. Customized Novel Routing Metrics for Wireless Mesh-based Swarm-of-Drones Applications.

Internet of Things, 11:100265, 2020. doi:10.1016/j.iot.2020.100265.

[41] O. Esrafilian, R. Gangula, and D. Gesbert. Autonomous UAV-aided Mesh Wireless Networks. In *IEEE INFOCOM 2020 – IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 634—640, July 2020. doi:10.1109/INFOCOMWKSHPS50562.2020.9162753.

[42] Eric A. Cai, Davis K. Furukawa, Dylan C. Leighton, Gustavo A. Velazquez, Haowei Zhang, Davide Callegaro, and Marco Levorato. Dynamic Mesh Network for Telemetry Propagation and Communications in Coordinated Drone Swarms, 2019. https://bit.ly/40511KZ. Accessed on 18 December 2020.

[43] Axel Neumann, Corinna Aichele, Marek Lindner, and Simon Wunderlich. Better Approach To Mobile Ad-hoc Networking (B.A.T.M.A.N.). Internet-Draft draft-wunderlich-openmesh-manet-routing, Internet Engineering Task Force (IETF), Apr 2008. https://tools.ietf.org/html/draft-wunderlich-openmesh-manet-routing. Accessed on 9 January 2021.

[44] O. Shrit, S. Martin, K. Alagha, and G. Pujolle. A New Approach to Realize Drone Swarm Using Ad-Hoc Network. In *2017 16th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, pages 1–5, June 2017. doi:10.1109/MedHocNet.2017.8001645.

[45] B. Barritt, T. Kichkaylo, K. Mandke, A. Zalcman, and V. Lin. Operating a UAV Mesh & Internet Backhaul Network using Temporospatial SDN. In *2017 IEEE Aerospace Conference*, pages 1–7, March 2017. doi:10.1109/AERO.2017.7943701.

[46] Muhammad Asyraf Zulkifley, Mohsen Behjati, Rosdiadee Nordin, and Mohd Saufee Zakaria. Mobile Network Performance and Technical Feasi-

bility of LTE-Powered Unmanned Aerial Vehicle. *Sensors*, 21(8):2848, 2021. doi:10.3390/s21082848.

[47] Azade Fotouhi, Haoran Qiang, Ming Ding, Mahbub Hassan, Lorenzo Galati Giordano, Adrian Garcia-Rodriguez, and Jinhong Yuan. Survey on UAV Cellular Communications: Practical Aspects, Standardization Advancements, Regulation, and Security Challenges. *IEEE Communications Surveys & Tutorials*, 21(4):3417–3442, 2019. doi:10.1109/COMST.2019.2906228.

[48] Third Generation Partnership Project (3GPP). Enhanced LTE support for aerial vehicles. TR 36.777, Third Generation Partnership Project (3GPP), Jan 2018. https://rb.gy/hr2nl. Accessed on July 29, 2023.

[49] Siva D. Muruganathan et al. An Overview of 3GPP Release-15 Study on Enhanced LTE Support for Connected Drones, 2019. doi:10.48550/arXiv.1805.00826.

[50] William D. Ivancic, Robert J Kerczewski, Robert W. Murawski, Konstantin Matheou, and Alan N Downey. Flying Drones Beyond Visual Line of Sight Using 4g LTE: Issues and Concerns. In *2019 Integrated Communications, Navigation and Surveillance Conference (ICNS)*, pages 1–13, 2019. doi:10.1109/ICNSURV.2019.8735246.

[51] Huan Cong Nguyen, Rafhael Amorim, Jeroen Wigard, IstváN Z. KováCs, Troels B. Sørensen, and Preben E. Mogensen. How to Ensure Reliable Connectivity for Aerial Vehicles Over Cellular Networks. *IEEE Access*, 6:12304–12317, 2018. doi:10.1109/ACCESS.2018.2808998.

[52] Jiangbin Lyu and Rui Zhang. Network-Connected UAV: 3-D System Modeling and Coverage Performance Analysis. *IEEE Internet of Things Journal*, 6(4):7048–7060, 2019. doi:10.1109/JIOT.2019.2913887.

[53] Arne Colpaert, Evgeny Vinogradov, and Sofie Pollin. Aerial Coverage Analysis of Cellular Systems at LTE and mmWave Frequencies Using 3D City Models. *Sensors*, 18(12):4311, 2018. doi:10.3390/s18124311.

[54] Shuowen Zhang and Rui Zhang. Radio Map-Based 3D Path Planning for Cellular-Connected UAV. *IEEE Transactions on Wireless Communications*, 20(3):1975–1989, 2021. doi:10.1109/TWC.2020.3037916.

[55] Hakmin Lee, Ye-Bon Kim, and Han Lim Lee. Reconfigurable Antenna for UAV-Assisted Wide Coverage Air-to-Ground Communications. *IEEE Access*, 10:88034–88042, 2022. doi:10.1109/ACCESS.2022.3199700.

[56] Sebastian Sadowski and Petros Spachos. RSSI-Based Indoor Localization With the Internet of Things. *IEEE Access*, 6:30149–30161, 2018. doi:10.1109/ACCESS.2018.2843325.

[57] Y. Chouchang and H. r. Shao. WiFi-based indoor positioning. *IEEE Communications Magazine*, 53:150 – 157, March 2015.

[58] Fabrizio Carpi, Marco Martalò, Luca Davoli, Antonio Cilfone, Yingjie Yu, Yi Wang, and Gianluigi Ferrari. Experimental analysis of RSSI-based localization algorithms with NLOS pre-mitigation for IoT applications. *Computer Networks*, 225:109663, 2023. doi:10.1016/j.comnet.2023.109663.

[59] R. M. M. R. Rathnayake, Madduma Wellalage Pasan Maduranga, Valmik Tilwari, and Maheshi B. Dissanayake. RSSI and Machine Learning-Based Indoor Localization Systems for Smart Cities. *Eng*, 4(2):1468–1494, 2023. doi:10.3390/eng4020085.

[60] Navneet Singh, Sangho Choe, and Rajiv Punmiya. Machine Learning Based Indoor Localization Using Wi-Fi RSSI Fingerprints: An Overview. *IEEE Access*, 9:127150–127174, 2021. doi:10.1109/ACCESS.2021.3111083.

[61] Biljana Risteska Stojkoska, Jordan Palikrushev, Kire Trivodaliev, and Slobodan Kalajdziski. Indoor localization of unmanned aerial vehicles based on RSSI. In *IEEE EUROCON 2017 -17th International Conference on Smart Technologies*, pages 120–125, 2017. doi:10.1109/EUROCON.2017.8011089.

[62] Atthaphon Booranawong, Phoowanarth Thammachote, and Yuttana Sasiwat. Real-time tracking of a moving target in an indoor corridor of the hospital building using RSSI signals received from two reference nodes. *Medical & Biological Engineering & Computing*, 60:439–458, 2022. doi:10.1007/s11517-021-02489-6.

[63] Hongyu Jin and Panos Papadimitratos. Off-the-shelf Wi-Fi Indoor Smartphone Localization. In *2022 17th Wireless On-Demand Network Systems and Services Conference (WONS)*, pages 1–4, 2022. doi:10.23919/WONS54113.2022.9764448.

[64] Rong Peng and Mihail L. Sichitiu. Angle of Arrival Localization for Wireless Sensor Networks. In *2006 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*, volume 1, pages 374–382, 2006. doi:10.1109/SAHCN.2006.288442.

[65] Paulson Eberechukwu Numan, Hyunwoo Park, Christos Laoudias, Seppo Horsmanheimo, and Sunwoo Kim. Smartphone-Based Indoor Localization via Network Learning With Fusion of FTM/RSSI Measurements. *IEEE Networking Letters*, 5(1):21–25, 2023. doi:10.1109/LNET.2022.3226462.

[66] Kevin Jiokeng, Gentian Jakllari, Alain Tchana, and André-Luc Beylot. When FTM Discovered MUSIC: Accurate WiFi-based Ranging in the Presence of Multipath. In *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, pages 1857–1866, 2020. doi:10.1109/INFOCOM41043.2020.9155464.

[67] Yuichiro Sugiyama, Kentaro Kobayashi, and Wataru Chujo. A study on indoor drone positioning using Wi-Fi RTT ranging. In *2023 VTS Asia Pacific Wireless Communications Symposium (APWCS)*, pages 1–5, 2023. doi:10.1109/APWCS60142.2023.10234042.

[68] Valentín Barral Vales, Omar Campos Fernández, Tomás Domínguez-Bolaño, Carlos J. Escudero, and José A. García-Naya. Fine Time Measurement for

the Internet of Things: A Practical Approach Using ESP32. *IEEE Internet of Things Journal*, 9(19):18305–18318, 2022. doi:10.1109/JIOT.2022.3158701.

[69] DecaWave DW1000. http://www.decawave.com/products/dwm1000-module. Accessed: October 8, 2023.

[70] Carlos S. Álvarez-Merino, Emil J. Khatib, Hao Qiang Luo-Chen, Joel Llanes Michel, Sebastián Casalderrey-Díaz, Jesus Alonso, and Raquel Barco. WiFi FTM and UWB Characterization for Localization in Construction Sites. *Sensors*, 22(14), 2022. doi:10.3390/s22145373.

[71] Abdulrahman Alarifi, AbdulMalik Al-Salman, Mansour Alsaleh, Ahmad Alnafessah, Suheer Alhadhrami, Mai A. Al-Ammar, and Hend Al-Khalifa. Ultra Wideband Indoor Positioning Technologies: Analysis and Recent Advances. *Sensors*, 16:1–36, 05 2016.

[72] B. Barua, N. Kandil, and N. Hakem. On performance study of TWR UWB ranging in underground mine. In *2018 Sixth International Conference on Digital Information, Networking, and Wireless Communications (DINWC)*, pages 28–31, April 2018. doi:10.1109/DINWC.2018.8356990.

[73] Jorge Peña Queralta, Carmen Martínez Almansa, Fabrizio Schiano, Dario Floreano, and Tomi Westerlund. UWB-based System for UAV Localization in GNSS-Denied Environments: Characterization and Dataset. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4521–4528, 2020. doi:10.1109/IROS45743.2020.9341042.

[74] Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Daniela Rus. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5135–5142, 2020. doi:10.1109/IROS45743.2020.9341176.

[75] César Cadena, Dorian Gálvez-López, Fabio Ramos, Juan D. Tardós, and José Neira. Robust place recognition with stereo cameras. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5182–5189, 2010. doi:10.1109/IROS.2010.5650234.

[76] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. volume 3951, pages 404–417, 07 2006. doi:10.1007/11744023_32.

[77] Mikaela Angelina Uy and Gim Lee. PointNetVLAD: Deep Point Cloud Based Retrieval for Large-Scale Place Recognition. 06 2018. doi:10.1109/CVPR.2018.00470.

[78] Lukas Schaupp, Mathias Burki, Renaud Dube, Roland Siegwart, and Cesar Cadena. OREOS: Oriented Recognition of 3D Point Clouds in Outdoor Scenarios. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3255–3261, 2019. doi:10.1109/IROS40897.2019.8968094.

[79] Xieyuanli Chen, Thomas Läbe, Andres Milioto, Timo Röhling, Olga Vysotska, Alexandre Haag, Jens Behley, and Cyrill Stachniss. OverlapNet: Loop Closing for LiDAR-based SLAM. *CoRR*, abs/2105.11344, 2021. arXiv:2105.11344.

[80] Wondimu K. Zegeye, Seifemichael B. Amsalu, Yacob Astatke, and Farzad Moazzami. WiFi RSS fingerprinting indoor localization for mobile devices. In *2016 IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pages 1–6, 2016. doi:10.1109/UEMCON.2016.7777834.

[81] Priyath Fonseka and Kumbesan Sandrasegaran. Indoor localization for IoT applications using fingerprinting. In *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, pages 736–741, 2018. doi:10.1109/WF-IoT.2018.8355105.

[82] Shuang Shang and Lixing Wang. Overview of WiFi fingerprinting-based indoor positioning. *IET Communications*, 16(7), 2022. doi:10.1049/cmu2.12386.

[83] Navneet Singh, Sangho Choe, and Rajiv Punmiya. Machine Learning Based Indoor Localization Using Wi-Fi RSSI Fingerprints: An Overview. *IEEE Access*, 9:127150–127174, 2021. doi:10.1109/ACCESS.2021.3111083.

[84] Wen Liu, Changyan Qin, Zhongliang Deng, and Haoyue Jiang. LRF-WiVi: A WiFi and Visual Indoor Localization Method Based on Low-Rank Fusion. *Sensors*, 22(22), 2022. doi:10.3390/s22228821.

[85] Z.S. Hashemifar, C. Adhivarahan, A. Balakrishnan, and et al. Augmenting visual SLAM with Wi-Fi sensing for indoor applications. *Autonomous Robots*, 43(12):2245–2260, 2019. doi:10.1007/s10514-019-09874-z.

[86] Khairuldanial Ismail, Ran Liu, Zhenghong Qin, Achala Athukorala, Billy Pik Lik Lau, Muhammad Shalihan, Chau Yuen, and U-Xuan Tan. Efficient WiFi LiDAR SLAM for Autonomous Robots in Large Environments. In *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, pages 1132–1137, 2022. doi:10.1109/CASE49997.2022.9926530.

[87] L. Davoli et al. From Micro to Macro IoT: Challenges and Solutions in the Integration of IEEE 802.15.4/802.11 and Sub-GHz Technologies. *IEEE Internet of Things Journal*, 5(2):784–793, April 2018. doi:10.1109/JIOT.2017.2747900.

[88] LoRa GPS HAT for Raspberry Pi. https://www.dragino.com/products/lora/item/106-lora-gps-hat.html. Accessed on June 20, 2023.

[89] LoRa and LoRaWAN Specifications. https://tinyurl.com/y2kcawv8. Accessed on June 20, 2023.

[90] Espressif. ESP32: A feature-rich MCU with integrated Wi-Fi and Bluetooth connectivity for a wide-range of applications, 2017. https://tinyurl.com/bdfdu25k. Accessed on 7 January 2021.

[91] DHT11 Humidity & Temperature Sensor. https://rb.gy/mlfha. Accessed on June 20, 2023.

[92] Gaia Codeluppi, Luca Davoli, and Gianluigi Ferrari. Forecasting Air Temperature on Edge Devices with Embedded AI. *Sensors*, 21(12), 2021. doi:10.3390/s21123973.

[93] LoRaWAN 1.0.3 Regional Parameters. https://rb.gy/dadwr. Accessed on June 20, 2023.

[94] Kingman J. F. C. The single server queue in heavy traffic. *Mathematical Proceedings of the Cambridge Philosophical Society*, 57(4):902–904, 1961. doi:10.1017/S0305004100036094.

[95] SX1272/3/6/7/8 LoRa Modem Design Guide, 2013. https://tinyurl.com/243h5zvp. Accessed on June 20, 2023.

[96] Sheldon M. Ross. *Introduction to Probability Models*. Academic Press, 2014.

[97] Diana Mindrila and Phoebe Balentyne. Confidence Intervals. https://www.westga.edu/academics/research/vrc/assets/docs/confidence_intervals_notes.pdf. Accessed on June 20, 2023.

[98] The Things Network (TTN). https://www.thethingsnetwork.org/. Accessed on 29 December 2020.

[99] The daemon-systems.org Project. Dhcpcd configuration file. https://www.daemon-systems.org/man/dhcpcd.conf.5.html. Accessed on September 15, 2023.

[100] The Linux Documentation Project. Linux namespaces. https://man7.org/linux/man-pages/man7/namespaces.7.html. Accessed on September 15, 2023.

[101] TTN Mapper: a LoRaWAN Coverage Mapping Tool, 2020. https://ttnmapper.org/. Accessed on 5 January 2021.

[102] Juha Petajajarvi, Janhunen Janne Mikhaylov Konstantin, Jari Iinatti, and Marko Pettissalo. Performance of a low-power wide-area network based on lora technology: Doppler robustness, scalability, and coverage. *International Journal of Distributed Sensor Networks*, Mar 2017. doi:10.1177/1550147717699412.

[103] Eclipse Foundation. Zenoh: Zero Overhead Pub/Sub, Store/Query and Compute. https://zenoh.io/. Accessed on June 20, 2023.

[104] Luca Davoli, Massimo Moreni, and Gianluigi Ferrari. *A Sink-oriented Routing Protocol for Blue Light Link-based Mesh Network*, pages 21–31. Telecommunications. Institution of Engineering and Technology, 2022. doi:10.1049/PBTE101E_ch2.

[105] Olivier Bernard Andre Seller and Nicolas Sornin. Low power long range transmitter, 2013. EP2763321A1. URL: https://patents.google.com/patent/EP2763321A1/en.

[106] Institute of Electrical and Electronics Engineers (IEEE). 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 10: Mesh Networking. *IEEE Std 802.11s-2011*, pages 1–372, Sep. 2011. doi:10.1109/IEEESTD.2011.6018236.

[107] Federal Aviation Administration (FAA). https://www.faa.gov/. Accessed on 1 April 2021.

[108] Gilles Callebaut, Guus Leenders, Chesney Buyle, Stijn Crul, and Liesbet Van der Perre. LoRa Physical Layer Evaluation for Point-to-Point Links and Coverage Measurements in Diverse Environments. *IEEE European Conference on Networks and Communications (EuCNC)*, 2019.

[109] Juha Petäjäjärvi, Konstantin Mikhaylov, Antti Roivainen, Tuomo Hänninen, and Marko Pettissalo. On the Coverage of LPWANs: Range Evaluation and Channel Attenuation Model for LoRa Technology. *14th International Conference on ITS Telecommunications (ITST)*, Dec 2015. doi:10.1109/ITST.2015.7377400.

[110] G. R. Hiertz, D. Denteneer, S. Max, R. Taori, J. Cardona, L. Berlemann, and B. Walke. IEEE 802.11s: The WLAN Mesh Standard. *IEEE Wireless Communications*, 17(1):104–111, Feb 2010. doi:10.1109/MWC.2010.5416357.

[111] Antonio Cilfone, Luca Davoli, Laura Belli, and Gianluigi Ferrari. Wireless Mesh Networking: An IoT-Oriented Perspective Survey on Relevant Technologies. *Future Internet*, 11(4):99, Apr 2019. doi:10.3390/fi11040099.

[112] S. M. S. Bari, F. Anwar, and M. H. Masud. Performance study of hybrid Wireless Mesh Protocol (HWMP) for IEEE 802.11s WLAN mesh networks. In *2012 International Conference on Computer and Communication Engineering (ICCCE)*, pages 712–716, July 2012. doi:10.1109/ICCCE.2012.6271309.

[113] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561, Internet Engineering Task Force (IETF), July 2003. https://tools.ietf.org/html/rfc3561. Accessed on 9 January 2021.

[114] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol (OLSR). RFC 3626, Internet Engineering Task Force (IETF), Oct 2003. https://tools.ietf.org/html/rfc3626. Accessed on 9 January 2021.

[115] J. Chroboczek. The Babel Routing Protocol. RFC 6126, Internet Engineering Task Force (IETF), April 2011. https://tools.ietf.org/html/rfc6126. Accessed on 9 January 2021.

[116] J. Chroboczek. Extension Mechanism for the Babel Routing Protocol. RFC 7557, Internet Engineering Task Force (IETF), May 2015. https://tools.ietf.org/html/rfc7557. Accessed on 9 January 2021.

[117] H. T. Friis. A Note on a Simple Transmission Formula. *Proceedings of the IRE*, 34(5):254–256, May 1946. doi:10.1109/JRPROC.1946.234568.

[118] Syed Kamrul Islam and Mohammad Rafiqul Haider. *Sensors and Low Power Signal Processing*. Springer US, New York, 2010. doi:10.1007/978-0-387-79392-4.

[119] A technical overview of LoRa and LoRaWAN. https://www.loraalliance.org/sites/default/files/2018-04/what-is-lorawan.pdf. Accessed on 29 December 2020.

[120] Decoding Lora, 2020. https://revspace.nl/DecodingLora. Accessed on 19 December 2020.

[121] Understanding I/Q Signals and Quadrature Modulation, 2020. https://bit.ly/3FxcOYZ. Accessed on 19 December 2020.

[122] LoRa Alliance. RP002-1.0.0 LoRaWAN Regional Parameters. https://tinyurl.com/mtpr62r6. Accessed on 28 December 2020.

[123] The Things Conference. How Spreading Factor affects LoRaWAN device battery life, Nov. 2019. https://rb.gy/dplfg. Accessed on 27 December 2020.

[124] Augustin Aloys, Jiazi Yi, Thomas Clausen, and William Mark Townsley. A Study of LoRa: Long Range & Low Power Networks for the Internet of Things. *Sensors*, 16:1466, Apr 2016. doi:10.3390/s16091466.

[125] LoRaWAN 1.0.2 Regional Parameters, Feb. 2017. https://bit.ly/3FqK8Ro. Accessed on 30 December 2020.

[126] Tommaso Polonelli, Davide Brunelli, Achille Marzocchi, and Luca Benini. Slotted ALOHA on LoRaWAN-Design, Analysis, and Deployment. *Sensors*, 19(4):838, Feb 2019. doi:10.3390/s19040838.

[127] An In-depth look at LoRaWAN Class A Devices. https://bit.ly/3FqKGqq. Accessed on 14 December 2020.

[128] LoRaWAN 1.1 Regional Parameters, Oct. 2017. https://lora-alliance.org/wp-content/uploads/2020/11/lorawan-regional-parameters-v1.1ra.pdf. Accessed on 9 January 2021.

[129] An In-depth Look at LoRaWAN Class B Devices. https://bit.ly/473WWJ2. Accessed on 14 December 2020.

[130] An In-depth Look at LoRaWAN Class C Devices. https://rb.gy/fg1b6. Accessed on 14 December 2020.

[131] Semtech. LoRaWAN Airtime Calculator. https://tinyurl.com/loracalc. Accessed on 29 December 2020.

[132] The Things Network. Limitations: data rate, packet size, 30 seconds uplink and 10 messages downlink per day Fair Access Policy, 2016. https://bit.ly/36AVwbd. Accessed on 29 December 2020.

[133] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne. Understanding the Limits of LoRaWAN. *IEEE Communications Magazine*, 55(9):34–40, Sep. 2017. doi:10.1109/MCOM.2017.1600613.

[134] European Telecommunications Standards Institute (ETSI). Short Range Devices (SRD) operating in the frequency range 25 MHz to 1,000 MHz;

Part 2: Harmonised Standard for access to radio spectrum for non specific radio equipment, Jun 2018. https://www.etsi.org/deliver/etsi_en/300200_300299/30022002/03.02.01_60/en_30022002v030201p.pdf. Accessed on 11 January 2021.

[135] Nelson C. Almeida, Rodrigo P. Rolle, Eduardo P. Godoy, Paolo Ferrari, and Emiliano Sisinni. Proposal of a Hybrid LoRa Mesh / LoRaWAN Network. *IEEE International Workshop on Metrology for Industry 4.0 & IoT*, Jul 2020. doi:10.1109/MetroInd4.0IoT48571.2020.9138206.

[136] Huang-Chen Lee and Kai-Hsiang Ke. Monitoring of Large-Area IoT Sensors Using a LoRa Wireless Mesh Network System: Design and Evaluation. *IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT*, Sep 2018. doi:10.1109/TIM.2018.2814082.

[137] Heon Huh and Jeong Yeol Kim. LoRa-based Mesh Network for IoT Applications. *IEEE 5th World Forum on Internet of Things (WF-IoT)*, Jul 2019. doi:10.1109/WF-IoT.2019.8767242.

[138] W. Steven Conner, Jan Kruys, Kyeongsoo (Joseph) Kim, and Juan Carlos Zuniga. IEEE 802.11s Tutorial Overview of the Amendment for Wireless Local Area Mesh Networking, Nov 2006. https://www.ieee802.org/802_tutorials/06-November/802.11s_Tutorial_r5.pdf. Accessed on 27 December 2020.

[139] LoRa Line Of Sight Range Test, 2018. https://www.mesh-net.co.uk/lora-line-sight-range-test/. Accessed on 5 January 2021.

[140] LoRa World Record Broken: 832km/517mi using 25mW, 2018. https://bit.ly/3S9ob0x. Accessed on 28 December 2020.

[141] A. Guillen-Perez, R. Sanchez-Iborra, M. Cano, J. C. Sanchez-Aarnoutse, and J. Garcia-Haro. WiFi Networks on Drones. In *2016 ITU Kaleidoscope: ICTs for a Sustainable World (ITU WT)*, pages 1–8, Nov 2016. doi:10.1109/ITU-WT.2016.7805730.

[142] Selvakumar Manickam. A Drone-based IoT Approach to Agriculture Automation and Increase Farm Yield. *SSRN Electronic Journal*, 2020. doi:10.2139/ssrn.3713675.

[143] Gaia Codeluppi, Antonio Cilfone, Luca Davoli, and Gianluigi Ferrari. VegIoT Garden: a modular IoT Management Platform for Urban Vegetable Gardens. In *2019 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor)*, pages 121–126, Oct 2019. doi:10.1109/MetroAgriFor.2019.8909228.

[144] Afshin Ameri E., Baran Cürüklü, Branko Miloradović, and Mikael Ektröm. Planning and Supervising Autonomous Underwater Vehicles through the Mission Management Tool. In *2020 Global OCEANS*, pages 1–1, Oct 2020.

[145] Zhaoyu Zhai, José Fernán Martínez, Victoria Beltran, and Néstor Lucas Martínez. Decision Support Systems for Agriculture 4.0: Survey and Challenges. *Computers and Electronics in Agriculture*, 170:105256, 2020. doi:10.1016/j.compag.2020.105256.

[146] Laura Belli, Simone Cirani, Luca Davoli, Lorenzo Melegari, Màrius Mónton, and Marco Picone. An Open-Source Cloud Architecture for Big Stream IoT Applications. In Ivana Podnar Žarko, Krešimir Pripužić, and Martin Serrano, editors, *Interoperability and Open-Source Solutions for the Internet of Things: International Workshop, FP7 OpenIoT Project, Held in Conjunction with SoftCOM 2014, Split, Croatia, September 18, 2014, Invited Papers*, pages 73–88. Springer International Publishing, Sep 2015. doi:10.1007/978-3-319-16546-2_7.

[147] Scott Xiang Fang, Siu O'Young, and Luc Rolland. Development of Small UAS Beyond-Visual-Line-of-Sight (BVLOS) Flight Operations: System Requirements and Procedures. *Drones*, 2(2), 2018. doi:10.3390/drones2020013.

[148] William D. Ivancic, Robert J. Kerczewski, Robert W. Murawski, Konstantin Matheou, and Alan N. Downey. Flying Drones Beyond Visual Line of Sight Using 4G LTE: Issues and Concerns. In *2019 Integrated Communications, Navigation and Surveillance Conference (ICNS)*. IEEE, Apr 2019. doi:10.1109/icnsurv.2019.8735278.

[149] Huan Cong Nguyen, Rafhael Amorim, Jeroen Wigard, Istvan Z. Kovacs, and Preben Mogensen. Using LTE Networks for UAV Command and Control Link: A Rural-Area Coverage Analysis. In *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*. IEEE, Sep 2017. doi:10.1109/vtcfall.2017.8287894.

[150] Chen Zhong, Ziyi Zhao, Chen Luo, M. Cenk Gursoy, Qinru Qiu, Carlos Caicedo, Franco Basti, and Adrian Solomon. A Cost-Benefit Analysis to Achieve Command and Control (C2) Link Connectivity for Beyond Visual Line of Sight (BVLOS) Operations. In *2020 Integrated Communications Navigation and Surveillance Conference (ICNS)*. IEEE, Sep 2020. doi:10.1109/icns50378.2020.9222956.

[151] M. Tortonesi, C. Stefanelli, E. Benvegnu, K. Ford, N. Suri, and M. Linderman. Multiple-UAV coordination and communications in tactical edge networks. *IEEE Communications Magazine*, 50(10):48–55, Oct 2012. doi:10.1109/MCOM.2012.6316775.

[152] Bruno José Olivieri de Souza and Markus Endler. Evaluating Flight Coordination Approaches of UAV Squads for WSN Data Collection Enhancing the Internet Range on WSN Data Collection. *Journal of Internet Services and Applications*, 11(1), Jul 2020. doi:10.1186/s13174-020-00125-4.

[153] Ubiquiti Networks, Inc. Ubiquiti UniFi 6 Pro Access Point. https://eu.store.ui.com/eu/en/products/u6-pro. Accessed on September 15, 2023.

[154] IEEE. IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC)and Physical Layer (PHY) Specifications Amendment 1: Radio Resource Measurement of Wireless LANs. *IEEE Std 802.11k-2008 (Amendment to IEEE Std 802.11-2007)*, pages 1–244, 2008. doi: 10.1109/IEEESTD.2008.4544755.

[155] Intel NUC - Next Unit of Computing. Online. https://www.intel.com/content/www/us/en/products/boards-kits/nuc.html. Accessed on September 15, 2023.

[156] European Telecommunications Standards Institute (ETSI). ETSI TS 136 214 V9.1.0 (2010-04) Technical Specification LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer - Measurements (3GPP TS 36.214 version 9.1.0 Release 9). Technical report, European Telecommunications Standards Institute (ETSI), 2010. https://www.etsi.org/deliver/etsi_ts/136200_136299/136214/09.01.00_60/ts_136214v090100p.pdf. Accessed on May 12, 2023.

[157] Sierra Wireless. EM9191 5G NR Sub-6 GHz Module. https://t.ly/TKh6u. Accessed on May 12, 2023.

[158] Techship MU201 Adapter (M.2 key B to USB3 type A, Dual SIM), 2019. https://t.ly/SE6G7. Accessed on July 29, 2023.

[159] Jonas Sedin, Sebastian Euler, and Xingqin Lin. Drones and Networks: Mobility Support. https://www.ericsson.com/en/blog/2019/1/drones-and-networks-mobility-support. Accessed on May 12, 2023, November 2018.

[160] Y. Zhao, X. Li, Y. Ji, and C. Xu. Wireless Power-Driven Positioning System: Fundamental Analysis and Resource Allocation. *IEEE Internet of Things J.*, 6(6):10421–10430, 2019.

[161] IEEE Standards Association. IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs). Standard 802.15.4-2011, IEEE, 2011.

[162] Jens Wagner, Axel Strobel, Niko Joram, Ralf Eickhoff, and Frank Ellinger. FMCW system aspects for multipath environments. In *2011 8th Workshop on Positioning, Navigation and Communication*, pages 89–93, 2011. doi:10.1109/WPNC.2011.5961021.

[163] GEPRC CineLog25 quad-copter. https://geprc.com/product/geprc-cinelog25-hd-runcam-link-wasp-cinewhoop-drone/. Accessed: October 8, 2023.

[164] European Global Navigation Satellite System Agency (GSA). Gnss user technology report. Technical Report Issue 2, European Commission, and GSA, August 2018.

[165] Omar Hashem, Khaled A. Harras, and Moustafa Youssef. Accurate indoor positioning using IEEE 802.11mc round trip time. *Pervasive and Mobile Computing*, 75:101416, 2021. doi:10.1016/j.pmcj.2021.101416.

[166] Yu Gu, Inseok Hwang, Su Liu, and Yaoguang Wei. WiFi-fingerprint based indoor localization map, 7 2017.

[167] Theodore S. Rappaport. *Wireless Communications: Principles and Practice*. Prentice-Hall, 2002.

[168] 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–. Technical report, IEEE, 2007.

[169] Wi-Fi RTT (IEEE 802.11mc), 2021. https://source.android.com/devices/tech/connect/wifi-rtt. Accessed: October 8, 2023.

[170] The ESP32-S3 is the new flagship MCU from Espressif. https://esp32s3.com. Accessed: October 8, 2023.

[171] Luc Jaulin. 5 - instantaneous localization. In Luc Jaulin, editor, *Mobile Robotics*, pages 171–196. Elsevier, 2015. doi:10.1016/B978-1-78548-048-5.50005-X.

[172] T. Kariya and H. Kurata. *Generalized Least Squares*. Wiley, Hoboken, 2004.

[173] LSMR: An Iterative Algorithm for Sparse Least-Squares Problems. *SIAM Journal on Scientific Computing*, 33(5):2950–2971, 2011. arXiv:https://doi.org/10.1137/10079687X, doi:10.1137/10079687X.

[174] R.E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45, 1960. doi:10.1115/1.3662552.

[175] G.A. Einicke and L.B. White. Robust Extended Kalman Filtering. *IEEE Trans. Signal Process.*, 47(9):2596–2599, September 1999. doi:10.1109/78.782219.

[176] Lindsay Kleeman. *Understanding and Applying Kalman Filtering*. Department of Electrical and Computer Systems Engineering, Monash University, Clayton, 2006.

[177] Vedran Kordic. *Kalman Filter*. IntechOpen, Rijeka, May 2010. doi:10.5772/233.

[178] T3-S3 V1.0 ESP32-S3 LoRa SX1280 2.4G Development Board. https://www.lilygo.cc/products/t3s3-v1-0?variant=43043715678389. Accessed: October 8, 2023.

[179] Department of Defense World Geodetic System 1984. Department of Defense World Geodetic System 1984. Technical Report 2nd edition, Defense Mapping Agency, 1991-09-01. https://apps.dtic.mil/sti/pdfs/ADA280358.pdf. Accessed: October 8, 2023.

[180] James R. Clynch. Earth Coordinates. Technical report, SAGE Publications, February 2006. https://bit.ly/45FnDmu. Accessed: October 8, 2023.

[181] Guangyang Zeng, Biqiang Mu, Jiming Chen, Zhiguo Shi, and Junfeng Wu. Global and Asymptotically Efficient Localization From Range Measurements. *IEEE Transactions on Signal Processing*, 70:5041–5057, 2022. doi:10.1109/TSP.2022.3198167.

[182] Daniele Cattaneo, Matteo Vaghi, and Abhinav Valada. LCDNet: Deep Loop Closure Detection and Point Cloud Registration for LiDAR SLAM. *IEEE Transactions on Robotics*, 38(4):2074–2093, 2022. doi:10.1109/TRO.2022.3150683.

[183] Alexander Kyuroson, Niklas Dahlquist, Nikolaos Stathoulopoulos, Vignesh Kottayam Viswanathan, Anton Koval, and George Nikolakopoulos. Multimodal Dataset from Harsh Sub-Terranean Environment with Aerosol Particles for Frontier Exploration, 2023. arXiv:2304.14520.

[184] John Immerkær. Fast Noise Variance Estimation. *Computer Vision and Image Understanding*, 64(2):300–302, 1996. doi:10.1006/cviu.1996.0060.

[185] Kenji Koide, Masashi Yokozuka, Shuji Oishi, and Atsuhiko Banno. Voxelized GICP for Fast and Accurate 3D Point Cloud Registration. EasyChair Preprint no. 2703, EasyChair, 2020.

[186] Kenny Chen, Brett T. Lopez, Ali-akbar Agha-mohammadi, and Ankur Mehta. Direct lidar odometry: Fast localization with dense point clouds. *IEEE Robotics and Automation Letters*, 7(2):2000–2007, 2022. doi:10.1109/LRA.2022.3142739.

[187] Navneet Singh, Sangho Choe, and Rajiv Punmiya. Machine Learning Based Indoor Localization Using Wi-Fi RSSI Fingerprints: An Overview. *IEEE Access*, 9:127150–127174, 2021. doi:10.1109/ACCESS.2021.3111083.

[188] Junhang Bai, Yongliang Sun, Weixiao Meng, and Cheng Li. Wi-Fi Fingerprint-Based Indoor Mobile User Localization Using Deep Learning. *Wireless Communications and Mobile Computing*, 2021(6660990), 2021. doi:10.1155/2021/6660990.