Design and experimental evaluation of unmanned aircraft systems communications

Politecnico di Bari

Department of Electrical and Information Engineering
ELETTRICAL AND INFORMATION ENGINEERING
Ph.D. Program
SSD: IINF-03/A – TELECOMMUNICATIONS

Final Dissertation

Design and Experimental Evaluation of Unmanned Aircraft Systems Communications

---

by
GRIECO Giovanni

Supervisor:

Prof. Luigi Alfredo Grieco

Coordinator of Ph.D. Program:
Prof. Mario Carpentieri

Course n°37, 01/11/2021-31/10/2024

Politecnico di Bari

LIBERATORIA PER L'ARCHIVIAZIONE DELLA TESI DI DOTTORATO

Al Magnifico Rettore
del Politecnico di Bari

Il sottoscritto GRIECO Giovanni, nato a Bari (BA) il 01/03/1996, residente a Bari (BA) in via Luca de Samuele Cagnazzi 57, 70124 Bari (BA), e-mail giovanni@grieco.dev, iscritto al 3° anno di Corso di Dottorato di Ricerca in Ingegneria Elettrica e dell'Informazione, ciclo XXXVII ed essendo stato ammesso a sostenere l'esame finale con la prevista discussione della tesi dal titolo:

Design and Experimental Evaluation of Unmanned Aircraft Systems Communications

DICHIARA

1) di essere consapevole che, ai sensi del D.P.R. n. 445 del 28.12.2000, le dichiarazioni mendaci, la falsità negli atti e l'uso di atti falsi sono puniti ai sensi del codice penale e delle Leggi speciali in materia, e che nel caso ricorressero dette ipotesi, decade fin dall'inizio e senza necessità di nessuna formalità dai benefici conseguenti al provvedimento emanato sulla base di tali dichiarazioni;
2) di essere iscritto al Corso di Dottorato di ricerca in Ingegneria Elettrica e dell'Informazione ciclo XXXVII, corso attivato ai sensi del "*Regolamento dei Corsi di Dottorato di ricerca del Politecnico di Bari"*, emanato con D.R. n.286 del 01.07.2013;
3) di essere pienamente a conoscenza delle disposizioni contenute nel predetto Regolamento in merito alla procedura di deposito, pubblicazione e autoarchiviazione della tesi di dottorato nell'Archivio Istituzionale ad accesso aperto alla letteratura scientifica;
4) di essere consapevole che attraverso l'autoarchiviazione delle tesi nell'Archivio Istituzionale ad accesso aperto alla letteratura scientifica del Politecnico di Bari (IRIS-POLIBA), l'Ateneo archivierà e renderà consultabile in rete (nel rispetto della Policy di Ateneo di cui al D.R. 642 del 13.11.2015) il testo completo della tesi di dottorato, fatta salva la possibilità di sottoscrizione di apposite licenze per le relative condizioni di utilizzo (di cui al sito http://www.creativecommons.it/Licenze), e fatte salve, altresì, le eventuali esigenze di "embargo", legate a strette considerazioni sulla tutelabilità e sfruttamento industriale/commerciale dei contenuti della tesi, da rappresentarsi mediante compilazione e sottoscrizione del modulo in calce (Richiesta di embargo);
5) che la tesi da depositare in IRIS-POLIBA, in formato digitale (PDF/A) sarà del tutto identica a quelle **consegnate**/inviate/da inviarsi ai componenti della commissione per l'esame finale e a qualsiasi altra copia depositata presso gli Uffici del Politecnico di Bari in forma cartacea o digitale, ovvero a quella da discutere in sede di esame finale, a quella da depositare, a cura dell'Ateneo, presso le Biblioteche Nazionali Centrali di Roma e Firenze e presso tutti gli Uffici competenti per legge al momento del deposito stesso, e che di conseguenza va esclusa qualsiasi responsabilità del Politecnico di Bari per quanto riguarda eventuali errori, imprecisioni o omissioni nei contenuti della tesi;
6) che il contenuto e l'organizzazione della tesi è opera originale realizzata dal sottoscritto e non compromette in alcun modo i diritti di terzi, ivi compresi quelli relativi alla sicurezza dei dati personali; che pertanto il Politecnico di Bari ed i suoi funzionari sono in ogni caso esenti da responsabilità di qualsivoglia natura: civile, amministrativa e penale e saranno dal sottoscritto tenuti indenni da qualsiasi richiesta o rivendicazione da parte di terzi;
7) che il contenuto della tesi non infrange in alcun modo il diritto d'Autore né gli obblighi connessi alla salvaguardia di diritti morali od economici di altri autori o di altri aventi diritto, sia per testi, immagini, foto, tabelle, o altre parti di cui la tesi è composta.

Luogo e data Bari, lì 19/12/2024 Firma Giovanni Grieco

Il/La sottoscritto, con l'autoarchiviazione della propria tesi di dottorato nell'Archivio Istituzionale ad accesso aperto del Politecnico di Bari (POLIBA-IRIS), pur mantenendo su di essa tutti i diritti d'autore, morali ed economici, ai sensi della normativa vigente (Legge 633/1941 e ss.mm.ii.),

CONCEDE

- al Politecnico di Bari il permesso di trasferire l'opera su qualsiasi supporto e di convertirla in qualsiasi formato al fine di una corretta conservazione nel tempo. Il Politecnico di Bari garantisce che non verrà effettuata alcuna modifica al contenuto e alla struttura dell'opera.
- al Politecnico di Bari la possibilità di riprodurre l'opera in più di una copia per fini di sicurezza, back-up e conservazione.

Luogo e data Bari, lì 19/12/2024 Firma Giovanni Grieco

"*Le onde elettromagnetiche di per sé non generano luce. Tutto è buio nel cosmo [...] Si potrebbe dire che il cosmo si accende solo quando appare l'uomo, che sa non soltanto vedere queste luci, ma interpretarle.*" – Piero Angela

Abstract

6G networks aim to deliver ultra-low-latency global broadband through Non-Terrestrial Networks (NTN), which integrate drones, high-altitude platforms, and satellites. This project introduces contributions for advancing NTN capabilities, starting with the Drone Control Layer (DCL), a middleware solution enabling efficient operation of mixed drone swarms, equipped with interfaces for hardware abstraction, communication, and drone interconnectivity. Furthermore, IoD-Sim is presented, an open-source simulator for modelling NTN environments, including Intelligent Reflecting Surfaces (IRS) for optimised coverage. IoD-Sim allows realistic simulation of communication protocols and drone mobility in diverse scenarios. In the Internet of Things (IoT), drones extend IoT device battery life through Wireless Power Transfer (WPT), efficiently transmitting data to CubeSats, which simulations show significant gains in data transfer. Additionally, safety and security are addressed via Explainable AI (XAI), for drone spatial awareness, and Counter-Unmanned Aircraft System (C-UAS), for unauthorised drone detection using multi-sensor fusion techniques. Finally, a secure service chain model introduces custom authentication and authorisation to safeguard data flow in Terrestrial/NTN systems, integrated in a comprehensive cloud-based service oriented architecture. These contributions support the development and deployment of resilient space-air-ground integrated communication services, benefiting both research and industry.

Contents

List of Acronyms

3D three-dimensional

3GPP 3rd Generation Partnership Project

5G fifth generation

6G sixth generation

AA authentication and authorisation

ABAC attribute-based access control

AF amplify-and-forward

AI artificial intelligence

API application programming interface

AST abstract syntax tree

ATC air traffic control

ATR automatic target recognition

AWGN additive white Gaussian noise

BS base station

BSS basic service set

BVLoS beyond visual line of sight

C-UAS counter-unmanned aircraft system

CDF cumulative distribution function

COTS commercial off-the-shelf

CSF cyber-security framework

CSMA carrier sense multiple access

DCL Drone Control Layer

DL deep learning

DLU digital land use

DOC Drone Orchestration Centre

DTI detection, tracking, and identification

DVB digital video broadcasting

DVB-T digital video broadcasting - terrestrial

EbI eastbound interface

EIRP effective isotropic radiated power

eNB evolved Node-B

EU European Union

FANET flying ad-hoc network

FBS flying base station

FDMA frequency division multiple access

FSM finite state machine

GDI Global Drone Identifier

GEO geostationary equatorial orbit

GN ground node

GNSS Global Navigation Satellite System

GRI Global Role Identifier

GSO geosynchronous orbit

GU ground user

GUI graphical user interface

HAP high altitude platform

HTTP Hyper-Text Transport Protocol

IdP identity provider

IEEE Institute of Electrical and Electronics Engineers

IMU inertial measurement unit

IoD Internet of Drones

IoD-Sim Internet of Drones simulator

IoT Internet of Things

IP Internet Protocol

IR intermediate representation

IRS Intelligent Reflective Surface

IS identity server

ITU-R International Telecommunication Union Radiocommunication Sector

JRC Joint Research Centre

JSON JavaScript Object Notation

JWT JSON Web Token

JWKS JSON Web Key Sets

KPI key performance index

LASER Light Amplification by Stimulated Emission of Radiation

LEO low Earth orbit

LiDAR Light Detection and Ranging

LightGBM Light Gradient Boosting Machine

LoS line of sight

LSA Logical Swarm Address

LTE Long-Term Evolution

MAC Medium Access Control

MCC Mission Control Centre

MCL maximum coupling loss

MCS modulation and coding scheme

MEO medium Earth orbit

MINLP mixed-integer non-linear programming

MISO multiple-input-single-output

ML machine learning

NAT Network Address Translation

NbI northbound interface

NIST National Institute of Standards and Technologies

NLoS non line of sight

NN neural network

NOMA non-orthogonal multiple access

NR New Radio

ns-3 Network Simulator 3

NTN non-terrestrial network

OFDMA orthogonal frequency division multiple access

PCH pre-compiled header

PLR packet loss ratio

PoI point of interest

PRU passive reflective unit

PTZ pan-tilt-zoom

QoS quality of service

RADAR Radio Detection and Ranging

RAN radio access network

REM radio environment map

REST representational state transfer

RF radiofrequency

RoI region of interest

ROS Robot Operating System

RSSI received signal strength indicator

SAGIN space-air-ground integrated network

SAR synthetic aperture RADAR

SASL simple authentication and security layer

SbI southbound interface

SCA successive convex approximation

SDR software-defined radio

SHAP shapley additive explanations

SINR signal-to-interference-plus-noise ratio

SNR signal-to-noise ratio

SOAP Simple Object Access Protocol

SONAR Sound Navigation and Ranging

SQL Structured Query Language

SSID service set identifier

SWaP size, weight, and power consumption

T/NTN terrestrial/non-terrestrial network

TCP Transmission Control Protocol

TDMA time division multiple access

TLS Transport Level Security

TN terrestrial network

TR technical report

TS technical specification

TSDB time series database

UA unmanned aircraft

UAS unmanned aircraft system

UAV unmanned aerial vehicle

UDP User Datagram Protocol

UE user equipment

UPA uniform planar array

WbI westbound interface

WPT wireless power transfer

XAI explainable artificial intelligence

XML extensible markup language

ZSP zone service provider

List of Figures

List of Tables

Personal Scientific Contributions

Works published during my doctoral endeavour or still waiting for peer review are listed hereby.

International journals

- Giovanni Grieco, Giovanni Iacovelli, Pietro Boccadoro and Luigi Alfredo Grieco. **Internet of Drones Simulator: Design, Implementation, and Performance Evaluation**. *IEEE Internet of Things Journal* 10:2 (2023), 1476–1498. DOI: 10.1109/JIOT.2022.3207324

- Giovanni Grieco, Giovanni Iacovelli, Daniele Pugliese, Domenico Striccoli and Luigi Alfredo Grieco. **A System-Level Simulation Module for Multi-UAV IRS-Assisted Communications**. *IEEE Transactions on Vehicular Technology* 73:5 (2024), 6740–6751. DOI: 10.1109/TVT.2023.3342298

- Giovanni Iacovelli, Giovanni Grieco, Antonio Petrosino, Luigi Alfredo Grieco and Gennaro Boggia. **Fair Energy and Data Rate Maximization in UAV-Powered IoT-Satellite Integrated Networks**. *IEEE Transactions on Communications* 72:4 (2024), 2457–2469. DOI: 10.1109/TCOMM.2023.3343417

International conferences

- Giovanni Grieco, Domenico Striccoli, Giuseppe Piro, Raffaele Bolla, Gennaro Boggia and Luigi Alfredo Grieco. **Authentication and Authorization in Cyber-Security Frameworks: a Novel Approach for Securing Digital Service Chains**. In: *2022 IEEE 8th International Conference on Network Softwarization (NetSoft)*. 2022, 468–473. DOI: 10.1109/NetSoft54395.2022.9844030

- Giovanni Grieco, Giovanni Iacovelli, Pietro Boccadoro and Luigi Alfredo Grieco. **On the Design of the Drone Control Layer**. In: *European Wireless 2021; 26th European Wireless Conference*. 2021, 1–7

- Giovanni Grieco, Giovanni Iacovelli, Mattia Sandri, Marco Giordani, Michele Zorzi and Luigi Alfredo Grieco. **Preliminary Performance Evaluation of a Satellite-to-HAP Communication Link**. In: *European Wireless 2023; 28th European Wireless Conference*. 2023, 340–345

Book chapters

- Alessandro Carrega, Giovanni Grieco, Domenico Striccoli, Manos Papoutsakis, Tomas Lima, José Ignacio Carretero and Matteo Repetto, 1–31. In: *Cybersecurity of Digital Service Chains: Challenges, Methodologies, and Tools*. Ed. by Joanna Kołodziej, Matteo Repetto and Armend Duzha. Cham: Springer International Publishing, 2022. ISBN: 978-3-031-04036-8. DOI: 10.1007/978-3-031-04036-8_1

Technical reports

- Giovanni Grieco, Danilo Amendola and David Anderson. **Counter-drone systems and data fusion**. Tech. rep. KJ-01-24-196-EN-N. ISSN: 1831-9424. ISBN: 978-92-68-22705-3. Luxembourg (Luxembourg): Publications Office of the European Union, 2024. DOI: 10.2760/6037951

Submitted manuscripts, waiting for revision

- Giovanni Grieco, Arcangela Rago, Gennaro Boggia and Luigi Alfredo Grieco. **Cyber-Physical Spatial Awareness for Safe UAVs in XAI-enhanced 6G Networks: Vision, Challenges, and Opportunities** (2024). Submitted to *IEEE Network Magazine*, pending peer review

- Giovanni Grieco, Daniele Pugliese, Adnan Rashid and Ilaria Cianci. **IoD-Sim: an Open-Source Simulator for 6G Integrated Terrestrial/Non-Terrestrial Networks** (2024). Submitted to *Elsevier Computer Networks – Software Article*, pending peer review

Introduction

6G networks are set to provide sustainable, reliable, ubiquitous, and ultra-low-latency mobile broadband communications for a variety of industries. In order to guarantee coverage across the globe, non-terrestrial networks (NTNs) represent a key component for the development of three-dimensional (3D) networks, which envisions the integration of terrestrial with non-terrestrial mobile infrastructure. Indeed, NTN is a networking architecture that stems from the interplay between Internet-connected unmanned aerial vehicles (UAVs), high altitude platforms (HAPs), and satellites enabled by wireless communication technologies. These networked entities can unleash disruptive scenarios in many application domains, including, but not limited to, ubiquitous connectivity, mobility as a service, emergency response, monitoring, delivery, and surveillance. At the same time, to really capitalise on their potential, accurate modelling techniques are required to seize the fine details that characterise the features and limitations of UAVs, HAPs, satellites, wireless communications, and networking protocols. A key challenge for the NTN is further represented by the technological fragmentation of its components, from the hardware to their firmware and their integration with cloud services.

To this end, this Ph.D. project firstly focused on designing a middleware solution, namely Drone Control Layer (DCL), which enables complex mission design when heterogeneous swarms of drones are considered. The DCL is made of four interfaces which abstract underlying drivers and hardware, provide a set of common primitives to applications, enable communications between drones, and other logical entities. To illustrate its applicability, relevant scenarios of interest are analysed in details.

Furthermore, it has been proposed an overhauled version of the Internet of Drones simulator (IoD-Sim), a comprehensive and versatile open source tool that addresses the many facets of Internet-connected drones, with extensions to research and develop the integration of terrestrial/non-terrestrial networks (T/NTNs) and test disrupting 6G devices, namely Intelligent Reflective Surfaces (IRSs). IoD-Sim, based on Network Simulator 3 (ns-3), is organised into

a 3-layer stack and composed of (i) telecommunication primitives for different standardised protocol stacks, (ii) the implementation of key fundamental features of a 6G T/NTN scenario, and (iii) a set of tools that speeds up the graphical design for every possible use case. While NTNs, especially drones, are disrupting aerial mobility, the recent employment of IRSs in conjunction with UAVs introduces more degrees of freedom to achieve a flexible and prompt mobile coverage. As the concept of smart radio environment is gaining momentum across the scientific community, the simulator allows the assessment and the performance analysis of UAV-aided IRS-assisted communication systems. Starting from the mathematical formulation of the radio channel, extending 3rd Generation Partnership Project (3GPP)-compliant models when needed, the simulator implements the IRS as a peripheral that can be attached to a drone. Such device can be dynamically configured to organise the IRS into patches and assign them to assist the communication between two nodes. While UAVs can be simulated for groundbreaking use cases in low-altitude flights, the simulator allows their interconnection with HAPs and satellites, pursuing a combination of platforms that paves the way to wide coverage and reliable communication in remote and inaccessible areas, and/or where terrestrial infrastructure is unavailable. Simulation fidelity is ensured by also taking into account features beyond plain telecommunications, such as (i) Earth's curvature, (ii) the possibility to work with different coordinate systems, such as Cartesian and geographic, and (iii) realistic mobility patterns. Every aspect of the simulation can be designed in JavaScript Object Notation (JSON) language or through a graphical user interface (GUI), thus facilitating its configuration. In order to prove the huge potential of this proposal, a scenario campaign is presented and analysed from both a software perspective and a telecommunications standpoint, discussing several key performance indicators (KPIs), such as radio environment map (REM), signal-to-interference-plus-noise ratio (SINR), maximum achievable rate, and average throughput. The peculiarities of this open-source tool are of interest for researchers in academia, as they will be able to extend it to model upcoming specifications, including, but not limited to, mobile and space communications. Still, it will certainly be of relevance in industry to accelerate the design phase, thus reducing the time to market of 6G T/NTN-based services.

NTNs represent a valuable solution also for Internet of Things (IoT) devices in remote areas. Due to the low-power nature of IoT devices, an UAV can prevent the energy depletion of these ground nodes (GNs) by employing wireless power

transfer (WPT) through an array antenna. Starting from the mathematical modelling of such a scenario, two mixed-integer non-linear programming (MINLP) problems are formulated to fairly maximise (i) the energy distribution and (ii) the total amount of data transmitted to a low Earth orbit (LEO) CubeSat [3]. Therefore, it is necessary to optimise the drone kinematics, the transmission scheduling plan, and the beamforming vectors of the array antenna. To cope with their non-convexity, both problems are mathematically manipulated to reach a tractable form, for which two optimisation algorithms are proposed and their complexity analysed. To prove the effectiveness of the overall solution, a comprehensive simulation campaign is conducted under several parameter settings, such as number of GNs and UAV antenna elements with different transmission power levels. Finally, the proposal is compared with a baseline, which confirms the superiority of the proposal up to 7 times in terms of total transmitted data.

While compelling use cases were simulated and proposed, safety and security concerns are of paramount discussion, as the proliferation of drones leads to a wide range of legitimate, nuisance, and malicious applications.

Regarding safety, a major challenge for UAVs is their limited trustworthiness and spatial awareness, especially in complex environments, such as smart cities. Artificial intelligence (AI), a native component of 6G, offers solutions throughout the 6G architecture. However, the opacity of AI, particularly in deep learning (DL) models, presents reliability and trust issues. In this regard, explainable artificial intelligence (XAI) represents a compelling solution, enhancing model transparency, performance, and robustness, thereby ensuring safe UAV missions. Indeed, by leveraging XAI, UAVs can better plan and execute missions using high-resolution geospatial data, network coverage maps, and weather forecasts. During flights, real-time information updates and telemetry enhance situational awareness and their safety. Thus, XAI facilitates understanding and decision-making at all mission stages, ensuring UAV cyber-physical spatial safety.

Regarding security, instead, to address the growing concerns surrounding unauthorised drone usage, the counter-unmanned aircraft system (C-UAS) technology has been developed to mitigate potential threats. To this end, the complexities of C-UAS implementations are further examined, focusing on the detection, tracking, and identification (DTI) of unauthorised unmanned aircraft systems (UASs) in public spaces and critical infrastructures. Given the vast and diverse nature of sensors used in C-UAS, the study highlights the importance of multi-sensor data fusion for accurate DTI in real time. It presents a taxonomy of

sensors used in C-UAS, emphasising the need for a comprehensive, integrated approach to enhance system reliability and effectiveness. Finally, it discusses the challenges of implementing a multi-sensor data fusion algorithm through a meta-study analysis of the current literature.

Security concerns must be further addressed when these mobile entities are integrated with modern computing paradigms (i.e., cloud, edge, and IoT.) This integration is bringing connectivity and pervasive computing to an unforeseeable level, which boosts service-oriented architectures and microservice patterns to create digital services with data-centric models. In such a scenario, it is important to guarantee data confidentiality, integrity, as well as authentication and authorisation (AA) procedures between the communicating parties of a service chain. Cyber-security frameworks (CSFs) are explicitly designed for this purpose. They rely on the integration of different software modules, mutually interfaced to accomplish complex security tasks. Nevertheless, it is important to guarantee a high level of protection during data exchange among the modules. Currently, standardised AA mechanisms are implemented through proprietary "as-a-service" products, but the deployment of a mature on-premise solution is still missing. To bridge this gap, is it proposed an AA module that automatically protects the information flowing among the modules of CSFs. It guarantees resource availability only to authenticated subjects. Thus, their operations are confined in what actions they are authorised for. Experimental tests show that the proposed module enables AA procedure delegation among CSF modules, which eases their implementation, while maximizing the flexibility of the set of access control policies and an efficient protection of the services.

In order to provide more insights on all these discussion points, the thesis has been structured as follows:

- Chapter 1 introduces the main background common to all these works.
- Chapter 2 proposes the Drone Control Layer.
- Chapter 3 presents, analyses, and discusses the Internet of Drones simulator.
- Chapter 4 represents a collection of use cases and simulations to demonstrate the value of the simulator in developing 6G T/NTN scenarios. Furthermore, a convex optimisation problem is faced for the study of T/NTN architecture integrated with low-power IoT devices.

- Chapter 5 highlights the potential impact of XAI in spatial situational awareness of drones in complex environments.
- Chapter 6 discusses methods and key challenges in detecting, tracking, and identify drones in monitored air space.
- Finally, Chapter 7 proposes a federated security mechanism in the context of a cyber-security framework, in order to ensure secure data exchange between parties in a complex digital service chain.

To conclude, the main findings and future research directions are summarised in the closing chapter on page 191.

1 Introduction to Non-Terrestrial Networks

This chapter presents an overview of the context relevant to the core contributions discussed in the subsequent chapters, with a particular focus on the role of NTNs in future 6G communication systems.

Integrated T/NTNs represent one of the key enablers for future 6G wireless communication goals. They aim to provide long-distance, highly-mobile, and reliable low-latency communications. Additionally, they support other 6G pillars, such as enhanced mobile broadband and low-power ultra-massive machine-type communications [12]. These characteristics answer emerging needs of manifold verticals, such as eHealth, intelligent transportation systems, immersive multimedia entertainment, automotive, and cyber-physical security [13–15]. One of the most challenging aspect is that the growing number of users, as well as service diversity, has been enabled primarily by the expansion of traditional terrestrial wireless communication systems [16]. At the same time, emerging applications impose challenging requirements that must be addressed through the technological advancement of innovative telecommunication facilities [17]. In this context, 6G mobile infrastructure [13] promises an ubiquitous coverage across Earth that leverages an integrated access backhaul that unifies ground, aerial, and space infrastructures [18], thus flexibly extending and enhancing the traditional fifth generation (5G) ecosystem [13, 19]. Given the richness of Earth atmosphere at different levels of altitude, T/NTNs include a plethora of heterogeneous devices. As illustrated in Figure 1.1, traditional terrestrial infrastructure and users are distributed over the ground and on the oceans. UAVs can be deployed in the troposphere, HAPs in the stratosphere, and LEO, medium Earth orbit (MEO), geostationary equatorial orbit (GEO), and geosynchronous orbit (GSO) satellites in space. All of them cooperate for the realisation of NTNs in each segment of belonging, while interconnecting between segments to create a 3D wireless connectivity infrastructure, introducing different challenges on the design, implementation, and testing worth of studying. Notably, NTNs [20] are capable of bridging geographical divides, and provide broadband stand-alone connectivity even in absence of terrestrial networks (TNs) (e.g., in rural or remote

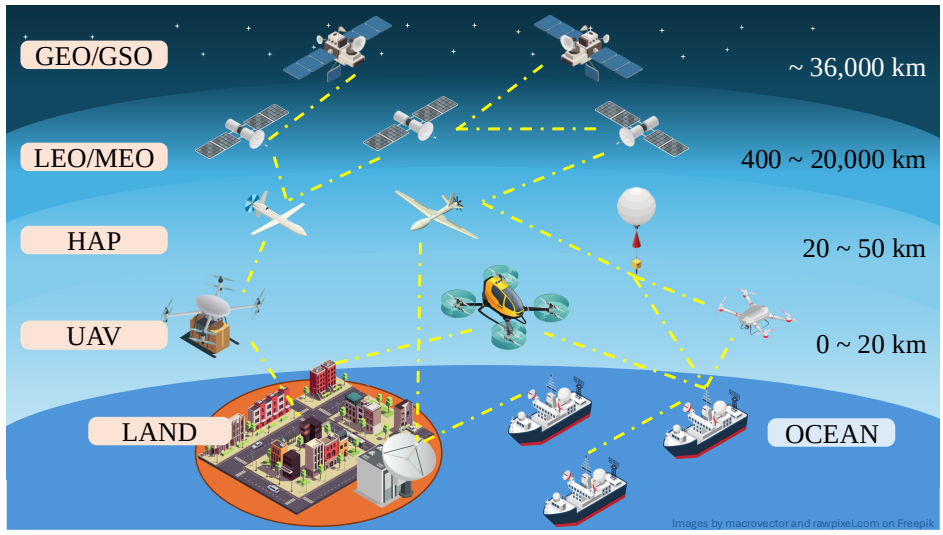


Figure 1.1: The envisioned integration of NTNs, made by UASs, HAPs, and satellites, with land and ocean TNs for 6G communications. Altitudes reported on the right are indicative [11].

areas) or when TNs are unavailable (e.g., in case of natural disasters.) In the context of NTNs, three key technologies that hold very high potential are UAVs, HAPs, and satellites.

1.1 The Internet of Drones

UAVs [21], also known as drones, are highly-mobile and versatile platforms that implement a variety of use cases, ranging from the consolidating last-mile delivery to the experimental flying base station (FBS). They can be multi-copters and / or fixed-wing in structure and, according to their mechanical implementation, they are constrained in terms of the possible hardware that can be installed, thus limiting the computational, memory, energy, and communication capabilities. The interconnection of UAVs with the rest of the Internet [22] gave rise to the Internet of Drones (IoD) [23], which eases trajectory planning, mission design, flight control, resource optimisation, and swarm management at scale [24]. At first, it might appear as an extension of the IoT, with unmanned aerial vehicles (UAVs) playing the role of smart objects able to fly. Nevertheless, in

the IoD, drones are tasked with completing mission plans with multiple objectives. Since they can also fly in organised groups, namely swarms, it is worth remarking that they are made able to continuously optimise their trajectory and coordinate among themselves. Drones are currently involved in the delivery of value-added services in many applications, especially in smart cities [25–28] and in harsh environments [29] (e.g., oceans, deserts, and hazardous places,) including goods delivery, environmental surveying, first-aid units in disruptive events [23, 28], and FBS in 5G & beyond scenarios, providing on-demand connectivity to multiple users [23, 28, 30–33]. Smart cities are among the most challenging application scenarios, with ever-changing players and behavioural patterns, which makes it hard to address public safety requirements, especially at scale [34]. All this turned the IoD from a niche subject to a mainstream research topic in networking, paving the way for an integrated 3D network of terrestrial and non-terrestrial mobile infrastructure [29]. It must be noted that the adoption of drones in the industry is also a huge commercial opportunity, as testified by the several forecasts already available for multiple business sectors [23]. Even though several applications are now including drones, and they may look like commercial off-the-shelf (COTS) utilities, the design of complex IoD systems still requires advanced methodologies to effectively unleash the potential of services based on networked drones. Moreover, given the variety of available drones on the market, an accurate suitability assessment based on their characteristics is required.

1.2 UAVs enhanced with Intelligent Reflective Surfaces

Along the physical properties of these network entities, more research on mobile communications is critical for the upcoming network infrastructure. For instance, 6G is pushing to (i) improve sub-6 GHz spectrum efficiency, (ii) unlock mmWave and THz communications, and (iii) employ metasurfaces [35].

One of the most challenging aspects that these systems encounter is the Shannon capacity limit, which is especially bounded by the available bandwidth. For this reason, the research and standardisation communities are focusing on mmWave and THz spectrum to unlock ultra-wide channel capacity [36–38]. Nonetheless, the environment can also be controlled to turn adverse effects, such as multi path, into advantages [35]. In this regard, IRSs [39] allow to

control the radio environment by optimally reflecting incident electromagnetic waves through a matrix of passive reflective units (PRUs), thus yielding passive beamforming [40]. Differently from the traditional antenna array systems, IRSs can not only be deployed as fixed, stand-alone entities on buildings, but they also satisfy size, weight, and power consumption (SWaP) constraints of drones. Consequently, the integration of IRSs and UAVs [40–43] leads to more degrees of freedom that can be properly tuned to cope with the ever-changing channel condition, providing the possibility to re-establish the line of sight (LoS) and to reduce the overall path loss.

1.3 High altitude platforms and satellites

To some extent, HAPs are subject to constraints similar to UAVs in order to maintain a stable flight in the stratosphere. Moreover, it is imperative to differentiate the type of satellites participating in the communication system of interest, as it ranges from small CubeSat deployed in mega-constellations at LEO height, which is subject to non-negligible Doppler shifts, to GEO one, which follows the Earth's rotation [44].

HAPs, also known as stratospheric platforms, are UAVs soaring in the stratosphere at altitudes ranging from 20 to 50 km. These platforms can be equipped with propulsion systems, typically based on propellers and electric motors, to move to different locations [45]. This capability allows them to be deployed as needed, providing coverage to specific areas or addressing dynamic communication demands. Moreover, HAPs can establish wireless links with satellites, other HAPs, low-altitude UAVs, and/or terrestrial networks. Indeed, flying at high altitudes, HAPs can offer wide coverage and great LoS connections, and establish reliable communication links in previously inaccessible regions. In turn, satellites have been used for decades, primarily for navigation, meteorology, and television broadcasting. However, with the advent of 5G, satellites are now considered as an integral part of the communication infrastructure, to support cost-effective, high-capacity, wide-coverage connectivity on the ground [46, 47].

The integration of HAPs and satellites with TNs brings several advantages. First, these aerial and space platforms can effectively bridge the digital divide by bringing high-speed connectivity to remote areas where ground infrastructure is limited or absent [48]. Moreover, HAPs and satellites play a key role in disaster response and recovery scenarios, providing emergency communication

networks when TNs are (temporally or permanently) disrupted or unavailable. Furthermore, as the demand of data-hungry applications increase, HAPs and satellites can supplement existing TNs and relieve congestion by offloading traffic. Additionally, they support critical applications requiring ubiquitous and uninterrupted connectivity in remote or mobile environments, such as for autonomous vehicles, smart cities, and IoT devices. However, this potential can be maximised if HAPs and satellites work together as a multi-layered integrated network [49], rather than as stand-alone solutions. For instance, the HAP layer can act as a wireless relay to improve the link quality of an upstream satellite. At the same time, the satellite layer can offer the HAP a ready-to-use link for the backhaul, as well as an alternative access to the core network.

All of this brought space-air-ground integrated networks (SAGINs) [50] to represent a flexible solution to provide wireless access services with high data rate and reliability, which are key enablers for a variety of both civil and military applications. Furthermore, recent 3GPP standardisation efforts [51–54] identified non-terrestrial networks (NTNs) [29, 55] as a solution to grant connectivity where traditional TNs are not practical or cost-effective. Currently, low Earth orbit (LEO) satellite constellations are employed to provide full-coverage broadband services for ground users through space-ground interconnection. Manufacturing and launching processes for these constellations have matured, enabling the implementation and deployment of these systems at scale [56, 57].

1.4 Enhanced Internet of Things through wireless power transfer and non-terrestrial infrastructure

Network architectures can benefit from their high mobility, easy deployment, and reusability [23]. Specifically, drones play a pivotal role in the realm of the IoT [58], representing an enabling technology to provide pervasive connectivity even where the classical communication infrastructure is not available. The IoT allows the interconnection between physical and digital realms, revolutionizing industries by offering disruptive prospects for automation, efficiency, and data-driven decision-making.

The integration of satellite and UAV communications in the IoT domain enables real-time monitoring, autonomous operations, and novel solutions across industries such as agriculture, transportation, and surveillance. This combina-

tion results in a full ecosystem, propelling progress toward a smarter and more connected society.

Despite the great advantages in terms of seamless and reliable connectivity, the energy lifetime of IoT devices represents a challenging aspect that is usually not taken into account, especially in harsh environments. To this end, WPT [59] has been recognised as an effective solution to cope with this issue. In traditional WPT systems, specialised energy transmitters are installed at fixed locations to send radiofrequency (RF) signals to charge IoT nodes, especially low-power ones. However, the range of these systems is limited by the low efficiency of end-to-end power transmission over long distances. Therefore, fixed-location energy transmitters must be densely deployed to wirelessly recharge a large number of low-power devices, which would significantly increase the cost and hinder large scale implementation.

To tackle this issue, the majority of the scientific literature focuses on the combination of WPT and UAVs as a solution to support an IoT network in terms of power delivery and information transmission. In particular, they focus on the optimisation of different aspects, such as the movements of the UAV [60–67], power allocation [61, 63–65, 67, 68], energy harvesting time [62–65, 68, 69], and the beamforming vectors of the antenna [66, 67].

However, the state of the art does not consider the potential of satellites, and their integration with drones and WPT, as a comprehensive solution for IoT networks in harsh conditions.

To this end, it is compelling to combine these technologies in order to investigate an UAV-powered IoT-satellite integrated network, where a drone wirelessly recharges a set of GNs, while a LEO CubeSat provides connectivity for data exchange. Specifically, the objective is to achieve a fair maximisation in terms of harvested energy and transmitted data.

On this topic, the scientific literature is currently focusing on enhancing traditional IoT networks by (i) expanding their coverage and (ii) improving the battery life of the devices. Regarding the former, some intriguing contributions include the design of (i) communication and protocol schemes by adapting terrestrial technology to the space segment [70], (ii) resource allocation schemes able to improve energy efficiency [71], and (iii) more reliable LEO satellite-terrestrial communication techniques [72]. In this context, UAVs employed as FBS represent a valuable methodology to achieve ubiquitous connectivity.

For instance, the authors in [73] aim at improving the perceived network qual-

ity by the user and minimizing the communication outages, while enhancing the data rate and the fairness of the transmission. Most noteworthy, the efforts placed a great emphasis on the pairing of these two technologies by defining a hybrid network that leverages UAVs as relays to support satellite communications. In this context, the spectral efficiency and the outage probability are optimised in [74], by proposing a UAV relay selection and power allocation scheme. Other contributions design the transmission scheduling and the UAV trajectory to increase the system capacity [75] and energy efficiency through non-orthogonal multiple access (NOMA) [76].

For what concerns the battery life of the devices, WPT emerged as a disruptive technology for energy harvesting [77]. For instance, the authors in [78] propose a method that allows a node to first gather energy and then use it to transmit. Specifically, they investigate the optimal duration of a timeslot in a time division multiple access (TDMA)-based protocol, which maximises the spectrum efficiency. Other approaches also consider the presence of an IRS [79] to maximise the transferred power [80] and the throughput [81] of the users.

Moreover, cutting-edge approaches integrated the potential of WPT technology with the high mobility of the UAVs. A first setup is envisioned in [82] and [83], where energy-constrained nodes are optimally served by a UAV which acts as a FBS, powered by WPT.

Recent works are also exploring the employment of UAV as a stand-alone WPT source. Clearly, one of the most critical aspects to be optimised is the UAV trajectory, which affects many facets of the mission, such as (i) the total amount of collected data [65], (ii) the age of information [69], (iii) the energy harvested by the GNs [60, 61, 64, 66, 67], (iv) the out-of-service probability [62], and (v) the UAV power consumption [63]. In particular, a novel scheme aided by an IRS, which simultaneously addresses WPT and information transmission for IoT sensors, is proposed in [65]. The protocol is divided into two phases: in the first one the drone recharges the devices while in the second one it gathers the data. The objective is to maximise the total network sum rate by optimally deriving the trajectory, the power allocation, the energy harvesting scheduling of the nodes, and the phase-shift matrix of the surface. Furthermore, the authors in [69] investigate a scenario in which a UAV recharges the GNs, collects data, and then transfers them to a data center. The aim is to minimise the average age of information by jointly optimizing the UAV trajectory and the time allocated for WPT. Moreover, the approach proposed in [60] intervenes by supplying power

at GNs with limited battery capacities deployed in remote areas. Given that UAV is employed as a wireless power supplier and data collector, its overall energy consumption must be optimised subject to task collection and resource budget requirements. In [62], it is discussed the minimisation of the energy depletion of GNs, and hence their outage probability. The latter is subject to the UAV elevation angle and the time slot allocation between the energy harvesting and the information transmission of each GN. Finally, the authors in [63] study a scenario in which a UAV is in charge of sustaining the devices of a network by periodically flying back and forth from a fixed position. Two approaches are proposed to minimise the average UAV power consumption by determining the trajectory, the duration of working periods, and the charging phase. The works above, however, consider a single antenna to perform WPT. To fill the gap, the approaches proposed in [66] and [67] investigate the impact of an antenna array to increase energy efficiency by taking advantage of beamforming.

1.5 Toward 3GPP 6G use cases and requirements

Normative work is being conducted to evolve the network infrastructure into a global ubiquitous and interoperable system. Initiatives by 3GPP standardisation for NTN are greatly summarised in [84], including use cases, requirements, spectrum allocation, current industrial endeavours, and a history of 3GPP releases focused on the convergence of TNs with NTNs.

The 3GPP technical report (TR) 22.822 [85] outlines a set of use cases for 5G NTNs, designed to complement traditional TN services. These use cases prioritise service continuity in underserved areas, service ubiquity for global coverage, and service scalability for extended coverage and traffic offloading. Key applications include global coverage for NB-IoT and mMTC, smart goods tracking, governmental activities (e.g., border surveillance, event monitoring, secure communication, and traffic management), disaster response (e.g., drone-based access points with LEO satellite backhaul), and remote monitoring of critical infrastructure for operations without latency requirements and continuous data transfer [84].

Advancements in 5G NTN, i.e. 3GPP Release 17 onwards, focus on standardising enhanced multicast and broadcast services to enable widespread distribution of television and digital data across large areas [84].

Looking ahead to 6G, a unified 3D layered system architecture is proposed,

integrating TNs and NTNs to deliver the desired quality of service (QoS) for diverse use cases. These include (i) enhanced mobile broadband (e.g., multimedia access), (ii) immersive communications (e.g., holography, mixed reality), (iii) ultra-massive machine-type communications for even smarter and connected IoT, (iv) ultra-critical operations (e.g., remote surgery, haptic Internet), (v) integrated communication and sensing for environmental and resource optimisation, and (vi) distributed AI at the edge. While 6G is currently in the conceptual phase, future developments will clarify and refine its use cases [84].

Regarding the integration of drones in the 3GPP ecosystem, work has been done since Release 16, where technical specification (TS) 22.125 [86], TS 23.256 [87], and TR 22.825 [88] define operations (the first two) and use case scenarios (the last one) for the remote identification and tracking of UAS, for which it is expected that the network infrastructure operates transparently for the timely delivery of such data to a remote unmanned traffic management system. Such works pave the way for beyond visual line of sight (BVLoS) flights of UAVs relying on the cellular network. Based upon these studies, TR 33.854 [89] provides a comprehensive overview on key security issues in adopting such use cases, while TS 23.255 [90] presents the capabilities necessary for applications in order to efficiently use the 3GPP network infrastructure.

Currently, for Releases 18 and 19, it is possible to notice improvements of the 5G specification wrt. the inclusion and integration of drones, particularly on (i) New Radio (NR) [91], its Medium Access Control (MAC) specification [92], and its service requirements [93]; (ii) radio access capabilities of user equipments (UEs) [94] and their transmission/reception [95]; (iii) radio resource control specification [96]; finally, (iv) the radio access network (RAN) application protocols NG [97], Xn [98], and F1 [99].

2 Envisioning a Drone Control Layer

This chapter introduces the DCL, a novel middleware architecture designed to facilitate the scalable deployment and management of drones, thereby advancing the realization of the IoD vision [22, 23].

The technological landscape of UAVs is extremely heterogeneous, therefore interoperability issues could arise in complex missions. As a matter of fact, current available applications strongly depend on specific hardware and software environments which threatens/slows down drones' employment at scale. To boost the deployment of the IoD, such dependencies should be avoided thanks to a *de-verticalizing* platform that grants portability. Furthermore, drone infrastructures require highly qualified personnel with expertise spanning over multiple aspects of hardware, mechatronics, network, and software engineering. A middleware with a flexible software interface eases the management of multiple subsystems and their requirements to be satisfied [100].

Thanks to the recent advancements in cloud and edge computing, the use of a middleware in the IoD enables complex mission design, off-the-shelf software-defined components, integrated service provisioning, and management at a glance. Mission plan can be envisioned as a composition of different containers in a micro-service development environment [101]. On these bases, this work designs a middleware solution, namely DCL, that is located between the transport and application layer. It helps to define the underlying platform to abstract mission planning from drone peculiarities, while providing a safe and unified control structure. Meanwhile, the DCL grants the management of drones by means of a common set of interfaces with predictable responses. This facilitates applications' portability. The DCL identifies each elementary component of a given mission plan in order to further assign them to dedicated, yet specific, core modules, which cooperate towards mission accomplishment. The middleware has four interfaces which enable (i) the abstraction of underlying drivers and hardware, (ii) the use of common primitives for application development, (iii) the communication between drone and other logical entities, and (iv) core functionality extensibility.

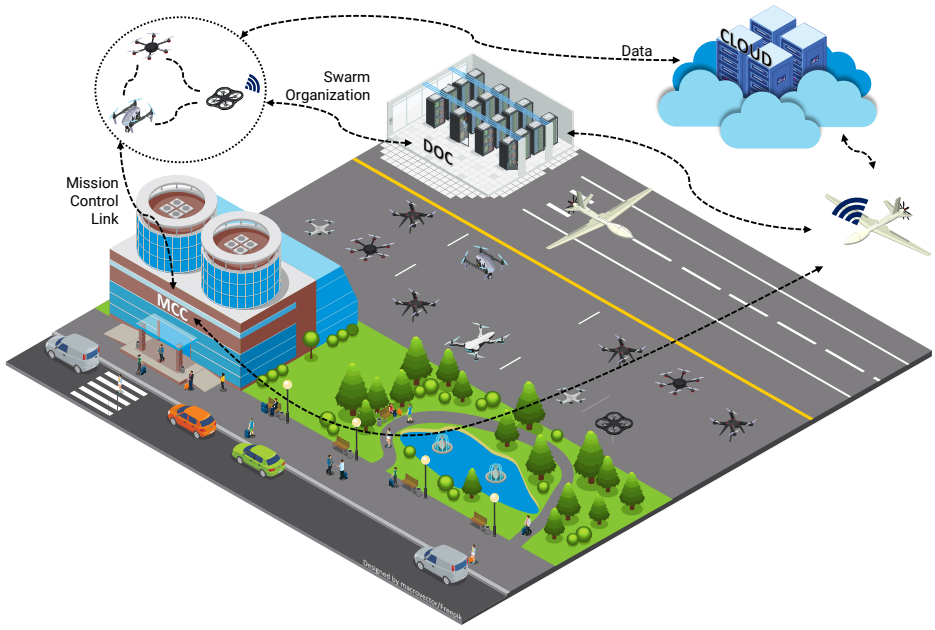


Figure 2.1: Schematic representation of the envisioned scenario.

Differently from the current state of the art [102–107], which envisions drone as stand-alone entities, the DCL allows seamless management and coordination among swarms of heterogeneous drones. Moreover, the DCL enables the deployment across multiple unmanned systems.

Inspired by cloud and edge practices, DCL-enabled drones working together with other logical entities in the IoD (e.g., unmanned traffic management, air traffic control (ATC) services, weather stations, and recharging stations) can be envisioned as resources, thus naturally becoming platforms as a service. The proposed solution allows a wide applicability in several scenarios of interest. For the sake of concreteness, public safety and FBS scenarios are deeply investigated.

2.1 Reference scenario

The reference scenario (Figure 2.1) envisions swarms of drones assigned to different missions. Each mission is composed by a specific set of operations. Drones are grouped to form a swarm based on their on-board equipment and

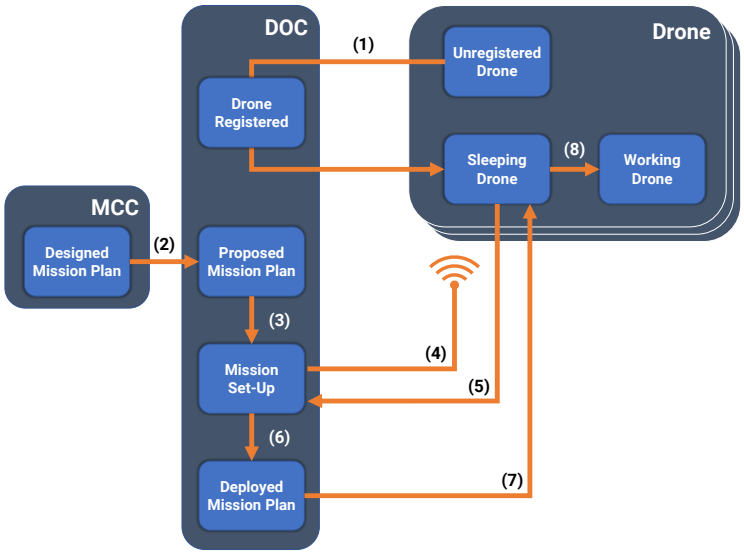


Figure 2.2: Addressing solution for the identification of the swarms and roles related to drones.

capabilities. Mission assignment and drones' enrolment are handled by dedicated logical entities:

- Mission Control Centre (MCC): ground control infrastructure that designs the mission and monitors the swarm.
- Drone Orchestration Centre (DOC): a central hub used to enrol drones in a mission by matching their characteristics with the operations to be done.
- Cloud-based Application Services: high-level functionalities for data mining and analysis related to the mission.

Drones are assigned to a specific mission before take off. The assignment procedure is illustrated in Figure 2.2 and composed by the following steps:

1. Drones join the IoD network thanks to a registration procedure at the DOC. On top of it, the drone is able to announce its availability to the network. As a consequence, the drone is enlisted among those that the

DOC can assign to a mission. To this end, the drone receives a logical address, namely Global Drone Identifier (GDI).

2. The design of the configuration includes the mission plan and the list of roles to be covered within the mission. This is carried out by the MCC. Once the DOC is notified with the published configuration, drones can be assigned to the mission.
3. The DOC is now able to process the received information, to specify the mission goal in terms of required drones' capabilities, to achieve the desired objective.
4. The mission plan is announced to all drones in the IoD network.
5. The idle (i.e., not involved in a mission) and suitable (i.e., with the proper characteristics) drones are required to probe the pre-flight checklist and announce its outcome to the DOC. In particular, their compatibility can either be *partial* or *full*.
6. The received information are processed by the DOC to select the drones that are most suitable for the mission. As a consequence, a structure that couples each drone to its activity is set up.
7. The DOC leverages the GDI to communicate to each drone its assignment. It might happen that the number of idle drones does not cover the mission needs. In this case, the DOC notifies it to the MCC to abort the mission.
8. If the assignment is concluded successfully, a bootstrap procedure takes place to define the logical swarm domain. Members of the group become logically identifiable by a context-based addressing schema focused on swarm participation and drone identification.

Once the swarm formation setup is completed, drones can start exchanging data with the MCC and execute the mission tasks.

2.2 Interfaces characterisation

The architecture of the DCL is graphically introduced in Figure 2.3. The DCL includes four main interfaces: northbound interface (NbI), southbound interface (SbI), westbound interface (WbI), and eastbound interface (EbI).

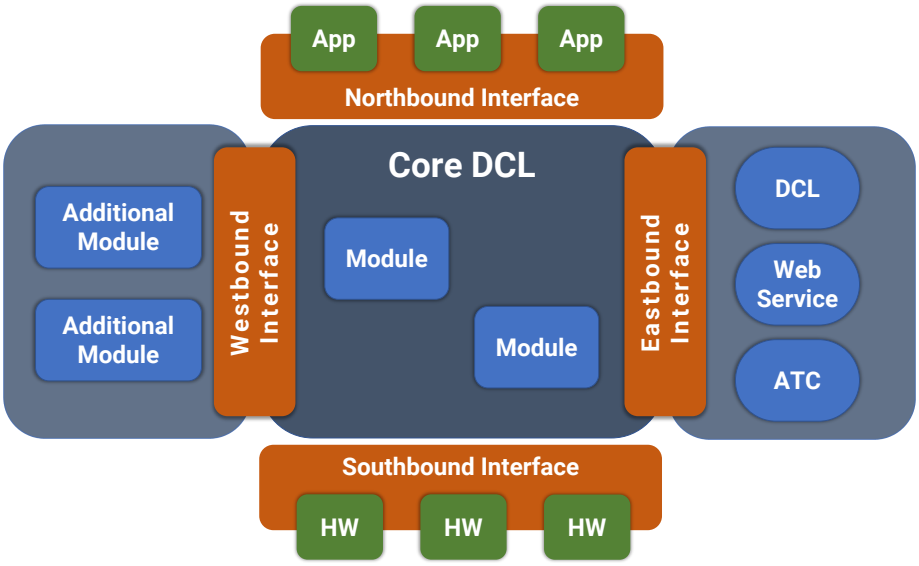


Figure 2.3: High-level overview of DCL Interfaces and their types of applicability.

2.2.1 Northbound Interface

The NbI supports the development of high-level applications on top of the DCL to ease drone's control, communications and mission planning. As illustrated in Figure 2.4, the NbI provides an event-driven notification mechanism that eases information exchanges towards upper-level applications. For instance, drones in a swarm are able to react to updates and dynamically adjust their configuration to newer roles, if needed.

At the same time, applications may use Input/Output functionalities to exchange data and information with other DCL modules according to their exposed features on the interface. Messages travel in both directions using push-based, pull-based, pub/sub mechanisms according to application design and its requirements. Through the NbI, the application can inspect the network and automatically discover its topology and recognise other network entities. It can also exchange mission information to cooperate for task accomplishment. In the same manner, it would be possible to optimise flight control operations and make use of hardware capabilities.

In this way the application can influence drone operations and profits from

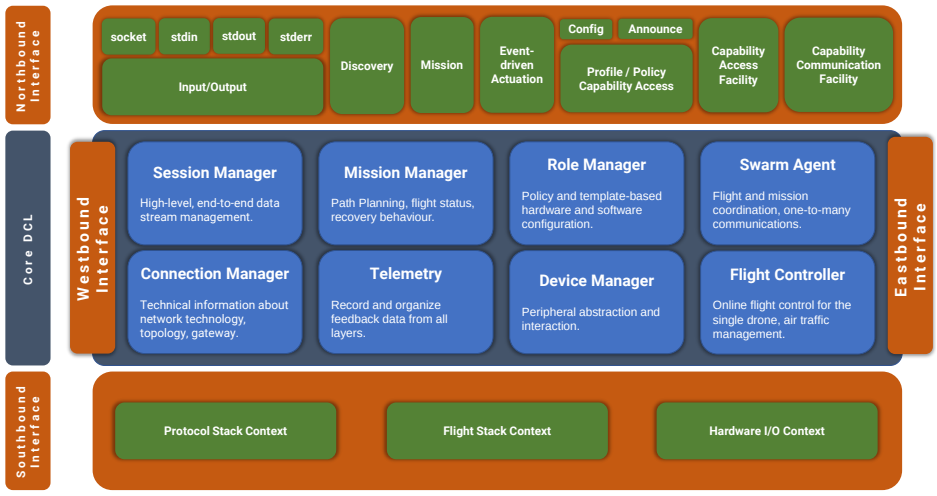


Figure 2.4: High-level architecture of the DCL.

its capabilities, while the DCL can ensure operations are correctly applied to the drone and notifies the application in case of state changes.

2.2.2 Southbound Interface

The SbI enables the interactions among the on-board resources and all the upper layers. Such a component solves the problems arising from vendor-related dependencies. In fact, this interface recognises and supports multiple protocol stacks, radio interfaces, flight control primitives, and hardware resources in order to provide a set of software abstractions. The SbI is organised in three logical contextual blocks, as depicted in Figure 2.4. The Protocol Stack Context abstracts on-board radio interfaces providing low-level networking primitives. The Flight Stack Context is a collection of drivers to control mechanical components of the drone and, hence, its motion. Finally, Hardware I/O Context aims at managing interactions with underlying peripherals, i.e., sensors and actuators.

2.2.3 Westbound Interface

The WbI enables the development of DCL integration with additional modules and customised algorithms in order to optimise operations and support emerging

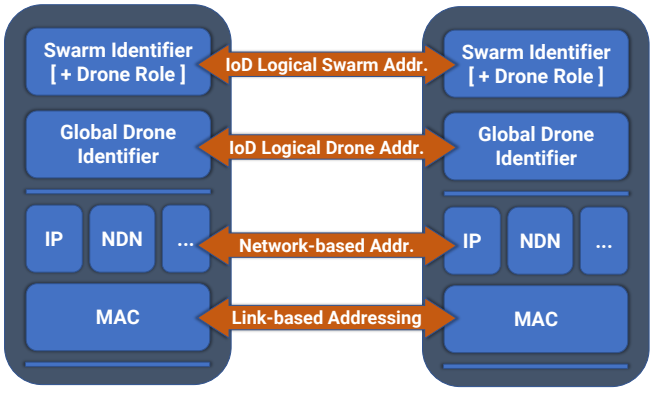


Figure 2.5: Logical addressing types supported by the DCL EbI.

applications. Extensions can cover a wide range of further developments, such as energy optimisation.

2.2.4 Eastbound Interface

The EbI allows horizontal logical communications with other DCL-enabled entities to synchronise information, e.g., tasks, missions, and establish swarm networks. It also allows to establish direct connections between the drone and the outer world, i.e., Internet.

DCL entities in the IoD can be reached by means of contextual logical addresses. In particular, as shown in Figure 2.5, addressing is two-folded: drones have an address of their own (i.e., the GDI,) while, in case they belong to a swarm, they can be addressed as a group entity. Such addressing scheme is independent from the particular communication stack used by drones and will be explained in detail in the following Section.

2.3 Core characterisation

The architectural components of the DCL, and its organisation (Figure 2.4,) can be considered as the base fabric of all primitives that assist the planning of a mission. These elements characterise the kernel of a general purpose drone.

What will be herein described may either be used as off-the-shelf functionalities or further extended to enable new services on top of them.

2.3.1 Device Manager

The Device Manager provides handlers to orchestrate drone's hardware in terms of its capabilities. As a consequence, the Device Manager can be used to validate the suitability of the drone for the specific mission plan. Each hardware component is managed by a specific *Device Driver*. The latter exposes multiple high-level capability objects to communicate with such hardware component. Moreover, it publishes a manifest which includes a description of hardware functionalities together with its state, as represented in Figure 2.6. For instance, from an application point of view, the Device Manager does not provide detailed information about the specific on-board camera. Instead, it will indicate the drone with an on-board camera as usable for a mission during which images/video signals have to be gathered.

A single capability offers a data-driven, bidirectional communication channel to exchange data with the *Capability Communication Facility*. The joint adoption of the Device Manager and its drivers decouples high-level communication at the NbI from the specific hardware communication methodology (sequential or random access, exclusive or multiple access) and the detection of the component itself (driver, bus, battery, peripheral, and hot-plug support.)

2.3.2 Role Manager

Leveraging the high-level description provided by the Device Manager, the Role Manager is responsible for matching drones to a mission plan. To this aim, the specific mission requirements are encoded in a structured format and described as follows:

- *Rule*: it enables the drone's configuration with reference to the working context and it is composed by:
  - *Subject*: the drone capability affected by the *Rule*.
  - *Predicate*: the action or condition that the *Subject* has to observe.

  Each *Rule* can either be *Observed* or *Not Observed*. To provide a concrete example, a certain amount of energy could be specified by the MCC as

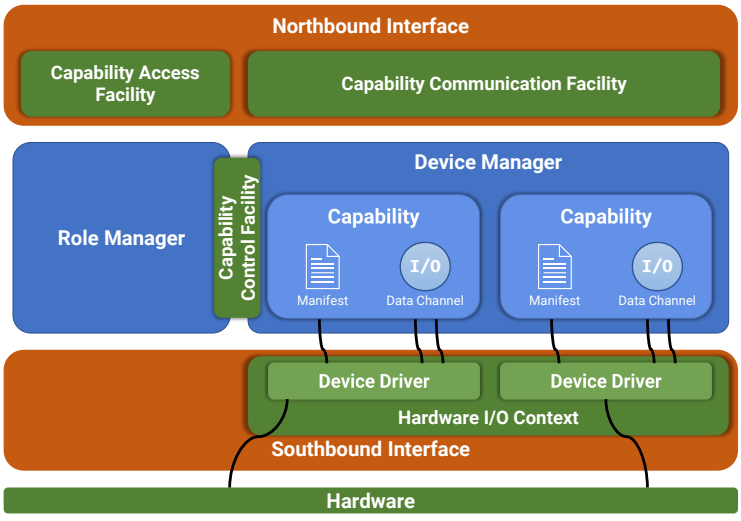


Figure 2.6: Device Manager schematic characterisation.

a requirement for the given mission plan. In case the condition for that mission is satisfied, the *Rule* is set to *Observed*.

- *Policy*: it is described by a decision tree, where each node uniquely identifies a *Rule*. The tree exploration is performed to evaluate drone suitability, given the mission requirements, encoded in these Rules. This structure has been chosen to ease rules evaluation through recursion. In case a rule cannot be observed, its subtree will not be satisfied, as well. According to this structure, the *Policy* can either be *Mandatory* or *Optional*: if the *Policy* is *Mandatory* the drone must observe each *Rule*, otherwise it may observe them, based on its capabilities. Thus, the *Policy* can assume three distinct states: *Observed*, *Partially Observed*, and *Not Observed*.

- *Role*: one or more *Policies* characterizing the working context of a drone and defines its abilities wrt. the mission requirements. Each *Role* is uniquely identified by a Global Role Identifier (GRI) and can either be in the state of *Unassigned*, if the *Role* has not been assigned to a drone, or *Assigned*, if so. More drones can be employed to cover the same *Role* in the mission, hence they will be identifiable under the same GRI.

- *RoleSet*: group of *Roles* that satisfies mission requirements by employing multiple drones. Each *RoleSet* can assume the state of *Unassigned*, if no drone have been selected yet, or *Assigned* if each swarm component is compliant with the specified *Roles*.

Following the aforementioned Policy-based model, the Role Manager has to detect the optimal *Role* for the drone. The declarative nature of drone capabilities allows an easier detection of drone compatibility with the candidate *Role*.

For each *Rule* of a *Policy*, the Role Manager has to link *Subject* with capability and *Predicate* with the functionalities declared on the capability manifest. This operation is achievable through the employment of a marshalling mechanism. In this way, it is possible to completely abstract drone hardware at mission design and control. Thus, a *Role* is the characterisation of the ideal drone for the mission. The importance of the Role Manager not only includes the necessity to find an optimal drone for the mission, but also to configure and contextualise the candidate drone in order to perform mission tasks ahead. This also comprehends mission requirements, e.g., geofencing and restrictions in the working airspace.

The contextualisation of DCL components, depending on the mission, is important for the Device Manager. A high-level application can use a capability by requesting its access at the *Capability Access Facility* (reported as building block of the upper layer in Figure 2.6) which is managed by the Role Manager. If such capability is compatible with *Role* confinements and requirements, the access is granted. The application can then interface with the *Capability Communication Facility*, managed by the Device Manager, to use the requested and initialized resource.

In case a drone might not be able to complete the mission as expected, the MCC notifies the DOC that a swarm is no longer suitable for that mission. Nonetheless, there could be new drones available, ready to deal with that mission in a more efficient way, otherwise the mission is aborted. In the former case, the DOC suspends the mission and re-configures its *RoleSet*. The process is similar to the previous one: the DOC announces the *RoleSet* availability, awaits drones' response, and selects again a suitable group that replaces the current operating swarm for the mission. When the new swarm is ready, the DOC notifies back the MCC and the mission is resumed.

2.3.3 Telemetry

The Telemetry module aggregates and elaborates drone data in an uniform and declarative way. Information derived from the SbI, e.g., inertial measurement unit (IMU), state, and diagnostic data, converge into this module and are provided to applications through the NbI or external entities via the EbI. This continuous data stream enables the assessment of drone operations being correctly executed. Moreover, this module can also act as an emergency black box in order to analyse data related to an unexpected event, like a system failure.

2.3.4 Connection Manager

The Connection Manager is focused on connection abstraction, which decouples data exchange from the protocol stack in use, thus yielding to context independent communications. The module maintains links with remote hosts using multiple radio and networking technologies according to drone *Role* and application requirements. Concretely, the module is in charge of finding the optimal setup for the requested communication channel. A fundamental additional service is neighbourhood discovery, such as nearby drones and other relevant entities, e.g., MCC and DOC.

2.3.5 Session Manager

Leveraging the functionalities exposed by the Connection Manager, the Session Manager handles simultaneous logical links to ease context-based information exchange to/from external entities. The module provides a simple interface to open/close connections and send/receive end-to-end, broadcast and multicast information. As depicted in Figure 2.5, drone addressing is done through the use of a GDI to establish a session with it. Based on mission context and drone role, two more multicast addresses can be used, namely Logical Swarm Address (LSA) and GRI. The former is useful to broadcast information to an entire swarm, while the latter is appropriate in situations where a specific group has to be reached. Moreover, the LSA can be combined with the GRI to interact with drones that are assuming a specific role within the swarm.

2.3.6 Flight Controller

The Flight Controller provides a common set of commands that allows flight manoeuvrers, addressing hardware and software complexity. The Flight Controller has the responsibility of accepting such commands and controlling drone movements using its specific functionalities and drivers. The control logic that characterises the Flight Controller must be flexible enough in order to find and use particular drone flight capabilities, e.g., drone's trajectory expressed as a sequence of steps, to change course in mid-flight in a short time, to stop in mid-air, to do acrobatics, and to move with high accuracy in constrained environments.

2.3.7 Swarm Agent

The Swarm Agent is specifically designed to handle information regarding mutual coordination among drones forming a swarm. Such messages allow swarm cooperation in order to achieve mission tasks. Specific functionalities include flight coordination and collision avoidance that rely on a continuous exchange of the relative position and speed, on top of low-latency communication channels. The Swarm Agent entrusts the EbI and the Session Manager to synchronise with other Swarm Agents. Clearly, such feature is not available if the drone is not part of a swarm.

2.3.8 Mission Manager

The Mission Manager is focused on monitoring the drone status and on planning the sequence of tasks to be completed. This sequence of tasks can be received directly by the MCC in a structured format named *Mission Plan*, or be streamed from another drone locally coordinating the swarm. Upon schedule, the Mission Manager orchestrates DCL modules to achieve each task, disseminating the derived information to the appropriate modules. Furthermore, during the mission it is of importance to analyse the telemetry and detect anomalies or any other critical element that could suggest the activation of emergency procedures. It is also important that the Mission Manager is highly configurable and flexible, to confirm it as a general purpose endpoint. Task scheduling and operational strategies can differ among missions or be limited by environmental restrictions.

2.4 Application domains and relevant use cases

To prove the applicability of the DCL, two real-world use cases are described hereby, starting from the bootstrap procedure they share. Without loss of generality, it is hereby assumed that all the operations described in the reference scenario have been completed. At the beginning of the mission, the drone receives the mission plan by the DOC.

The mission plan is a set of information including: flight trajectories, specific role assigned to the drone, as well as its possible membership in a swarm. All the components of the mission plan are contextualised and dispatched to the reference logical modules to be initialized.

When the mission includes a single drone, the addressing scheme requires the GDI and the GRI, so that the drone will be assigned both a logical address and an information connected with its role in the mission. In case the mission requires a swarm, the addressing scheme will still require the GDI and the GRI, but the two will come along with the LSA. This is motivated by the fact that the LSA indicates the group of drones belonging to a specific logical network.

The bootstrap procedure continues with the notification of the new mission plan to the applications by the DCL through the NbI. When this bootstrap phase is over, the mission can begin.

2.4.1 Public safety

Drone technology has been applied in monitoring and containing the COVID-19 pandemic in several countries worldwide. Crowd surveillance has been successfully applied in China to detect gatherings of people. In Spain, drones equipped with speakers were used to inform people about current regulations. In particular situations, drones were equipped with multi-spectral cameras to measure body temperature en masse, or with medical supplies to safely ship to the destination [108].

In a similar use case, two main roles for drones are envisioned. The former, namely announcer, is in charge of broadcasting safety measures, e.g., observing social distancing, frequently sanitizing hands, and using masks. The latter, namely detector, employs multi-spectral cameras to detect body temperatures and crowds.

Announcers have to roam the whole area of interest avoiding collisions. To this aim, core functionalities may be extended with optimisation algorithms

implemented through the WbI, and used by applications via NbI. Moreover, a collision avoidance mechanism, relying on intra-swarm communications, is provided by the Swarm Agent. Indeed, it is assumed that each drone periodically broadcasts its Global Navigation Satellite System (GNSS) position through the EbI that relies on the Session Manager, leveraging the LSA. The application that characterises this drone class picks pre-recorded audio messages available on-board and dispatches them through the audio capability exposed by the Device Manager. Then, the SbI takes care of encoding and playing audio messages.

Detectors have to patrol the same area of interest, searching for crowds and measuring body temperatures. Even in this case, optimisation algorithms are necessary to maximise, without colliding, the covered area. Video signals, obtained from the camera capability, feed an application-level machine learning (ML) model. It is able to recognise gatherings, generating a notification that is sent to a cloud-based service monitored by authorities. At the same time, an alert is delivered to announcers to reach and warn the target audience. Both intra/inter swarm communications take place thanks to EbI, which leverages Swarm Agent and Session Manager primitives.

Throughout the mission, both classes continuously stream telemetry to the MCC for ATC activities. Telemetry messages are handled by the dedicated DCL module. Also, for each drone, Flight Controller relies on the Mission Manager, which constantly updates trajectory according to surroundings.

2.4.2 Flying base station

Among 5G & beyond applications in which drones are involved, one of the most promising is the FBS. Scientific literature has been deeply analysed and this use case has been discussed from an optimisation point of view [109, 110]. In fact, drones are warmly recommended in all those situations in which connectivity is poor or lacking, e.g., rural zones, or disaster areas. This is due to their ability to extend, restore, or, in the worst case provide, radio coverage to ground users (GUs).

In this use case, it is assumed that each drone is equipped with two radio interfaces. Each of them is dedicated to a communication link, one for UAV-GU and the other for UAV-ground base station (BS).

Upon reception of the GU traffic, packets reach the SbI, specifically the Protocol Stack Context and then are forwarded to the relay application, through the NbI. The application is in charge of processing and routing data over the service

gateway via the EbI. To this aim, specific optimisation frameworks are needed to manage radio resources, leveraging extension modules of the WbI. The same operations are applied to the inbound traffic received from the service gateway. Moreover, among the outputs of the optimisation process there is the trajectory, which is utilised by the Mission Manager. This information, that is forwarded to the Flight Controller, is crucial to guarantee optimal coverage and service quality. Based on the available intra-swarm communication technologies, drones can also be organised into multi-hop topologies to further extend the distance between UAV serving GUs and the ground BS. Furthermore, the Swarm Agent is involved to coordinate the fabric of interconnected UAVs.

3 The Internet of Drones simulator

As T/NTN communication architectures increase in complexity, it is of utmost importance to ensure that researchers have a companion tool that allows them to study and understand how these devices would operate in an integrated context. At the same time, industries need a state-of-the-art platform to observe such advancements, track their performance, and iterate over their future 6G-enabled products in a fast-paced environment. To this end, in order to enable such use cases, a comprehensive system-level communication simulation platform is needed, which offers a balance between (i) a high-level scenario configuration language, (ii) an extensible model architecture, (iii) a high-quality foundation library, and (iv) a rich set of complementary tools for data analysi..

The IoD-Sim is an open-source system-level network simulator[1], written in C++ and based on the well-known ns-3 event-based simulator [1, 2, 6, 111], extending it with over 200 C++ source and header files, 120 classes, and 34000 lines of code. It represents a significant step forward in the design and evaluation of 6G-enabled integrated T/NTNs, by extending the functionalities made for ns-3 with NTN entities, i.e., UAVs, HAPs, and satellites. Moreover, experimental 6G IRSs can be configured and attached to these mobile nodes in order to study use cases that involve mobile passive relaying. The scenario can be easily configured in a high-level language by the user, thus not requiring particular advanced coding expertise. Given its flexible and modular structure, it represents a valuable tool that paves the way for advanced end-to-end network design and evaluation for 6G T/NTNs.

In light of the foregoing, the key contributions of this chapter are hereby summarised:

- Relevant contributions in this field and other simulators are analysed in detail, while their features are compared to prove the novelty of this work.
- Since its first release [111], solely focused on the IoD and hence its name,

1 The current release of the source code (v4.0.1) along with any future releases can be retrieved at the following URL: https://github.com/telematics-lab/IoD_Sim/releases

the simulator has been carefully re-designed and thoroughly refactored as a three-layer software architecture, with both high-level configuration and reporting facilities to ease scenario design and data analysis.

- Through a high-level scenario configuration and mission design language, the simulator offers facilities that enable the design of a network infrastructure, drone and HAPs trajectory planning, satellite orbital characteristics, together with their hardware configuration, energy consumption models, and peripherals.
- A standardised channel model based on 3GPP TR 38.811 [51] has been extended from the original implementation for ns-3 [112] in order to evaluate the PHY-layer performance of the communication link between a HAP and a GEO satellite.
- A channel model expression for UAV-aided IRS-assisted communications is derived. In particular, a swarm of IRS-equipped drones is considered, in charge of enhancing the channel quality of GUs. The system adopts the orthogonal frequency division multiple access (OFDMA) scheme, which avoids interference among users. Nonetheless, the mathematical formulation still considers constructive/destructive interference patterns due to the presence of multiple IRSs. Furthermore, the IRSs are divided into patches of an arbitrary size, which can be assigned to different GUs. Based on these assumptions, a gain lowerbound expression is obtained by (i) reducing the number of degrees of freedom introduced by the controllable phase shifts, (ii) employing a mathematical approximation for the complex Gaussian product involved in the channel modelling, and (iii) imposing a fixed outage probability to cope with the inherent stochastic nature of the channel.
- Mobile IRSs can be configured and attached to UAVs and HAPs and, among the manifold configuration parameters, the simulator provides the possibility to dynamically change the number and the size of the patches and for how long a certain GU is served by a specific patch. Moreover, thanks to the fact that IoD-Sim is based on ns-3, it is possible to employ an arbitrary communication stack on top of the PHY layer model.

The main notation adopted in this chapter is described hereby. Boldface lower and capital case letters refer to vectors and matrices, respectively; $j = \sqrt{-1}$ is

the imaginary unit; atan2(x) denotes the four-quadrant arctangent of a real number x; $\mathbf{x}^{\mathrm{T}}$ is the transpose of a generic vector $\mathbf{x}$; $x \sim \mathcal{CN}(\mu, \sigma^2)$ define a circularly symmetric complex Gaussian distribution x with mean μ and variance σ^2; diag($\mathbf{x}$) represents a diagonal matrix whose diagonal is given by a vector $\mathbf{x}$.

3.1 Related work

To improve and speed up both the design and the prototyping phases of T/NTN systems, and hence achieve the stated vision, simulations are widely conceived as a useful aid. Even though simulating drones, HAPs, and satellites, together with 6G-enabled devices, is a challenging task, it has been dealt with by many contributions so far [112–127].

Most notably, ns-3 [128] with its 5G LENA module [129], OMNeT++ [130], and Vienna 5G [131] are well-known for their solid foundation in simulating terrestrial networks, based on the latest 3GPP and International Telecommunication Union Radiocommunication Sector (ITU-R) standards on 5G. Furthermore, there is ongoing work to enable 6G communications in ns-3 with Terasim [127]. As for satellites, there is an ongoing effort to extend ns-3 through its NTN module [112] and Satellite ns-3 [120], although these modules lack full end-to-end simulation for the former and the specific focus on digital video broadcasting (DVB) for the latter. However, it is still unclear whether satellite-to-HAP communication is feasible and, if so, how it can be realised. Finally, in order to design and test IRSs, WiThRay [121], SimRIS [122], and a Vienna 5G module [123] allow their modelling, phase shift optimisation, and the evaluation of their fading loss.

Regarding drone-assisted communications, these works approach these scenarios from two different points of view. The first focuses on the dynamics of the flight, thus including mechanical energy and kinetics; they employ Robot Operating System (ROS) [132] and Gazebo [133] as base platforms [116, 118]. The second, instead, focuses on accurate drone networking simulations [113–115, 117, 119], mainly based on ns-3 [128] and OMNeT++ [130], in which UAVs are envisioned as nodes exchanging data at certain frequencies using well-known protocols belonging to wireless networks, which can either be cellular or Wi-Fi. The contribution presented in [113] models UAVs and discusses their functionalities and possible applications. In particular, the proposal introduces FlyNetSim, a software that aims at simulating not only flight operations but also networking communication primitives and principles. The simulator can work with a group

Table 3.1: Summary of the comparison of the available solutions.

	[113]	[114]	[115]	[116]	[117]	[118]	[119]	IoD-Sim
Open-source code				✓		✓		✓
Modularity	✓	✓	✓					✓
Scalability		✓	✓					✓
Visual scenario configuration	✓	✓				✓		✓
Network simulations		✓	✓		✓	✓	✓	✓
Support multiple networking standards		✓			✓			✓
Multi-stack protocols support	✓							✓
Graphically-assisted trajectory design	✓	✓						✓
Aerodynamics simulations	✓	✓		✓				
Power consumption models					✓			✓
Hardware in the Loop support	✓	✓						
High-level application development support		✓		✓	✓		✓	✓
Ready-to-use IoD applications								✓
Machine-readable results		✓	✓	✓				✓
Human-readable results	✓	✓	✓	✓			✓	✓

of drones operating together in a reference ecosystem. The most interesting functionalities are: (i) UAV control over Wi-Fi, (ii) multi-network communications, (iii) Device-to-Device communications for swarms, and (iv) IoT scenarios and data streaming. In [114], instead, it is proposed CUSCUS, a simulation architecture for control systems in the context of drones' networks. The proposal is able to simulate the mechanisms for the control of drones operations and it is claimed to be highly flexible and scalable. The proposed simulator leverages the ns-3 capabilities to work with virtual interfaces simulating real-time systems, eventually composed by swarms. The contribution presented in [115] describes AVENS, which is a hybrid network simulation framework specifically designed to evaluate the performance of intelligent aerial vehicles. Here, drones are communicating using some of the most well-known communication protocols for flying ad-hoc networks (FANETs). Differently from other contributions, AVENS is focused on modelling realistic flight conditions. On top of that, it uses a layered architecture that acts as an interpreter and code generator, namely LARISSA, thanks to specified simulation parameters and settings. All the results are obtained by the integration and interoperability with the OMNeT++ simulator. The proposal in [116] is a simulation framework for UAS traffic management. It leverages both ROS and Gazebo to implement high-level flight services. The simulator can be used for prototyping missions and controlling both rotary and fixed-wing drones flying in the same environment. The work presented in [117] discusses a Java-based simulation framework for FANET networks and their applications. In particular, it models the coverage area of each device in the scenario. At the same time, it considers a mobility model for ground entities, i.e., humans in the operating area. Drones' characterisation is herein discussed in terms of limited autonomy and battery recharging needs. To achieve this aim, an energy consumption model has been included to evaluate the footprint associated with the flight of a drone. For the sake of completeness, it must be said that this work neglects the contributions due to collision issues and consequent behaviours. CORNET 2.0 [118] is a middleware to simulate robots in general, both in physical and networking contexts. It reaches the aim of designing a path planning solution that is simulated by Gazebo and Mininet-WiFi. The work presented in [119] proposes a discrete-time, event-based, co-simulation scheme for networks composed by multiple drones, also configured in swarms. The simulator can carry out both flight and network simulations. This solution is of interest because there is an intrinsic codependency between the flight status

and the networking operations carried out by each drone in the scenario. This contribution is of relevance because it is claimed to guarantee reliability and real-time availability thanks to the possible integration of existing simulators. This work also claims that other available simulators do not implement realistic and reliable mobility models for drones. A comparison among the main characteristics and features of the aforementioned contributions is provided in Table 3.1. Specifically, only the latter [119] shows some similarities with IoD-Sim. For example, both of them operate as discrete-time and event-driven simulators. Nevertheless, it is worth noting that the discrete-time operating mode of this work is motivated by the adoption of ns-3 [128] as a core network simulator. Another aspect is related to the synthetic trajectories that are implemented in IoD-Sim, that are described by closed-form mathematical expressions. Hence, in case the network simulator is substituted, the mobility models provided by IoD-Sim could be used even with continuous-time simulators.

Regarding IRS-aided communications, the scientific literature [121–126] do not address the lack of platforms able to simulate IRS-equipped UAVs, but solely proposes solutions focused on IRSs-aided communication systems. In this regard, [121] introduces WiThRay, a versatile framework which models the mmWave channel response in 3D environments by employing ray tracing. It allows to deploy and configure multiple BSs and IRSs, which serve mobile users. In [122], an open-source MATLAB-based simulator is developed, namely SimRIS, which leverages a channel model for mmWave frequencies, applicable in various indoor and outdoor environments. The simulator provides a simple GUI which gives the possibility to set up (i) the operating frequency, (ii) the terminal locations, and (iii) the number of IRS elements. [124] proposes a simulation framework, based on ns-3, to simulate IRS and amplify-and-forward (AF) systems. The end-to-end communication is implemented by employing the standardised 3GPP TR 38.901 channel model [134] and the 5G NR protocol stack. This contribution aims to (i) demonstrate whether IRS/AF nodes can be used to relay network traffic and (ii) dimension the number of IRS/AF nodes wrt. the number of users. [125] analyses the system-level simulation results of urban scenarios in which multiple IRSs are deployed in presence of a 5G cellular network. It emerges that the IRS performance strongly depends on its size and the operating frequency. In particular, this manuscript investigates the benefits brought by IRSs, in mid (C-band) and high (mmWave) frequency bands, by deriving outdoor and indoor coverage and per-resource block rate. Similarly to previous works, [126] introduces a

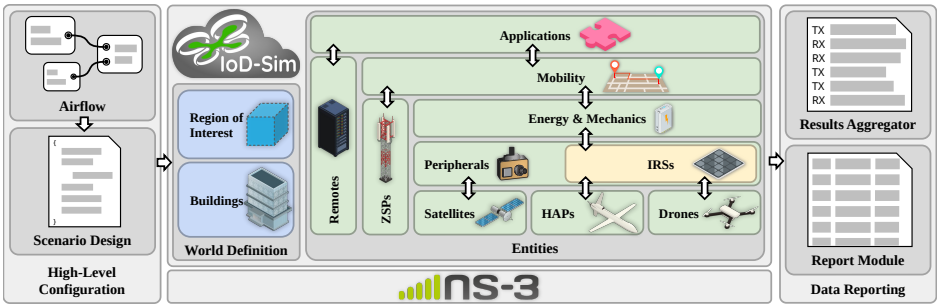


Figure 3.1: The IoD-Sim architecture.

system-level simulation platform implemented in C++ for 5G systems, which includes network topology, antenna pattern, large/small scale channel models, and many performance indicators. Specifically, this paper investigates the case in which the LoS propagation is dominant under far-field conditions. Moreover, the performance of phase quantization are also discussed and analysed. Besides, [123] implements an extension for the Vienna 5G simulator, which includes IRS modelling, its phase shifts optimisation, and large- and small-scale fading. All the contributions discussed above consider each surface associated only to a specific user that, on one hand, simplifies the mathematical modelling and the software implementation, but, on the other hand, limits the achievable system performance. Furthermore, the employment of aerial mobile IRSs, enabled by drones, is not taken into account, even if it would represent a big advantage in terms of flexibility and increase the scenario complexity.

3.2 Architectural overview

The overall architecture of the proposed simulator is depicted in Figure 3.1. This diagram frames the complexity and clarifies the organisation of the main software layers, each providing peculiar functionalities that are depicted as blocks. This time, differently from the original publication, the architecture is presented following a pipeline model to improve clarity. The end user would interact with the *high-level configuration* to design and configure a scenario, which is then processed by the *core*, and finally results can be evaluated through the *data reporting* facilities. From left to right, bottom-up, such architecture is described hereby.

3.2.1 High-level configuration

A configuration abstraction reduces the effort to set up complex scenarios by hiding the complexity of the integrations made in IoD-Sim and the definition of ns-3 scenarios, which requires C++ programming skills. To this end, scenarios can be designed and configured through a GUI program named *Airflow*. As illustrated in Figure 3.1, this tool leverages a domain-specific visual programming language to configure the entities and link them together as graphical blocks and connections. Once configured, the scenario is exported in JSON-encoded files. These configuration files, generated via Airflow or written manually, are then parsed by the simulator. This approach eases the learning curve of all the features provided by the platform, while keeping the configuration parameters in a data format that has an acceptable trade-off between human comprehension and ease of machine parsing.

3.2.2 Core

The central block, i.e. the *core*, implements T/NTN-related simulation facilities and it is organised into two main subgroups. The *World Definition* is motivated by the fact that a realistic network simulation must be modelled, taking into account cyber-physical aspects. Hence, IoD-Sim allows to simulate the physical space in which the simulation takes place, from the coordinate system to *Buildings* and *regions of interest (RoIs)*. Furthermore, *Entities* details all the aspects related to drones, zone service providers (ZSPs), and remote hosts, spanning from their peculiar characteristics, i.e., peripherals and mechanics for UAVs, to general ones, such as applications.

World definition

The simulator offers the possibility to define a simple 3D Cartesian space or a geographic one, enabling the choice to study simpler local networks or to reproduce realistic global ones, respectively. If the latter is chosen, IoD-Sim automatically converts the system of reference to a geocentric Cartesian one or to a Mercator projection as needed. Given the selected coordinate space, buildings and RoIs can be defined. While the former are key for realistic experimentation in smart cities, the latter represent imaginary 3D boxes that enables mobile entities to change their functions according to their current location.

Entities

Drones and HAPs are a specialisation of ns-3's `ns3::Node` object [128] adding support to crucial characteristics of these non-terrestrial entities, i.e., mechanical features, on-board peripherals, and applications. The essential properties that distinguish these entities are their mass, aircraft type, and flight altitude, along with their rotor disk area, drag coefficient of the rotor blades (if any,) and battery model. According to these details, a power consumption model can be configured to ensure a proper characterisation of the energy depletion curve and estimate their battery residual. Furthermore, along with the common fast and slow-fading phenomena of the Earth's troposphere, the altitude of these flying objects impacts on the channel model due to atmospheric absorption and scintillation effects [51]. UAVs and HAPs can also be characterised with a mobility model. Among the ones already provided by ns-3, IoD-Sim offers specific models that simulate a curvilinear trajectory based on points of interest (PoIs). The curve points, generated through an enhanced Bézier equation, are followed by the entities in a constant acceleration or with a given parametric speed equation.

GEO satellites represent a preliminary effort in IoD-Sim to embrace satellite communications, especially in the mmWave spectrum. Currently, it is possible to define satellite position through its geographic coordinates and distance from the Earth's surface, giving the possibility to evaluate fundamental link-level performance metrics, such as the SINR.

Each entity can be equipped with peripherals of different types, i.e., (i) a storage one that acts as a temporary buffer in case of loss of signal, (ii) a generic input peripheral that captures data, and (iii) an IRS for relaying use cases. As the input peripheral can be interleaved with the storage one to simulate buffered communications, the IRSs enable these entities to passively improve link efficiency between the ground users and the designed network infrastructure. Moreover, they can be logically subdivided into patches, each of which has a scheduling plan to dynamically serve certain nodes of interest or to just configure a fixed reflection angle.

The terrestrial infrastructure is composed of ZSPs that represent enhanced BSs with local edge services, e.g., ATC and weather forecast. These nodes can be interconnected with their core network and UAVs, which are key-enablers for an integrated T/NTN [22]. Behind the core network of choice, a backbone network simulates the presence of remote cloud services. Together with UAVs and HAPs, these entities can be configured with applications that unlock a

variety of use cases, spanning from telemetry reporting to multi-stack relaying, video recording, and streaming.

3.2.3 Data reporting

According to the configuration, data are extracted, aggregated, and summarised in a dedicated directory whose name includes the time of execution of the scenario and its name. Among the traditional text and PCAP-based trace files of ns-3, an extensible markup language (XML) summary is generated, which reports relevant data of the simulation, e.g., data traffic, *entities*' trajectories, their telemetry, and application-specific KPIs. The file formats have been chosen to ease data pre-processing and analysis with other tools and languages, e.g., Python and MATLAB, for which some reference examples are provided in the source code.

A *Report Module* guarantees the readability of simulation results in a clear XML schema. This module, together with the *Results Aggregator*, eases data processing with third-party tools.

3.3 Underlying Platform

The set of third-party libraries used as a foundation for the *core* of IoD-Sim is primarily composed by the GNU Scientific Library and ns-3. The former is a scientific computing framework that offer access to a wide variety of implementations in complex numbers, linear algebra, data analysis, and interpolation. The latter is a well-known and solid system-level discrete-time network simulator. Started in 2006, it is a collection of different C++ and Python objects that implements several aspects of networking elements. The fundamental building block of ns-3 is `ns3::Node`, an abstract object which represents a generic host in a network. It can be aggregated with other objects and models, e.g., the common Transmission Control Protocol (TCP)/Internet Protocol (IP) stack over Ethernet, to simulate networking behaviour. Other interesting features in ns-3 are (i) `ns3::Channel`, which simulates the communication channel between `ns3::Node` objects, (ii) `ns3::NetDevice`, which represents the node networking interface, and (iii) `ns3::Application`, which sits on top of the protocol stack to produce or consume high-level information.

Furthermore, a `ns3::Node` can be aggregated with *mobility* and *energy con-*

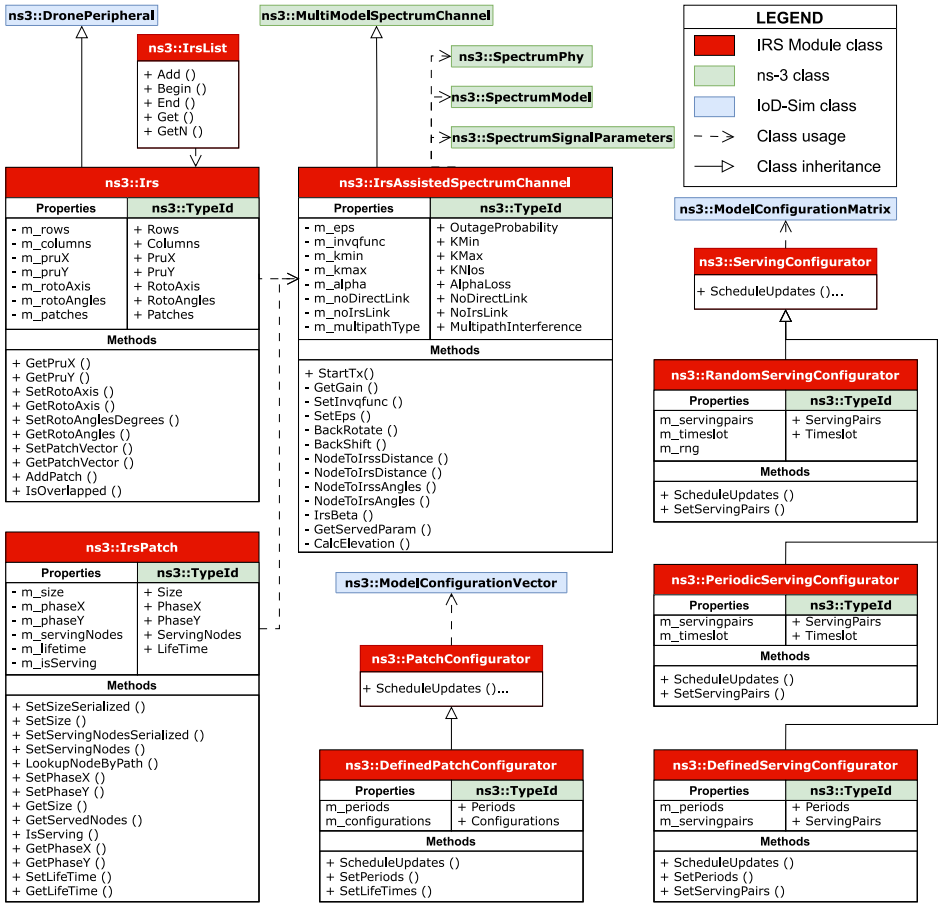


Figure 3.2: Class diagram of the IRS extension implemented in IoD-Sim.

sumption models. This possibility is not limited to those models, since the support can be extended to any other model that adds new features beyond basic networking. To this end, nodes have the potential to move in space and, hence, drain current from a `ns3::EnergySource`. Besides, traces and probes allow to track and record simulation data in log files that are typically encoded in textual ASCII or PCAP data formats.

3.4 Core of IoD-Sim

A simulation scenario requires the definition of a simulated world, described by *RoIs* and *buildings*. In such world, the *entities* are simulated in a network topology defined by a set of communication models. Each mobile entity, i.e. a UAV or a HAP, is characterised by a mission plan defined by a set of PoIs, which in turn describes a curvilinear trajectory. Furthermore, a mobile entity can be equipped with an energy consumption model, which relies on a set of mechanical properties and a set of peripherals. *Entities* in general can host one or more communication stacks and applications. While mobile entities and ZSPs are connected together according to the configuration of the T/NTN infrastructure, remotes are reachable through a backbone that simulates Internet behaviour. All these blocks are configurable through an abstraction interface focused on interpreting a high-level description of the scenario encoded in JSON format.

3.4.1 World definition

IoD-Sim offers the possibility to model the world and the environment according to the T/NTN use case of interest. The virtual world in IoD-Sim is a theoretically infinite space. Along the entities, the space can be filled with RoIs and buildings.

Coordinate system

Different coordinate systems ensure that the simulation is accurate and relies on opportunistic simplifications when available. A point in space, defined as $\mathbf{p} = [x, y, z]^\mathsf{T}$, $\mathbf{p} \in \mathbb{R}^3$, has three coordinates that can be defined in meters in a generic Cartesian space. Moreover, geographical coordinates are defined, in order to take into account Earth's curvature and study complex scenarios with satellites. In this case, λ and φ axes are expressed in radians representing latitude

and longitude, respectively, and the z axis in meters, which indicates the altitude of the point relative to Earth's surface.

When needed, a geographical point can be projected over an auxiliary Cartesian space, known as the EPSG:3857 WGS84/Pseudo-Mercator projection [135], which is defined as

$$\overline{\mathbf{p}} = [\overline{\lambda}, \overline{\varphi}, \overline{z}]^{\mathsf{T}}, \tag{3.1}$$

$$\overline{\lambda} = \frac{2^{\alpha}}{2\pi}(\lambda + \pi), \tag{3.2}$$

$$\overline{\varphi} = \frac{2^{\beta}}{2\pi}\left(\pi - \ln\left(\tan\left(\frac{\pi}{4} + \frac{\varphi}{2}\right)\right)\right), \tag{3.3}$$

$$\overline{z} = z, \tag{3.4}$$

where $\alpha = 25.059$ and $\beta = 24.665$. The latter two constants are used to normalise the Cartesian space's unit of measurement to meters.

To evaluate the distance between two geographical points, the geocentric is used as another auxiliary Cartesian system. This reference system has its point of origin at the centre of the Earth. To this end, the geographical coordinates are transformed using the WGS84 ellipsoid. By computing the polar radius as

$$r = \frac{a}{\sqrt{1 - e^2 \sin^2(\overline{\lambda})}}, \tag{3.5}$$

where $a = 6378137$ and $e = 0.0818191908426215$ are the Earth's semi-major axis and its eccentricity, respectively. Then, the corresponding points in geocentric coordinates are expressed as:

$$\lambda' = (r + \overline{z}) \cos(\overline{\lambda}) \cos(\overline{\varphi}), \tag{3.6}$$

$$\varphi' = (r + \overline{z}) \cos(\overline{\lambda}) \sin(\overline{\varphi}), \tag{3.7}$$

$$z' = ((1 - e^2)r + \overline{z}) \sin(\overline{\lambda}). \tag{3.8}$$

In this way, the distance between two geocentric points, e.g. $\mathbf{p'}^{\mathrm{a}}$ and $\mathbf{p'}^{\mathrm{b}}$, can be expressed as $d'^{\mathrm{ab}} = \|\mathbf{p'}^{\mathrm{b}} - \mathbf{p'}^{\mathrm{a}}\|$.

Buildings

The virtual world can be enriched with obstacles, i.e., *buildings*. They are used to represent urban scenarios, thus making simulations that are particularly suitable for research in smart cities. IoD-Sim provides an abstraction layer to configure and place buildings in the virtual world, relying on `ns3::BuildingsHelper` and `ns3::Building` objects. A `ns3::Building` is a collision-less 3D object with the following properties:

- `boundaries`, which defines the box dimension in the space. They are defined as an array of two points, e.g. $\mathbf{p}^{a}$ and $\mathbf{p}^{b}$, organised as $[p^{a,x}, p^{b,x}, p^{a,y}, p^{b,y}, p^{a,z}, p^{b,z}]$.
- `type` of building, which can be either `commercial`, `residential`, or `office`.
- type of `walls` material, which can be `wood`, `concreteWithWindows`, `concreteWithoutWindows`, and `stoneBlocks`.
- number of `floors`.
- number of `rooms` along the x and y axis, per floor. The rooms are placed in a grid position.

Such a feature is important for communication fading, which varies according to the characteristics of each building.

regions of interest

A `ns3::InterestRegion` represents a 3D box defined by a pair of points, similarly to how buildings are placed. Throughout the simulation, it is possible to retrieve and to update its current set of coordinates with `GetCoordinates()` and `SetCoordinates()` methods, respectively.

The whole set of RoIs is managed by a `ns3::InterestRegionContainer`, which provides a (i) `Create()` method to generate and index RoIs, (ii) `GetN()` to report the number of created regions, (iii) `GetRoi()` to retrieve the i^{th}, and (iv) `Begin()` and `End()` iterators to traverse the entire container. Moreover, the `InterestRegionContainer::IsInRegions()` method check if an entity is present in the region, enabling events to be triggered at any time during

Name	Measurement unit
Mass	kg
Rotor disk area	m^2
Drag coefficient	*(dimensionless)*
Weight force	N
Air density	kg/m^3

Table 3.2: `ns3::Drone` properties in IoD-Sim.

the simulation. For instance, UAVs and HAPs' peripherals can be enabled and disabled, leading to an optimisation of their power consumption.

3.4.2 Drones and high altitude platforms

IoD-Sim provides `ns3::Node` derivatives to consider the characteristics of key actors commonly found in a NTN simulation. The `ns3::Drone` class characterises a multi-copter UAV or a fixed-wing HAP and it is registered as a new `TypeId` in ns-3, along with its mechanical properties, shown in Table 3.2. While the first three properties can be defined by the user, the last two are a direct consequence of the given characterisation. `ns3::Drone` properties can be set by means of ns-3 attributes or by its public object interface. Its mass can be updated at any time by means of `SetMass()`. Upon update, the drone weight force is also updated in cascade by multiplying the new mass with the constant gravity acceleration. The rotor disk area and its drag coefficient can be updated in the same manner by means of `SetArea()` and `SetDragCoefficient()` methods, respectively. Furthermore, `ns3::Drone` properties can always be read any time during the simulation through ns-3 attributes and object getters, such as `GetMass()`, `GetWeight()`, `GetArea()`, and `GetDragCoefficient()`. Drones can be grouped together in `ns3::DroneContainer` and can be statically referenced by their unique identifier in the simulation through `ns3::DroneList`. According to the peculiar workflow of ns-3, to properly instantiate a `ns3::Drone` object, a `ns3::DroneContainer` is needed. The creation process consists of a call to the `ns3::Object::CreateObject<T>` function, where `T` is replaced with `Drone`. In order to ensure full compatibility with all ns-3 methods involving `ns3::Node` or `ns3::NodeContainer` classes, a dedicated mechanism has been developed. Every `ns3::Drone` goes through a static cast procedure,

i.e. `ns3::StaticCast`, which generates a `ns3::Node` object that is pushed into a `ns3::NodeContainer`. In this way, for each drone, two smart pointers refer to the same memory location but cast to the two required types. Besides, the `ns3::DroneContainer` class provides a specific iterator, together with two further methods which return the number of instantiated drones and a smart pointer to each. It is worth mentioning that only drones must use a `ns3::DroneContainer`, while ZSPs, together with other entities, must still be modelled as `ns3::Node` objects.

Peripherals

A UAV is outfitted with various peripherals designed to enhance its functionality. These peripherals encompass a broad array of devices, which are implemented in IoD-Sim via specific classes. The `ns3::DronePeripheral` object serves as a general-purpose on-board peripheral, characterised by the following properties:

- *Peripheral state* – governed by a simple finite state machine, which can be set to `ON`, `OFF`, or `IDLE`, paving the way for energy-aware algorithms.
- *Power consumption* – how much instantaneous power is required by the peripheral, expressed in Watts, for each state.
- *Reference RoIs* – where the peripheral should be operating. This is extremely useful for modelling certain peripherals and missions that depend on particular regions in space. For instance, a photo camera can be used and activated only when the drone is in the RoI, thus leading to an optimised use of power, storage, and data. If this parameter is not defined, the reference peripheral will be active over time.

`ns3::DronePeripheral` has been specialised in two subclasses. `ns3::StoragePeripheral` represents a generic storage device characterised by an attribute describing the initial amount of memory, which can be traced at runtime to record the space left. Device total capacity can be queried through `GetCapacity()` method. If a drone peripheral, e.g., a camera or any other sensor, wants to interact with the storage, it is possible to request space by specifying the amount of data through `Alloc()`. The inverse can be done with `Free()`. These operations can fail if there is no memory left or there are no data

Peripheral	Attribute	Description
Drone	PowerConsumption	Power consumption of the peripheral in J/s.
Storage	Capacity	The capacity of the disk in bit.
	DataRate	The acquisition data rate of the peripheral in bit.
Input	InitialRemainingCapacity	The starting remaining capacity in bit.
	DataAcquisitionTimeInterval	The time interval occurring between any data acquisition.
	HasStorage	Acquired data are offloaded to the StoragePeripheral.

Table 3.3: Drone peripherals properties.

to be freed, respectively. For this reason, a boolean value is returned by these methods to indicate if the requested operation was executed or not.

The ns3::InputPeripheral represents a generic input device, defined by an acquisition DataRate that remains constant throughout a specified DataAcquisitionTimeInterval. Once created, installed on a drone, and attached to a particular storage peripheral with the Install() method, the storage peripheral of reference can be changed with SetStorage(). If the peripheral is ON, AcquireData() simulates data acquisition at the given DataRate.

These two mentioned peripherals are closely linked together, as an ns3::InputPeripheral can transfer acquired data to an ns3::StoragePeripheral via a boolean attribute. However, this connection between input and storage is not obligatory. In real-world applications, an ns3::InputPeripheral can also transmit data directly to a processing unit or a remote host, bypassing the need for permanent storage. Table 3.3 provides a comprehensive list of the attributes of these classes. It is also important to note that all peripherals maintain a reference to the drone to which they are attached.

Moreover, for each ns3::Drone, a ns3::DronePeripheralContainer object is created to manage all its peripherals. This container is responsible for the creation of peripherals and, through the ns3::DronePeripheralContainer::InstallAll() method, sets the correct references to the host drone, and, eventually, to the target ns3::StoragePeripheral.

Mechanics and Energy Consumption

ns-3 offers energy-related simulation models through the definition of an abstract class `ns3::EnergySource`, which includes consumption, harvesting, and monitoring evaluations. While there is no dedicated energy source model specifically designed for drones, the `ns3::LiIonEnergySource` class is sufficiently versatile to be employed for simulation purposes [136, 137]. The `ns3::DeviceEnergyModel` class describes the `ns3::NetDevice` energy consumption by means of the drawn current. The installation procedure is eased by the helper class `ns3::DeviceEnergyModelHelper`, which employs the `Install()` method that links a `ns3::EnergySource` to a `ns3::NetDevice`.

When the battery object is initialized, it schedules an `ns3::Event`, which calls `ns3::EnergySource::CalculateTotalCurrent()`. This function retrieves the current drawn of every device associated with the `ns3::EnergySource`, by calling `ns3::DeviceEnergyModel::GetCurrentA()`. Subsequently, the energy consumption value is calculated and subtracted from the remaining one. Finally, the `ns3::Event` reschedules itself.

In order to accurately simulate the power consumption of a UAV, a specialisation of `ns3::DeviceEnergyModel` has been developed, i.e., `ns3::DroneEnergyModel`, is developed along with the helper class `ns3::DroneEnergyModelHelper`. Given a simulation duration T, the model splits it into $k = 1, \ldots, K$ equal discrete intervals. The power consumption model of the drone flying with a velocity $\mathbf{v}^{\mathrm{U}} = [v^{\mathrm{U,x}}, v^{\mathrm{U,y}}, v^{\mathrm{U,z}}]^{\mathsf{T}}$ is the following [138]:

$$P^{\mathrm{U}} = P^{\mathrm{l}} + P^{\mathrm{z}} + P^{\mathrm{d}}, \tag{3.9}$$

where

$$P^{\mathrm{l}} = \frac{W^2}{\sqrt{2}\rho A\sqrt{\Omega + \sqrt{\Omega^2 + 4(v^{\mathrm{U,h}})^4}}}, \tag{3.10}$$

$$\Omega = \left\| \left[v^{\mathrm{U,x}}, v^{\mathrm{U,y}} \right]^{\mathsf{T}} \right\|^2, \tag{3.11}$$

$$P^{\mathrm{z}} = W v^{\mathrm{U,z}}, \tag{3.12}$$

$$P^{\mathrm{d}} = \frac{1}{8} C^{\mathrm{D0}} \rho A \left\| \left[v^{\mathrm{U,x}}, v^{\mathrm{U,y}} \right]^{\mathsf{T}} \right\|^3, \tag{3.13}$$

$W = mg$, with m the mass of the drone and g the gravitational acceleration. Moreover, ρ is the air density, A is the total rotor disk area, C^{D0} is the profile drag coefficient depending on the geometry of the rotor blades, and $v^{U,h} = \sqrt{\frac{W}{2\rho A}}$ evaluates the power required for hovering operations. Such energy model can be aggregated to the object identifying the drone by means of the `ns3::DroneEnergyModelHelper`, thus evaluating its mechanical energy consumption. Such mechanical power consumption model is implemented in the method `ns3::DroneEnergyModel::GetPower()`. Similarly, the method `ns3::DroneEnergyModel::GetPeripheralsPowerConsumption()` returns the cumulative power consumption of all peripherals on board. The `ns3::DroneEnergyModel` object, registered as a new `ns3::TypeId` with no attributes, implements `ns3::DoGetCurrentA()` inherited from `ns3::DeviceEnergyModel`. Such method returns the total drawn current related to both mechanics and peripherals, in addition to networking operations. The energy model can be aggregated to a drone by means of `DroneEnergyModelHelper`, which provides an `Install()` method that aggregates it to `ns3::Drone`. It is worth specifying that `ns3::DroneEnergyModelHelper` implements the installation procedure in a different manner with respect to its parent, i.e., `ns3::DeviceEnergyModelHelper`. In fact, the `ns3::DroneEnergyModelHelper::Install()` method links a `ns3::EnergySource` to a `ns3::Drone` instead of a `ns3::NetDevice`. This aspect distinguishes the aim of IoD-Sim from the ns-3 one: to simulate all the relevant aspects of the drone, beyond the networking perspective. This justifies the implementation divergence from the ns-3 main goals. During the simulation, it is possible that the drone runs out of energy. To this end, the event is propagated through the execution of `HandleEnergyDepletion()` of the energy model, for which the time of depletion is logged for successive data analysis.

3.4.3 Zone service providers, remotes, and satellites

Apart from drones and HAPs, the simulator provides *ZSPs* [22], *remotes* and *GEO satellites*. ZSPs are smart BSs, modelled as `ns3::Node` objects and equipped with multiple `ns3::NetDevice`, which provide multi-protocol radio access, thus enabling communications between drones, HAPs, satellites, and the rest of the Internet. Typically, they are configured as ground entities that maintain a constant position in time by means of `ns3::ConstantPositionMobility-`

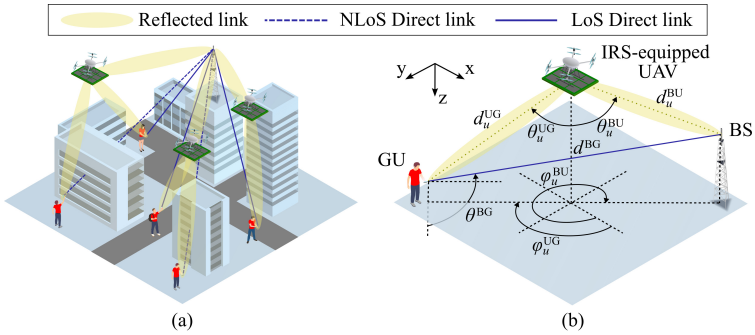


Figure 3.3: Overview of (a) the reference scenario and (b) the channel model geometry.

`Model`. *Remotes*, instead, are `ns3::Node` objects with no mobility model and only rely on installed applications which provide remote services to consumers. Remotes and ZSPs are interconnected through a backbone, simplified as a carrier sense multiple access (CSMA)-based bus network, that represents the Internet. This architecture allows service provisioning on different classes of nodes, employing remotes in the case of applications with high-computational costs, e.g., multimedia data processing, and ZSPs in the case of low-latency requirements, e.g., traffic management. Finally, *GEO satellites* are implemented along the same line of the previous two, as they are instances of `ns3::Node` with a `ns3::GeographicConstantPositionMobilityModel`. Such mobility model helps to set the position of the satellite through the definition of geographical coordinates, expressed in radians, and the altitude in meters.

3.4.4 Intelligent Reflective Surfaces

IoD-Sim paves the way for the design of missions that envision U IRS-equipped UAVs[2], flying at speed $v_u \in \mathbb{R}$, $u = 1, \ldots, U$, that improve the channel quality between a set of G GUs and the BS through a proper signal reflection, as illustrated in Figure 3.3. In this Section, for the sake of notations, the discussion considers the communication of a GU, in a given subcarrier centred in f Hz, at a certain time instant[3]. The positions of the drones, the GU, and the BS are denoted as $\mathbf{q}_u^{\mathrm{U}} = [x_u^{\mathrm{U}}, y_u^{\mathrm{U}}, z_u^{\mathrm{U}}]^{\mathsf{T}} \in \mathbb{R}^3$, $\mathbf{q}^{\mathrm{G}} = [x^{\mathrm{G}}, y^{\mathrm{G}}, z^{\mathrm{G}}]^{\mathsf{T}} \in \mathbb{R}^3$, and $\mathbf{q}^{\mathrm{B}} = [x^{\mathrm{B}}, y^{\mathrm{B}}, z^{\mathrm{B}}]^{\mathsf{T}} \in \mathbb{R}^3$.

2 The main notation used in this Section and Section 3.4.5 has been summarised in Table 3.4.
3 Assuming the kinetics is known to the BS and hence the Doppler effect is comprensated.

Symbol	Description	Symbol	Description
K	Mission duration.	d^{BG}	BS-GU distance.
U	Number of UAVs.	d_u^{UG}	UAV-GU distance.
G	Number of GUs.	d_u^{BU}	BS-UAV distance.
M^{R}	PRUs as patch rows.	g^{BG}	GU-BS direct link gain.
M^{C}	PRUs as patch columns.	β^{BG}	BS-GU power gain at 1 m.
$\mathbf{q}^{\mathrm{BS}}$	Location of the BS.	α	BS-GU link path loss exponent.
$\mathbf{q}^{\mathrm{G}}$	Location of the GUs.	κ^{BG}	K-factor for BS-GU link.
$\mathbf{q}_u^{\mathrm{U}}$	u-th UAV location.	Ω^{BG}	BS-GU link average power.
v_u	u-th UAV speed.	$\mathbf{g}_{u,p}^{\mathrm{UG}}$	Patch-GU channel gain.
$\phi_{u,p,m}$	m-th PRU phase shift.	$\mathbf{g}_{u,p}^{\mathrm{BU}}$	Patch-BS channel gain.
f	The carrier frequency.	$\mathbf{\Phi}_{u,p}$	Phase shift matrix.
w	PRU area.	P	Number of IRS patches.
κ_u^{UG}	K-factor for UAV-GU link.	κ_u^{BU}	K-factor for BS-UAV link.

Table 3.4: Main notation adopted for the description of IRS-assisted communications.

Accordingly, the far-field distances d_u^{UG}, d_u^{BU}, and d^{BG} are defined as $d^{\mathrm{ab}} = \|\mathbf{q}^{\mathrm{a}} - \mathbf{q}^{\mathrm{b}}\|$, with a, b $\in \{U, G, B\}$.

IRSs are composed by $N = N^{\mathrm{R}} \times N^{\mathrm{C}}$ PRUs, having the size $w = d^{\mathrm{X}} \times d^{\mathrm{Y}}$ m^2, with $d^{\mathrm{X}} = d^{\mathrm{Y}} \triangleq d$ being the length of the element sides. The midpoint of each PRU, wrt. the centre of the IRS, is $d\left[(i - \frac{1}{2}), (i' - \frac{1}{2})\right]^{\mathsf{T}}$ with $i = 1 - \frac{N^{\mathrm{C}}}{2}, \ldots, \frac{N^{\mathrm{C}}}{2}$, $i' = 1 - \frac{N^{\mathrm{R}}}{2}, \ldots, \frac{N^{\mathrm{R}}}{2}$. The PRUs are grouped into P patches of $M = M^{\mathrm{R}} \times M^{\mathrm{C}}$ elements, each one indexed as $m = 1, \ldots, M$. Moreover, each patch reflects the incident signal according to a phase shift matrix $\mathbf{\Phi}_{u,p} \in \mathbb{C}^{M \times M}$, with $p = 1, \ldots, P$, defined as

$$\mathbf{\Phi}_{u,p} = \mathrm{diag}\left(e^{j\phi_{u,p,1}}, \ldots, e^{j\phi_{u,p,m}}, \ldots, e^{j\phi_{u,p,M}}\right), \tag{3.14}$$

where $\phi_{u,p,m} \in [-\pi, \pi)$. It is worth specifying that for ease of readability, all the IRSs have the same number and size of patches but the model is straightforward extensible.

Finally, define $\{\theta_u^{\mathrm{UG}}, \theta_u^{\mathrm{BU}}\}$ and $\{\varphi_u^{\mathrm{UG}}, \varphi_u^{\mathrm{BU}}\}$ as the inclination and azimuth angles between the centre of the IRS and the GU/BS as $\theta^{\mathrm{ab}} = \mathrm{acos}\left(\frac{z^{\mathrm{a}} - z^{\mathrm{b}}}{d^{\mathrm{ab}}}\right)$ and $\varphi^{\mathrm{ab}} =$

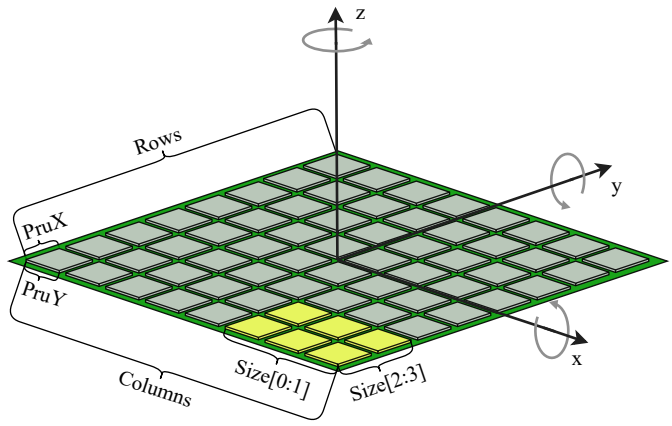


Figure 3.4: JSON configuration properties of an IRS with a patch highlighted in yellow.

$\text{atan2}\left(\frac{y^{\text{a}}-y^{\text{b}}}{x^{\text{a}}-x^{\text{b}}}\right)$. Similarly, θ^{BG} denotes the inclination angle related to the direct GU-BS link with respect to the GU. Moreover, the orientation of the IRS can be considered independent of the UAV motion by assuming that the IRS is attached by means of a gimbal, which electronically compensates such perturbations.

Such model is implemented in the core of IoD-Sim as a set of classes depicted in Figure 3.2. In particular, PHY layer communications are implemented by means of the `ns3::IrsAssistedSpectrumChannel` class, which extends the channel simulation capabilities originally provided by `ns3::MultiModelSpectrumChannel`. Specifically, this object evaluates the overall receiver gain[4], derived in Section 3.4.5, which considers both the reflected links, introduced by the IRSs, and the original direct link between the nodes of interest.

The IRS is described by the `ns3::Irs` class, which extends the generic peripheral one, i.e., `ns3::DronePeripheral`. The adoption of this interface benefits the implementation of the IRS as a device. In fact, it is possible to have (i) a state that can be put in either `OFF`, `IDLE`, or `ON`, and (ii) an associated energy

4 As computational optimisation, the gain is calculated only in the centre frequency of the power spectrum density. Since the bandwidth used by each user is much smaller than the carrier frequency, this approximation leads to a negligible frequency shift and hence an accurate channel gain evaluation.

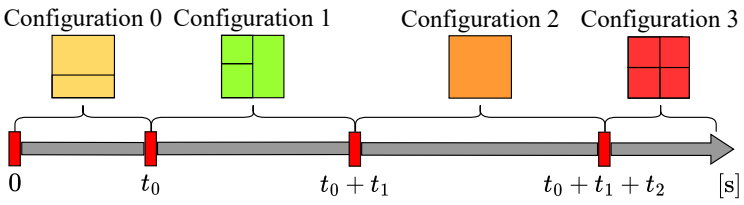


Figure 3.5: Different patch configurations applied over time, following the `ns3::DefinedPatchConfigurator` logic.

consumption model (even though it is negligible wrt. the main components that drain the UAV battery.) All `ns3::Irs` instances are referenced by a global register named `ns3::IrsList`, allowing them to be easily reachable through object paths formatted as `/IrsList/[IRS_Global_Index]`.

As illustrated in Figure 3.4, the IRS is characterised by different properties that can be set through the `ns3::TypeId` attributes: *Rows* and *Columns* for the IRS size; `PruX` and `PruY` for the dimension of each PRU; `RotoAxis` and `RotoAngles` to indicate an ordered sequence of axes and their rotation in degrees, respectively. For instance, `RotoAxis = ["X_AXIS"]` and `RotoAngles = [180]` indicate that the IRS should be rotated by 180 degrees around the x axis, i.e., the surface faces the ground.

Each `ns3::Irs` is organised into one or more `ns3::IrsPatch`, whose dimensions can be specified through the `Size` property: it has four values corresponding to the starting and ending PRUs' indexes along the x and y axes, i.e, `Size[0:1]` and `Size[2:3]`. Once the patches dimensions are set, they can be configured to support the communication of a specific pair of *Serving Nodes*.

In order to provide a flexible and dynamic configurations at runtime, the proposed implementation offers additional configurator classes `ns3::PatchConfigurator` and `ns3::ServingConfigurator`. The former sets up the number and size of IRSs patches, called *Patch Configurators*. The latter schedules the nodes to be served by each patch, namely *Serving Configurators*.

For what concerns *Patch Configurators*, the `ns3::DefinedPatchConfigurator` represents a basic reference already available in the module.

It allows the definition of different patch setups that the IRS adopts over time, as depicted in Figure 3.5. It can be observed that the simulation starts by dividing the IRS in two parts, as specified by *Configuration 0*. At time t_0, the patches are

reorganised to follow the map given by *Configuration 1*. This logic reiterates twice more, until the end of the simulation.

Regarding the nodes to be served, instead, they can be scheduled according to one of the available *Serving Configurator* algorithms: `ns3::DefinedServingConfigurator` which enables the definition of a list of node pairs to assist for different time intervals; `ns3::PeriodicServingConfigurator` which schedules node pairs in round-robin fashion for the same amount of time; `ns3::RandomServingConfigurator` randomly chooses which node pair to assist for a fixed time interval.

3.4.5 Channel model

Apart from the channel models available in ns-3 [139], i.e. Okumura Hata [139] and COST 231 [140] for Long-Term Evolution (LTE) and Wi-Fi while ITU-R P.1238 [141] and P.1411 [142] for indoor scenarios, IoD-Sim has introduced and extended the ones used for stratospheric HAPs, GEO satellites, and mobile IRSs.

Stratospheric and space communications

The HAP-to-satellite communication link is modelled according to the 3GPP TR 38.811 specification [51], which is in turn based on the cellular channel model presented in 3GPP TR 38.901 [134]. A first characterisation of the above has also been implemented in ns-3 in the `ns3-ntn` module [112], and eventually extended into IoD-Sim in the `ns3::ThreeGppNTN[...]ChannelConditionModel` and `ns3::ThreeGppNTN[...]PropagationLossModel` classes, as represented in Figure 3.6. Specifically, the simulator supports different 3GPP channel environments, i.e., dense urban, urban, suburban, and rural. The channel model accounts for several attenuation factors: basic path loss, atmospheric absorption, and scintillation.

Basic path loss It is characterised by three main components, and can be written as

$$L^{\mathrm{B}} = L^{\mathrm{FS}} + L^{\mathrm{SF}} + L^{\mathrm{CL}}, \tag{3.15}$$

where L^{FS} is the free-space path loss, L^{SF} is the shadow fading, and L^{CL} represents the clutter loss. The free-space path loss for the NTN scenario is defined as

Figure 3.6: Class diagram of HAP-to-satellite communications in IoD-Sim.

$$L^{\text{FS}} = 32.45 + 20\log_{10} f + 20\log_{10} d'^{\text{HS}}, \tag{3.16}$$

where f is the carrier frequency in GHz and d'^{HS} is the geocentric distance, measured in meters, between a HAP and a GEO satellite. The shadow fading is modelled as a log-normal random variable, i.e.,

$$L^{\text{SF}} \sim \mathcal{N}\left(0, \sigma^2\right), \tag{3.17}$$

and depends on the elevation angle, the visibility condition (i.e., LoS or not,) and the carrier frequency, which reference values are reported in the tables of 3GPP TR 38.811 [51]. The characterisation of the clutter loss follows a similar model, even though it is neglected in LoS.

Atmospheric absorption Unlike in the terrestrial channel, in the NTN scenario the propagation of the signal undergoes an additional attenuation due to penetration through the atmosphere. While the complete model is available in the ITU-R documents [143], the 3GPP TR 38.811 [51] specification adopt a simplified model considering only the annual mean values of absolute humidity, water-vapour density, water-vapour partial pressure, and dry air pressure for the atmosphere. Therefore, atmospheric absorption is given by

$$L^{\text{A}} = \frac{L^{\text{Az}}}{\sin(\vartheta^{\text{HS}})}, \tag{3.18}$$

where L^{Az} is the absorption loss in dB at the zenith angle at a given carrier frequency, which value is given in [143], and ϑ^{HS} is the actual elevation angle with reference to the HAP. The atmospheric absorption is relevant only for frequencies above 10 GHz, or in the case of an elevation angle lower than 10 degrees for all frequencies. An important absorption effect is due to the presence of oxygen, which produces a very significant attenuation at frequencies around 60 GHz [143, Annex 1.1].

Scintillation It determines the rapid fluctuations of the phase and the amplitude of the signal, caused by small-scale changes in the structure of the atmosphere. Specifically, scintillation is due to two different contributions: tropospheric scintillation and ionospheric scintillation. The former is particularly significant for frequencies above 10 GHz and at low elevation due to the longer

path of the signal, which is obviously not the case. It is modelled as the 99-percentile of the attenuation level observed in Toulouse (France) at 20 GHz, as reported in [51, Figure 6.6.6.2.1-1]. Ionospheric scintillation, instead, is relevant only for latitudes below 20 degrees, or for frequencies below 6 GHz. It is expressed as

$$L^{\mathrm{ISc}} = \left(\frac{f}{4}\right)^{-1.5} \frac{\mathrm{L}^{\mathrm{IScFluc}}(4\,\mathrm{GHz})}{\sqrt{2}}, \tag{3.19}$$

where $\mathrm{L}^{\mathrm{IScFluc}}(4\,\mathrm{GHz})$ represents the ionospheric attenuation level observed in Hong Kong between March 1977 and March 1978 at a frequency of 4 GHz [51, Figure 6.6.6.1.4-1].

Mobile Intelligent Reflective Surfaces

As for IRS-aided communications[5], the channel model assumes the use of the OFDMA scheme, which prevents interference among the involved entities. The GU and the BS employ a single-antenna for data exchange, that, together with each IRS element, are characterised by power radiation pattern functions (including antenna gains) denoted by F^{GU}, F^{BS}, and F^{IRS}. According to [144], the channel gain $g^{\mathrm{BG}} \in \mathbb{C}$ of the direct GU-BS link is

$$g^{\mathrm{BG}} = \sqrt{\beta^{\mathrm{BG}} d^{\mathrm{BG}\,-\alpha} F^{\mathrm{BG}}}\, h^{\mathrm{BG}}, \tag{3.20}$$

where β^{BG} is the channel power gain at the reference distance of 1 m, α is the path loss exponent, and $F^{\mathrm{BG}} = F^{\mathrm{BS}} F^{\mathrm{GU}}$. Moreover, h^{BG} is the channel coefficient expressed as

$$h^{\mathrm{BG}} = \sqrt{\frac{\kappa^{\mathrm{BG}}}{\kappa^{\mathrm{BG}}+1}}\,\bar{h}^{\mathrm{BG}} + \sqrt{\frac{1}{\kappa^{\mathrm{BG}}+1}}\,\tilde{h}^{\mathrm{BG}}, \tag{3.21}$$

where $\bar{h}^{\mathrm{BG}} \in \mathbb{C}$, with $|\bar{h}^{\mathrm{BG}}| = 1$, is the LoS deterministic component which describes the large-scale fading phenomena, and $\tilde{h}^{\mathrm{BG}} \sim \mathcal{CN}(0,1)$ is the non line of sight (NLoS) contribution which accounts for the small-scale fading and follows a circularly-symmetric complex Gaussian distribution with zero mean and unit variance. The envelope $|h^{\mathrm{BG}}|$ is generally Rician [145], with K-factor κ^{BG}

5 The main notation adopted in this Section has been summarised in Table 3.4.

and average power $\Omega^{\text{BG}} = 1$. Specifically, κ^{BG} can be expressed as a function of the elevation angle and reads

$$\kappa^{\text{BG}} = \kappa^{\text{MIN}} \exp\left(\frac{2}{\pi} \ln \frac{\kappa^{\text{MAX}}}{\kappa^{\text{MIN}}} \left| \frac{\pi}{2} - \theta^{\text{BG}} \right| \right), \tag{3.22}$$

with κ^{MIN} and κ^{MAX} the minimum and maximum possible K-factors, respectively. Similarly to Equations (3.20), (3.21), and (3.22), given the p-th patch of the u-th UAV, the channel gains $\mathbf{g}_{u,p}^{\text{UG}} \in \mathbb{C}^M$ and $\mathbf{g}_{u,p}^{\text{BU}} \in \mathbb{C}^M$, related to the GU and the BS, can be formulated as follows:

$$\mathbf{g}_{u,p}^{\text{UG}} = \sqrt{\beta^{\text{UG}} {d_u^{\text{UG}}}^{-2} F^{\text{UG}}} \mathbf{h}_{u,p}^{\text{UG}}, \tag{3.23}$$

$$\mathbf{g}_{u,p}^{\text{BU}} = \sqrt{\beta^{\text{BU}} {d_u^{\text{BU}}}^{-2} F^{\text{BU}}} \mathbf{h}_{u,p}^{\text{BU}}, \tag{3.24}$$

where $\mathbf{g}_{u,p}^{\text{UG}}$ and $\mathbf{g}_{u,p}^{\text{BU}}$ represent the channel vectors of the links characterised by K-factors κ_u^{UG} and κ_u^{BU}, respectively. Since each patch p coherently reflects the incident signal from the BS towards a GU and vice versa, all the phase shifts can be described in terms of two parameters, $\phi_{u,p}^{\text{X}}$ and $\phi_{u,p}^{\text{Y}}$, thus reducing the degrees of freedom by imposing that:

$$\ell\left(\left(i - \frac{1}{2}\right)\phi_{u,p}^{\text{X}} + \left(i' - \frac{1}{2}\right)\phi_{u,p}^{\text{Y}}\right) = \phi_{u,p,m}, \tag{3.25}$$

being $\ell = \frac{2\pi f d}{c}$ and c the speed of light. The overall channel gain that characterises the communication of a GU served by the swarm is

$$\Gamma = \sum_{u=1}^{U} \sum_{p=1}^{P} {\mathbf{g}_{u,p}^{\text{BU}}}^{\mathsf{T}} \mathbf{\Phi}_{u,p} \mathbf{g}_{u,p}^{\text{UG}} + g^{\text{BG}}, \tag{3.26}$$

which is intractable due to the product of complex Gaussians. Nonetheless, according to [144], the envelope can be approximated to a Rician random variable having K-factor $\kappa = \frac{v^2}{2\sigma^2}$ and average power $\Omega = v^2 + 2\sigma^2$, with v^2 and $2\sigma^2$ defined as

$$\nu^2 = \sum_{u=1}^{U}\sum_{p=1}^{P}\mu_{u,p}^2 + 2\sum_{u\geq u'}\sum_{p>p'}|\mu_{u,p}||\mu_{u',p'}|\cos(\omega_u-\omega_{u'})$$
$$+\lambda^2\overline{\kappa}^{\mathrm{BG}}+2\sum_{u=1}^{U}\sum_{p=1}^{P}|\mu_{u,p}||\lambda\sqrt{\overline{\kappa}^{\mathrm{BG}}}|\cos\left(\omega_u+\frac{\ell d_g^{\mathrm{BG}}}{d}\right), \tag{3.27}$$

$$2\sigma^2 = N\sum_{u=1}^{U}\eta_u^2\widetilde{\kappa}_u^{\mathrm{BUG}}+\lambda^2\widetilde{\kappa}^{\mathrm{BG}}, \tag{3.28}$$

with

$$\mu_{u,p} = \eta_u\sqrt{\overline{\kappa}_u^{\mathrm{BUG}}}\frac{\sin\left(\frac{\ell M^{\mathrm{C}}}{2}\psi_{u,p}^{\mathrm{X}}\right)\sin\left(\frac{\ell M^{\mathrm{R}}}{2}\psi_{u,p}^{\mathrm{Y}}\right)}{\sin\left(\frac{\ell}{2}\psi_{u,p}^{\mathrm{X}}\right)\sin\left(\frac{\ell}{2}\psi_{u,p}^{\mathrm{Y}}\right)}e^{-j\omega_u}, \tag{3.29}$$

$$\overline{\kappa}_u^{\mathrm{BUG}} = \frac{\kappa_u^{\mathrm{BU}}\kappa_u^{\mathrm{UG}}}{(\kappa_u^{\mathrm{BU}}+1)(\kappa_u^{\mathrm{UG}}+1)}, \widetilde{\kappa}_u^{\mathrm{BUG}} = \frac{\kappa_u^{\mathrm{BU}}+\kappa_u^{\mathrm{UG}}}{(\kappa_u^{\mathrm{BU}}+1)(\kappa_u^{\mathrm{UG}}+1)}, \tag{3.30}$$

$$\psi_{u,p}^{\mathrm{X}} = \sin\theta_u^{\mathrm{BU}}\cos\varphi_u^{\mathrm{BU}}+\sin\theta_u^{\mathrm{UG}}\cos\varphi_u^{\mathrm{UG}}+\phi_{u,p}^{\mathrm{X}}, \tag{3.31}$$

$$\psi_{u,p}^{\mathrm{Y}} = \sin\theta_u^{\mathrm{BU}}\sin\varphi_u^{\mathrm{BU}}+\sin\theta_u^{\mathrm{UG}}\sin\varphi_u^{\mathrm{UG}}+\phi_{u,p}^{\mathrm{Y}}, \tag{3.32}$$

$\omega_u = \ell\left(d_u^{\mathrm{BU}}+d_u^{\mathrm{UG}}\right)$, $\eta_u = \sqrt{\beta^{\mathrm{BUG}}d_u^{\mathrm{BU}-2}d_u^{\mathrm{UG}-2}F^{\mathrm{BUG}}}$, $\lambda = \sqrt{\beta^{\mathrm{BG}}d^{\mathrm{BG}-\alpha}F^{\mathrm{BG}}}$, $F^{\mathrm{BUG}} = F^{\mathrm{BU}}F^{\mathrm{UG}}$, $\overline{\kappa}^{\mathrm{BG}} = \frac{\kappa^{\mathrm{BG}}}{\kappa^{\mathrm{BG}}+1}$, $\widetilde{\kappa}^{\mathrm{BG}} = (\kappa^{\mathrm{BG}}+1)^{-1}$, and $\beta^{\mathrm{BUG}} = \beta^{\mathrm{BU}}\beta^{\mathrm{UG}}$.

Finally, given an outage probability ε, the channel power gain can be lower-bounded [144] as

$$\Gamma_\varepsilon = \frac{\zeta^2\Omega}{2(\kappa+1)}, \tag{3.33}$$

$$\zeta = \begin{cases}\sqrt{-2\log(1-\varepsilon)}e^{\frac{\kappa}{2}}, & \text{for } \kappa \leq \frac{K_0^2}{2}\\ \sqrt{2\kappa}+\frac{1}{2Q^{-1}(\varepsilon)}\times & \\ \quad\log\left(\frac{\sqrt{2\kappa}}{\sqrt{2\kappa}-Q^{-1}(\varepsilon)}\right)-Q^{-1}(\varepsilon), & \text{for } \kappa > \frac{K_0^2}{2}\end{cases} \tag{3.34}$$

with $Q^{-1}(x)$ being the inverse Q-function and K_0 the intersection of the sub-

functions at $\sqrt{2\kappa} > Q^{-1}(\varepsilon)$. Equations (3.33) and (3.34) are used in the implementation of the channel model in case of scenarios with IRSs.

3.4.6 Mobility

The simulator lies its foundations on models already provided by ns-3, e.g., waypoint, constant acceleration, and constant velocity mobility models. However, a significant limitation arises, as it is difficult with these models to describe a curvilinear trajectory, which also take into account the key points of interest in the mission plan. Another aspect to consider is that models such as `ns3::WaypointMobilityModel` couples the position of the drone with a given time instant, without taking into account the limitations imposed by the maximum speed of the UAV. Therefore, if the user does not properly design the path, this could lead to a simulation which does not reflect the reality. Moreover, in the setup phase it is necessary to specify all the points that create the trajectory.

To overcome these limitations, dedicated mobility models have been developed. In particular, the trajectory has been modelled using Bézier curves by specifying a set of PoIs. These are decoupled from the time of arrival, and the resulting trajectory is bounded to the mechanical characteristics of the drone. A specific structure implemented in IoD-Sim, namely `ns3::CurvePoint`, describes the 3D position vector of the Bézier curve together with the distances from the previous point and the starting one. Besides, a container object, i.e., `ns3::Curve`, is in charge of managing the points of the curve, i.e., `ns3::CurvePoint`, that are defined according to the interest points contained in a `ns3::FlightPlan`. When a `ns3::Curve` is instantiated, it populates the container according to the following.

Let $\mathbf{P} = \{\mathbf{P}_0, \mathbf{P}_1, \ldots, \mathbf{P}_{N-1}\}$ with $\mathbf{P}_i \in \mathbb{R}^3,\ \forall i = 0, \ldots, N-1$ be an ordered sequence of N interest points, $\mathbf{l} = \{l_0, l_1, \ldots, l_{N-1}\}$, $l_i \in \mathbb{N}^+$, the interest level associated to each point, $\Lambda = \left(\sum_{i=0}^{N-1} l_i\right) - 1$ and $L_i = \sum_{h=0}^{i-1} l_h$. The *trajectory generator* can be expressed as

$$\mathbf{G}(t) = \sum_{i=0}^{N-1} \mathbf{P}_i \sum_{j=0}^{l_i-1} \binom{\Lambda}{L_i + j} (1-t)^{\Lambda - L_i - j} t^{L_i + j}, \quad t \in [0,1] \tag{3.35}$$

It is worth noting that (3.35) is a revised version of the original Bézier equation, which does not practically allow to reach the interest points, except for

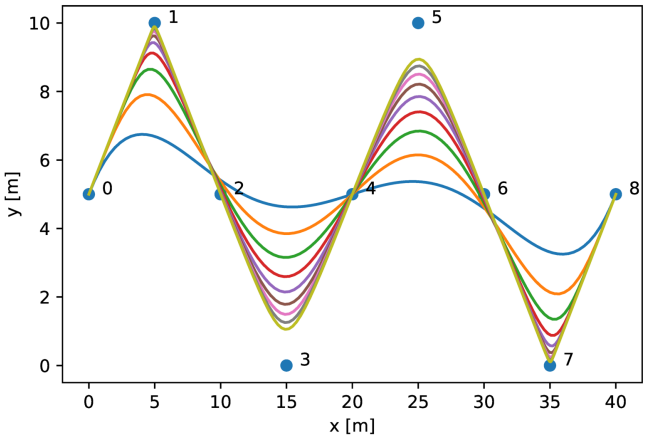


Figure 3.7: A set of trajectories, generated with (3.35), with different interest levels (from 1 to 10, incrementally) for PoIs 1, 3, 5, and 7. The other points have constant interest level set to 1.

the first and last one. An increment in the interest level l turns into a trajectory that passes closer to that point, as illustrated in Figure 3.7. A special case takes place when $l = 0$. A specific mechanism is provided to split the trajectory into two contiguous curves so that the drone is forced to fly over them. In this case, a `restTime` can be defined to set the hovering duration in seconds. Finally, the obtained trajectory is used by the new implemented models, i.e., `ns3::ConstantAccelerationDroneMobilityModel`, `ns3::ParametricSpeedDroneMobilityModel`, and the `Geographic` equivalents.

Constant acceleration drone mobility model

This mobility model employs Equation (3.35) and the uniform acceleration motion law to retrieve the points of the desired trajectory. Since the speed of the drone cannot increase indefinitely, after the maximum speed is reached, the uniform linear motion law is adopted. This object is implemented as an ns-3 model, and hence, has its own `TypeId` with attributes described in Table 3.5. In each instant of the simulation, IoD-Sim calls two methods, `DoGetPosition()`

Table 3.5: `ns3::ConstantAccelerationDroneMobilityModel TypeId` attributes.

Attribute	Description
`Acceleration`	Drone's constant acceleration, expressed in m/s^2.
`MaxSpeed`	Drone's maximum speed, expressed in m/s.
`FlightPlan`	Interest points for the trajectory.
`SimulationDuration`	Simulation duration, expressed in seconds.
`CurveStep`	Discretization step of the curve.

Table 3.6: `ns3::ParametricSpeedDroneMobilityModel TypeId` attributes.

Attribute	Description
`SpeedCoefficients`	The set of coefficients for the polynomial $v(t)$.
`FlightPlan`	Interest points of the trajectory.
`SimulationDuration`	Simulation duration, expressed in seconds.
`CurveStep`	Discretization step of the curve.

and `DoGetVelocity()`. They return both the position and the speed at current time of the drone, that is recomputed thanks to the `Update()` method.

Parametric speed drone mobility model

Similarly to the previous model, this mobility model is implemented as a ns-3 `TypeId`. However, this one takes a $v(t)$ speed profile in a polynomial form and, thanks to the modified Bézier equation (3.35), it retrieves the discretized trajectory. To ease the implementation, a specific attribute, i.e., `ns3::SpeedCoefficients`, is introduced to serve as a container of the $v(t)$ coefficients. These are elaborated (by employing the GNU Scientific Library) to constantly update the parameters by calling `UpdateSpeed()` and `UpdatePosition()` subroutines. A summary of the attributes of this mobility model is reported in Table 3.6.

3.4.7 Applications

IoD-Sim offers simple applications that can be used to communicate telemetry from a drone to a ZSP or to a *Remote* by adopting client-server paradigm, via User Datagram Protocol (UDP). Moreover, relying on the same architecture, two TCP-based applications are available to enable reliable data transfer between hosts. Besides, a Network Address Translation (NAT) application is provided to design relaying network architectures.

Telemetry applications

These applications are modelled as classes named `ns3::DroneClientApplication` and `ns3::DroneServerApplication`. The model asks for the `DestinationIpv4Address` and a `Port` of the remote entity that hosts the server application. Data are sent every `TransmissionInterval` and, whereas the drone has a storage peripheral, it is possible to free an equivalent amount of memory space. The configuration parameters are summarised in Table 3.7.

When the application is started, through the `ns3::Application::StartApplication()` method, a UDP-based communication, employing application-level acknowledgements, takes place. It is worth specifying that the application is stateful in order to support the *rendezvous process* which discovers the application server in the network, if no address is given. This process starts with the client application in NEW state. Therefore, a HELLO packet is sent to the destination address (or in broadcast,) thus implying a state transition in HELLO_SENT. If the application server receives such packet, it replies with an HELLO_ACK packet to confirm the reception. When the client receives the acknowledgement, its state changes again, into CONNECTED, which allows it to periodically send telemetry data. These packets are named UPDATE and UPDATE_ACK. The entire procedure is depicted in Figure 3.8.

The JSON-encoded telemetry is periodically transmitted, through the `SendPacket()` method, and received by the application server, through the `ReceivePacket()` method. HELLO and UPDATE packets transport a payload which is formatted in JSON with ASCII encoding. Its content is a JSON object with the following properties:

- The unique `id` of the drone in the simulation. This ensures that mobile entities, i.e., drones and HAPs, communications can be tracked over complex scenarios.

Table 3.7: Configuration parameters for telemetry applications.

Applicable to	Name	Type	Default Value	Description
Client	`DestinationIpv4Address`	String	`255.255.255.255`	IPv4 address of the remote application server.
Client and Server	`Port`	Unsigned Integer 32-bit	`80`	Port of the remote application server or listening port in the case of the server.
Client	`TransmissionInterval`	Double	`1.0`	Transmission interval of the telemetry updates being sent, in seconds.
Client and Server	`StartTime`	Double	Start of Simulation	Time at which to start the application, in seconds.
Client and Server	`StopTime`	Double	End of Simulation	Time at which to stop the application, in seconds.
Client	`FreeData`	Boolean	`false`	Free data from the equipped storage peripheral when they are transmitted.
Server	`StoreData`	Boolean	`false`	Store data to the equipped storage peripheral when they are received.

Table 3.8: Configuration parameters for generic traffic applications.

Applicable to	Name	Type	Default Value	Description
All Server and Clients	`Address`	String	`127.0.0.1`	Listening or remote address of the server.
All Server and Clients	`Port`	Unsigned Integer 16-bit	`4242`	Listening or remote port of the server.
All Clients	`PayloadSize`	Unsigned Integer 16-bit	`65470`	Size of the payload for each packet, in bytes. in the case of Storage Client, it is the maximum size to be used when freeing storage memory.
Periodic Client only	`Frequency`	Double	`1.0`	Number of times in a second when a new packet is sent to the server.

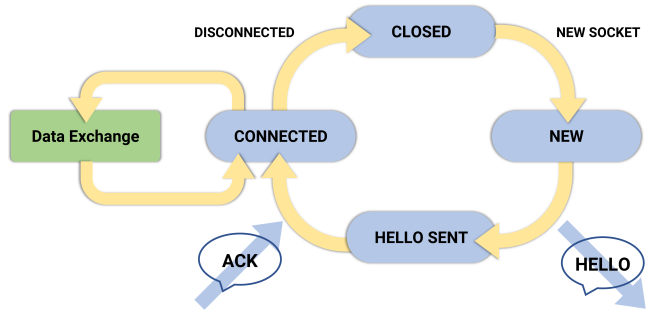


Figure 3.8: Finite state machine of the drone client and server application.

- An incremental `Ssn` that refers to the *sequence number*. It is used to easily check if a packet has been lost.
- `cmd` that refers to the type of packet, whether if HELLO, UPDATE, or an acknowledgment.
- `gps` coordinates with `lat` for latitude, `lon` for longitude, `alt` for altitude, and `vel` for the velocity vector. For simulated drones, the location refers to the virtual world coordinates.

The UDP packet payload is summarised in Table 3.9. When the application is stopped, the `StopApplication()` method is called. Clearly, these applications are developed so that multiple instances can run concurrently on the same entity if different ports are specified. Moreover, they are independent of the particular communication technology adopted.

Generic traffic applications

These applications model a reliable data transfer between a client and a server, which are implemented as `TcpPeriodicClientApplication` and `TcpEchoServerApplication` objects, respectively. The aim is to transfer a certain amount of information between the two hosts according to the specified `PayloadSize`, expressed in bytes, and `TransmissionFrequency`, measured in Hz, set on the client. The server is characterised by a socket, composed by a listening `Address` and `Port`. These configuration parameters are summarised in Table 3.8. To facilitate traffic analysis, each packet has a protocol data unit, formed

Table 3.9: UDP payload.

Field Name	Data Type	Description
`id`	Unsigned Integer 32-bit	ns-3 Global Node Identifier
`sn`	Unsigned Integer 32-bit	Packet Sequence Number
`cmd`	String	Packet Type
`gps`	Object	Drone location in space
`lat`	Double	Drone latitude
`lon`	Double	Drone longitude
`alt`	Double	Drone altitude
`vel`	Array of 32-bit Integers	Drone velocity in m/s

by a 12 bytes header, and the payload. The former contains information-level sequence number and the timestamp of creation; the latter is characterised by a recurring sequence of 16 bits that is incremented over time. These applications provide dedicated `TraceSource` objects that notify communication-related events such as new/closed connections and sent/received packets. An additional TCP-based client has been created to support drones that are typically equipped with a `StoragePeripheral`. To this end, `TcpStorageClientApplication` monitors the storage and, if memory is used, it transfers data to the remote server. If the transfer is acknowledged, memory is freed. This mechanism is relevant when drones are equipped with limited on-board memory. Indeed, the client can be used to transfer as much data as possible over the wireless medium to prevent out-of-memory events.

Relaying application

The *relaying application* is implemented through the class `ns3::NatApplication`. It is a specialised networking application that, given an `InternalNetDeviceId` and an `ExternalNetDeviceId`, provides a mechanism similar to NAT to a set of drones/HAPs placed in an internal network. The `NetDeviceId` is a numerical identifier that uniquely points to a network device mounted on the drone. During initialization, i.e., `DoInitialize()` method, the application modifies the static routing table of the internal network device to redirect all traffic to the loopback device. A specific callback, namely `RecvPktFromNetDev()`, notifies when a new frame arrives. It contains information such as the

Table 3.10: Memory organisation of protocol stacks used by the scenario configuration interface.

std::array

std::vector

Stack ID \ Layer	PHY	MAC	NET
0	WifiPhy	WifiMac	IPv4
1	WifiPhy	WifiMac	IPv6
2	LtePhy	LteMac	IPv4
⋮	⋮	⋮	⋮

Internet Assigned Numbers Authority standard layer 3 protocol identifier and the sender/receiver MAC addresses. The NAT forwarding behaviour leverages a hash map, i.e., *NAT Table*, where an external port number is coupled with the source IP address and port. Inbound frames are forwarded to the external network by replacing this information with the one of the relaying drone. The same rationale is applied for frames received from the external network.

3.4.8 Scenario configuration interface

The *scenario configuration interface* is an abstraction layer that allows the configuration of the entire simulation by means of JSON files. Indeed, they can be decoded and validated through RapidJSON in order to setup the simulation models. The output data classes are then used by the *general purpose scenario* to initialize objects that define the environment, the entities, and the simulator engine. To this end, the set of all objects that are used to characterise a scenario can be grouped into three categories:

- *Configuration objects* – Models that store parameters in a structured way, easily accessible in the C++ language.
- *Configuration helpers* – Checkers and decoders with the goal to produce a Configuration object or throw an error message.
- *Simulation helpers* – Objects that help organise pointers to structures commonly found in scenario development. They are used in the protocol stack matrix, shown in Table 3.10.

Additionally, *factory helpers* are defined as weakly-coupled extensions to ns-3 internal data structures to ease their initialisation. They are made to minimise modifications made to the ns-3 core framework, which is used by IoD-Sim. The entire system has been made extensible by design, so that it is possible to support further technologies and configurations with the addition of new configuration objects and helpers as needed. In this way, it is possible to further develop high-level configuration interfaces able to setup scenarios and hence to ease the design activity undertaken by the user.

Scenario configuration objects and helpers

The core of the abstraction layer is the `ns3::ScenarioConfigurationHelper`, a low-level object that directly deals with the JSON configuration file. This helper returns a set of specific data classes that contain exclusively the parameters required to configure IoD-Sim models. Each of them is also loosely coupled with a JSON validator and parser, also known as configuration helpers. The information embedded in these classes is then deserialised and employed by higher-level objects.

- `ns3::ModelConfiguration` describes `ns3::TypeId` objects through key-value pairs that reference the model attributes.
- `ns3::EntityConfiguration` describes an entity, whether it is a drone, a HAP. a ZSP, a remote, or a satellite. The object retrieves and stores all parameters related to the `ns3::NetDevice` to be installed on the entity, the mobility model to be applied, and the applications. Optionally, if the entity is a drone or a HAP, there can be defined the mechanics, the battery, and the peripherals. Its parser is called `ns3::EntityConfigurationHelper`.
- `ns3::RemoteConfiguration` denotes key characteristics of remotes. Specifically, a remote needs to know the global network layer ID of reference and the configurations of applications to be installed. Its parser is `ns3::RemoteConfigurationHelper`.
- `ns3::PhyLayerConfiguration` defines the required parameters needed to configure a PHY layer. It is the parent and interface of `ns3::LtePhyLayerConfiguration` and `ns3::WifiPhyLayerConfiguration` data classes. Its parser is `ns3::PhyLayerConfigurationHelper`.

- The `ns3::LtePhyLayerConfiguration` gets all the information needed to set up a PHY layer for LTE, such as its propagation loss model and its spectrum model.
- `ns3::WifiPhyLayerConfiguration` sets up the PHY layer of a Wi-Fi based protocol stack. The PHY layer configuration requires the higher-level Wi-Fi standard to be used, the antenna Rx gain, the data rate, the propagation delay and loss models.
- `ns3::MacLayerConfiguration` collects the required parameters needed to configure a MAC Layer. It is the parent and interface of `ns3::WifiMacLayerConfiguration`. Its parser is `ns3::MacLayerConfigurationHelper`.
- `ns3::WifiMacLayerConfiguration` configures a Wi-Fi basic service set (BSS). The service set identifier (SSID) and access point parameters are defined to create its basic infrastructure.
- `ns3::NetworkLayerConfiguration` defines the required parameters needed to configure the appropriate network layer. It is parent to the `ns3::Ipv4NetworkLayerConfiguration`. Its parser is named `ns3::NetworkConfigurationHelper`.
- `ns3::Ipv4NetworkLayerConfiguration` stores the network address and mask of the configured IPv4 Layer in the configuration file.
- `ns3::LteBearerConfiguration` decodes all the relevant parameters for an LTE bearer, such as its type and the QoS defined as a tuple of *guaranteed bit rate* and *maximum bit rate*.
- `ns3::LteNetdeviceConfiguration` collects the information needed by an LTE network device, such as its bearers. The *role* of the network device is then detected, whether it is a UE or an eNB.
- `ns3::NetdeviceConfiguration` defines for a generic network device. The main parameter stored is the global network layer ID, which is used to detect the stack and network to be attached when the network device is created and installed on a node. A specific configuration for Wi-Fi network devices is handled by `ns3::WifiNetdeviceConfiguration` with relevant MAC data to connect to the BSS.

Scenario simulation helpers

To enable complex scenarios that are related to the future integrated T/NTN communication paradigms, IoD-Sim enables the simulation of T/NTN networks in which multiple telecommunication protocols are used at the same time, applicable to drones, HAPs, and ZSPs. Currently, IoD-Sim supports two communication technologies that can be used concurrently: LTE and the Institute of Electrical and Electronics Engineers (IEEE) 802.11 family. Each protocol stack must be applied to a dedicated network device, i.e., `ns3::NetDevice`. The architecture of the simulator has been designed so that it eases the configuration phase. In order to facilitate the implementation and the installation of protocol stacks on the entities, additional helpers named *simulation helpers* have been developed to arrange the necessary common infrastructure to simulate communications among nodes. Thus, the developed *simulation helpers* are:

- `ns3::WifiPhySimulationHelper`, that initializes the PHY layer of a Wi-Fi-based protocol stack.
- `ns3::WifiMacSimulationHelper`, that creates the objects related to IEEE 802.11 MAC.
- `ns3::LtePhySimulationHelper`, that allocates the necessary resources to enable LTE communications.
- `ns3::Ipv4SimulationHelper`, that manages IPv4 networks for each protocol stack.

All the aforementioned can cooperate with the existing helpers in ns-3, such as `ns3::LteHelper`, `ns3::WifiHelper`, `ns3::YansWifiPhyHelper`, and `ns3::WifiMacHelper`.

General purpose scenario

A flexible and highly dynamic *general purpose scenario* has been developed in order to setup scenario's entities and, at the same time, to provide abstractions which minimise the effort from a programming perspective. It is fully dependent on a semantic analyser and allows the entire simulation platform to be compiled beforehand, providing ways to dynamically reconfigure the scenario at run-time. Its development started from the analysis and the detection of

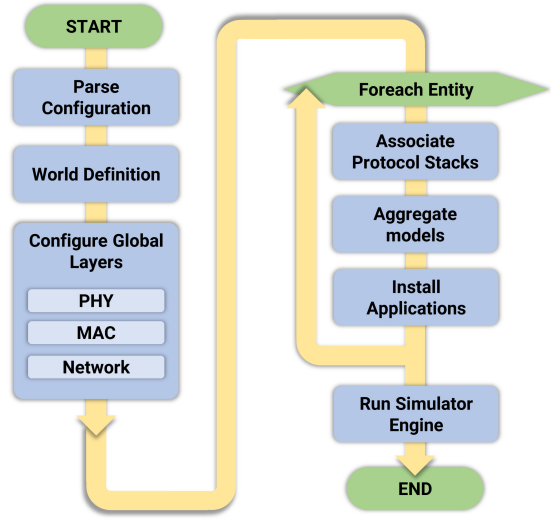


Figure 3.9: Logical flow to initialize and configure a scenario in IoD-Sim.

a common structure typically followed by the Open Systems Interconnection protocol stack. The entire workflow, depicted in Figure 3.9, is described hereby. The *general purpose scenario* is composed of two main parts: *configuration* and *run*. Scenario configuration, executed through the constructor `Scenario()`, is interleaved with the *scenario configuration interface*. The *run* part is identified by `operator()()` which is characterised by minimal C++ code that starts the ns-3 simulator engine. Moreover, it shows the progress status on the console and, optionally, it saves messages to a log file. The *general purpose scenario* requires the initialization of the *scenario configuration interface* through a JSON configuration file. Once the file is decoded, the number of entities are retrieved to create the initial structures, such as a `ns3::DroneContainer` and four `ns3::NodeContainer` objects. They keep track of ZSPs, remotes, and nodes that participate in the *Backbone Network*. Once the entities are created, they are registered to their respective global lists, such as `ns3::DroneList`, `ns3::ZspList`, and `ns3::RemoteList`. After entity creation, the ns-3 static configuration parameters are applied to the simulation. The method is called `ApplyStaticConfig()`. These parameters are a set of key-value pairs that represent certain features of ns-3 models. World definition is made through

`ConfigureWorld()` method. It is related to the configuration of buildings and RoIs. The virtual world set up is then followed by the configuration of PHY, MAC, and network global layers. As for the PHY layer part, if it is made for a Wi-Fi communication stack, the `ns3::WifiPhySimulationHelper` is employed with the specifications stored in `ns3::WifiPhyLayerConfiguration`. If the PHY layer is for LTE, instead, the `ns3::LtePhySimulationHelper` is set up with `ns3::LtePhyLayerConfiguration` parameters. The same procedure is applied for the global MAC layer configuration. The global *Network* layer is managed by `ns3::Ipv4SimulationHelper` for IPv4 networks with the specifications given by `ns3::NetworkLayerConfiguration`, i.e., network address, mask, and a default route. Global stacks are then linked to the configured entities. Moreover, for LTE devices, the bearer is created to ensure that applications have a logical communication channel with desired properties. When the entity network configuration is done, the mobility model is configured and the applications are installed. Furthermore, if the entity is a drone or a HAP, its peripherals are installed, together with the associated energy model. Once all entities are ready, the virtual internet backbone is configured. A CSMA bus is made for the backbone network, identified with address `200.0.0.0/8`. Hosts that can be part of this backbone network are remotes, *Packet Gateways* in the case of an LTE core network, or other routers in the case of the presence of a Wi-Fi BSS. Finally, in the case of LTE networks, their Radio Environment Maps are set up to generate images that represent the radiation map of the RAN.

Configuration schema

The entire scenario has been made parametric through the use of a JSON configuration file. Requested at startup, it is decoded and employed to configure and execute the simulation. In this work, the following configuration schema has been chosen for the General Purpose Scenario:

- `name` – A mandatory string representing the scenario name.
- `dryRun` – An optional boolean to run only the semantic analyser and check that the configuration file and model setup is valid. By default, it is set to `false`.
- `resultsPath`– A mandatory string representing an existing path to store simulation output files.

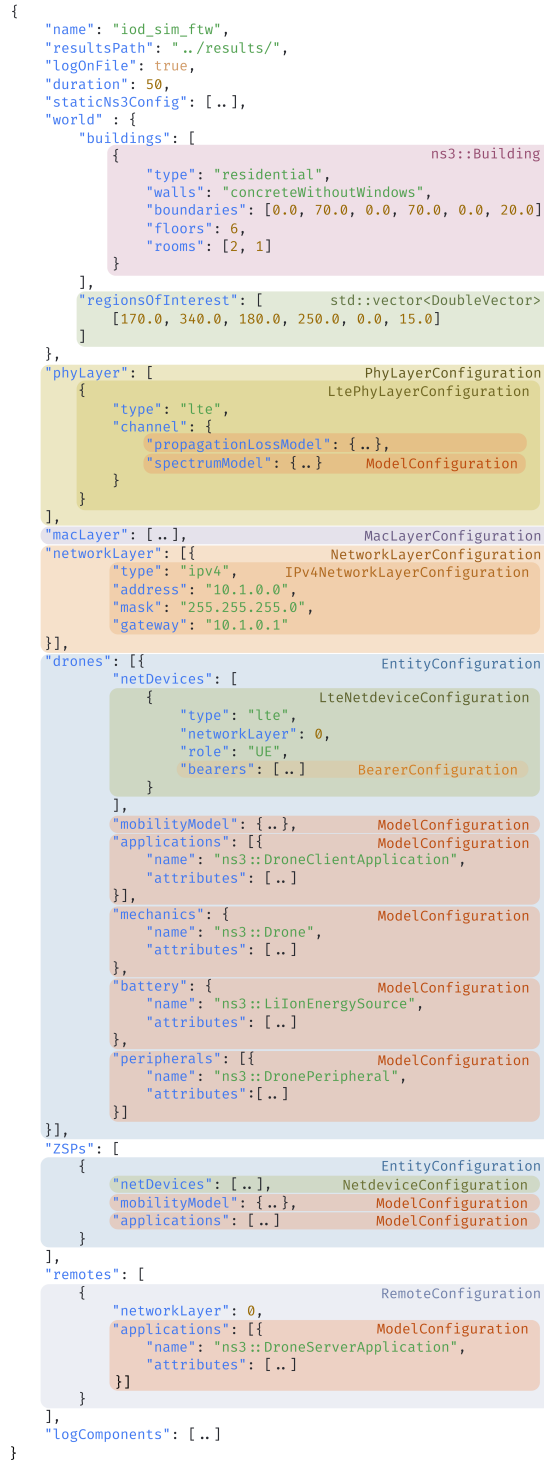


Figure 3.10: An excerpt of scenario configuration with an overlay of the models associated to the analysed parts.

```
{ "phyLayer": [{
  /* *** PHY Layer Characteristics *** */
  "channel": {
    "spectrumModel": {
      "name": "ns3::IrsAssistedSpectrumChannel",
      "attributes": [{"name": "KMin", "value": 6.0},
                     {"name": "KMax", "value": 10.0},
                     {"name": "KNlos", "value": 0.0},
                     {"name": "AlphaLoss", "value": 2.0},
                     {"name": "NoDirectLink", "value": false},
                     {"name": "NoIrsLink", "value": false},
                     {"name": "OutageProbability",
                      "value": 1e-2},
                     {"name": "MultipathInterference",
                      "value": "SIMULATED"}]
  }}}],
  /* *** Other Scenario Properties *** */
  "drones": [{
    /* *** Drone Properties *** */
    "peripherals": [
      { "name": "ns3::Irs",
        "attributes": [{"name": "Rows", "value": 100},
                       {"name": "Columns", "value": 100},
                       {"name": "PruX", "value": 0.01},
                       {"name": "PruY", "value": 0.01},
                       {"name": "RotoAxis","value": ["X_AXIS"]},
                       {"name": "RotoAngles", "value": [180.0]},
                       {"name": "PowerConsumption",
                        "value": [0.0, 1.0, 3.3]}],
        "aggregates": [{"name": "ns3::DefinedPatchConfigurator",
          "attributes": [{"name": "Configurations",
            "value": [
              [{"Size": [0, 49, 0, 99],
                "aggregates": [{
                  "name": "ns3::DefinedServingConfigurator",
                  "attributes": [
                    {"name": "ServingPairs",
                     "value": ["/NodeList/0", "/ZspList/0",
                               "/NodeList/1", "/ZspList/0"]},
                    {"name": "Periods", "value": [3.0, 2.0]}]
                }]},
                {"Size": [50, 99, 0, 99],
                "aggregates": [{
                  "name": "ns3::PeriodicServingConfigurator",
                  "attributes": [
                    {"name": "ServingPairs",
                     "value": ["/NodeList/0", "/ZspList/0",
                               "/NodeList/1", "/ZspList/0"]},
                    {"name": "Timeslot", "value": 1.0}]
              }]}],
              [{"Size": [0, 99, 0, 99],
                "aggregates": [{
                  "name": "ns3::PeriodicServingConfigurator",
                  "attributes": [
                    {"name": "ServingPairs",
                     "value": ["/NodeList/0", "/ZspList/0",
                               "/NodeList/1", "/ZspList/0"]},
                    {"name": "Timeslot", "value": 1.0}]
              }]}]]},
          {"name": "Periods", "value": [5.0, 15.0]}
]} ]}] }] }
```

Figure 3.11: Extract of a JSON scenario configuration that employs the IRS-aware spectrum channel model and installs a single IRS on a drone to serve a pair of nodes.

- `logOnFile` – A mandatory boolean to output scenario logging information on a file or on standard output.
- `duration` – A mandatory integer that specifies the simulation duration in seconds.
- `staticNs3Config` – A mandatory array of objects, each with `name` and `value` strings, to address ns-3 static configuration parameters. The array can be empty.
- `world` – An optional object containing the description of the simulated space, in particular whether to place buildings or regions of interest.
- `phyLayer` – A mandatory array of objects, each representing a PHY layer configuration to be used in the scenario. Each PHY object declares its `type`, which is a mandatory string. The chosen type must be supported by the semantic analyser. Additional parameters are specific to the kind of PHY layer being configured, most notable are the chosen propagation delay model and the propagation loss model.
- `macLayer` – Its description is similar to `phyLayer`.
- `networkLayer` – Its description is similar to `phyLayer`.
- `drones` – A mandatory array of objects, each representing a drone to be simulated. A drone requires the following properties to be configured: at least one `netDevices` in order to link it to a protocol stack and setup its network address assignment, a `mobilityModel` according to the ones available on IoD-Sim, at least one `application` that can be installed on a drone, a `mechanics` to define mechanical properties, and a `battery`. Optionally, a `peripherals` array can also be specified in order to equip I/O devices to the drone with a specific `PowerConsumption` indication. They may also be activated by specifying the region of interest through `RoITrigger` parameter.
- `ZSPs` – Its description is similar to `drones`.
- `remotes` – A mandatory array of objects, each representing a remote that is described by its set of `applications`.

- `nodes` – Additional nodes, such as GEO satellites or terrestrial UEs.
- `logComponents` – A mandatory array of strings to enable log components available in IoD-Sim.

An example of JSON configuration file that realises a simple scenario is shown in Figure 3.10.

Regarding the set up of IRSs and their scheduling plans, another JSON excerpt is given in Figure 3.11. Such feature is enabled by the `ns3::ModelConfigurationVector` and `ns3::ModelConfigurationMatrix` in order to dynamically apply different configurations at runtime. As it can be noticed, the channel model can be configured through the parameters that are described in Section 3.4.5, where `OutageProbability` is ϵ, `KMin` is κ^{MIN}, `KMax` is κ^{MAX}, and `AlphaLoss` is α. Furthermore, `NoDirectLink` and `NoIrsLink` represent booleans useful to analyse use cases where the direct and the reflected links are suppressed. `MultipathInterference`, instead, can assume three different values which affect the interference introduced by the direct and reflected links (the second cosine in Equation (3.27)): (i) `DESTRUCTIVE` for a purely destructive interference (i.e., worst case,) (ii) `SIMULATED` for the actual one, and (iii) `CONSTRUCTIVE` for no interference at all (i.e., best case.) Finally, `KNlos` is the K-factor adopted when the direct link is in NLoS. Such parameters can be further tuned to simulate better or worse channel conditions according to the simulation design requirements. IRS configuration can be declared in the JSON as a drone peripheral. In the example given in Figure 3.11, two configurations are applied to the IRS, with different time durations, specified in *Periods*. In the first one, the IRS is split in half with two patches: one patch assists the links of two GUs, with global index 0 and 1, for three and two seconds, respectively; the second one periodically serves the same users for one second each. Further, in the second configuration, the whole IRS is used for 15 seconds to serve both nodes in round-robin fashion for one second each. Finally, a power consumption profile is defined for the IRS controller.

3.5 Simulation development platform

The *scenario configuration interface*, discussed in the previous Section, eases the design and configuration of complex scenarios from a high-level perspective. Indeed, JSON greatly facilitates management and maintainability thanks to its

dry and human-readable syntax. However, the user experience is still hindered by the following:

- As IoD-Sim grows in size and introduces more complex and powerful models over time, the learning curve to effectively use this simulator steepens.
- This project is continuously developed and upgraded with new features, technologies, and standards. A high-level abstraction helps reduce the barrier for scenario developers in approaching new features and the required effort to implement a scenario.
- A general purpose configuration interface, provided in the form of JSON-encoded files, does not give any visual clue on scenario design. Indeed, plain text files alone require low-level knowledge of the simulator, thus implying that the users have to rely on their experience and imagination to effectively know all the aspects related to a complex scenario configuration, such as the number of drones, their trajectories, their purpose, their equipment, the topology of the ground infrastructure, the services exposed by remote nodes, and the links required to set up a 3D integrated T/NTN with HAPs and satellites.
- Error reporting messages cannot be easily understood by end users, forcing the use of a debugger to isolate the problem. Therefore, a semantic analysis would be beneficial to detect problems at scenario configuration.

To address all the points above, the IoD-Sim *simulation development platform* provides a set of extensions, i.e., the *report module*, output files for data analysis, and stand-alone applications for scenario design, such as *airflow*. These tools ease scenario design and analysis, thus ensuring that IoD-Sim can be easily introduced to newcomers, especially university students and researchers.

3.5.1 Report module

The *report module*, illustrated in Figure 3.12, is an extension of IoD-Sim which stores data at run-time and elaborates, at the end of simulation, a comprehensive summary. The aim of the extension is to introspect simulator's data structures to gather relevant data to be reported, e.g., data traffic, trajectory, and telemetry.

In order to provide flexible reporting, the data accumulators, which run through the simulation to acquire data through the use of *traces*, interface with a report abstraction layer to format data according to the drivers available, i.e., XML and time series database (TSDB). The former is both human and machine readable and adheres to a schema to describe the expected structure of the produced file. More insights about the structure of the proposed extension are provided hereby. The root XML element, i.e, `Simulation`, represents the summary of a scenario previously executed. The attributes that characterise the simulation are `scenario`, which is a string that carries the name of the scenario that was executed, and `executedAt`, which reports the date and time of execution of this simulation. Moreover, `Simulation` presents further information about simulation results, such as its `duration`, which is reported in `real` and `virtual` time, `World`, which contains the `Buildings` and `InterestRegions`, and entities containers. The first of these containers is `Zsps`, which is a complex XML type that summarises each ZSP through `position` described by the 3D coordinates, and `NetDevices`, which is a list of configured network devices. Each of them is described by structures that represent the configuration of the PHY, MAC, and network layers, together with the data traffic. Each captured packet is expressed by direction, length in bytes, timestamp, and textual representation of the payload. Similarly, `Drones` summarises the state of each `Drone`. This structure maintains the `NetDevices` already discussed for `Zsps`. Additionally, particular characteristics of drones are reported, such as `trajectory` and the set of onboard `Peripherals`. The former is defined by a list of points, each of them with its own timestamp. The latter reports the characteristics of the used peripherals type. Finally, `Remotes` are described only by their `NetDevices`. This output XML file is put together with other files relevant to the simulation in the `results` directory. Similarly, the TSDB driver[6] interfaces with TimescaleDB, an extension of PostgreSQL, to efficiently store and query time series data. An Structured Query Language (SQL) schema is defined to store Wi-Fi and LTE data, as well as SINR and drones' trajectory. They can be visualised on platforms such as Grafana, which provides a comprehensive set of graphical utilities to query and plot data.

6 The TSDB driver, together with a set of example SQL schemas and Grafana dashboards, is available on the `tsdb` branch of IoD-Sim repository: https://github.com/telematics-lab/IoD_Sim/tree/tsdb.

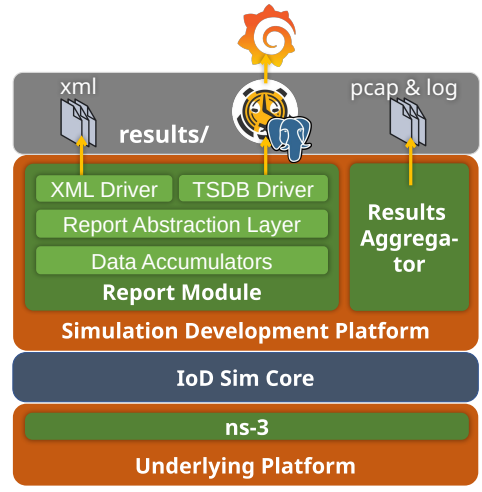


Figure 3.12: Block diagram of the report module.

3.5.2 Results aggregator

Log Files gather all the relevant information and debug messages about the internal components of the simulation. Primarily, the *General Purpose Scenario* emits `progress.log` and `IoD Sim.log` files. The former is the output of the progress information messages that are also delivered on the standard output during scenario execution. The latter contains all debug messages coming from the different internal components of IoD-Sim. The log components can be enabled by specifying them in the `logComponents` field of the scenario configuration JSON file.

`progress.log` file starts by determining the current date and time of the start of the simulation. For each second, it prints a status report on a single line. The status report presents the following fields:

- The simulation time instant at which the report is referring to.
- The speed up in simulating the scenario wrt. real time. This is dependent on the simulator performance and how many events are elaborated.
- The number of events processed in the time interval relative to the previous status report.

The file then ends with the current date and time and the duration of the simulation as *elapsed wall clock*.

Trace Files are ASCII-encoded text files that record all the activities regarding a specific Network Device. All the traces are bounded by what is sent or received at the MAC layer. A *Trace File* name is composed of three fields, separated by a hyphen: (i) the global layer name, (ii) the unique identifier of the host in the network, and (iii) the unique identifier of the host network device. For instance, `internet-2-1.tr` indicates that the trace has been done on the first network device of the second host in the virtual Internet network. *LTE Log Files* are ASCII-encoded text files that represent a series of statistics on relevant KPIs. These log files are focused on specific low-level layers of the LTE stack, particularly PHY, MAC, Radio Link Control, and Packet Data Convergence Protocol. For each layer, there are two separate trace files: one for *downlink* and one for *uplink* communications. As part of the *LTE Log Files*, there are also PCAP traces of the *S1-U* interface that links the RAN with the Evolved Packet Core. *PCAP Files* are well-known files that record network activity in the PCAP format and contain the traffic that occurred on a certain network device of a host. The filename format is similar to *Trace Files*. Due to the fact that these files are binary, a suitable decoder should be used to explore the data structure. A popular decoder is the `libpcap` open-source project, used by frameworks for PCAP data analysis, e.g., *Scapy*, and GUI programs such as *Wireshark*. As these *PCAP Files* are generated by a simulation, each captured frame is marked with the relative timestamp of the simulation. Therefore, each *PCAP File* starts with the transmission/reception of captured frames at 0 seconds.

3.5.3 Airflow

Airflow is a high-level abstraction tool that gives visual clues during simulation design, thus enriching the user experience, especially for newcomers. Airflow has been developed on top of Splash, a specialised transpiler for IoD-Sim. It scans the source code of the simulator and outputs visual blocks that can be referenced in the *Core Editor* to configure a scenario. Thanks to the GUI editor, a scenario can be exported into a JSON file that can be interpreted by IoD-Sim *Scenario Configuration Interface*. From a software design standpoint, as illustrated in Figure 3.13, the Airflow project is entirely decoupled from IoD-Sim. Its integration with the simulator relies on interfaces that enable bidirectional communications.

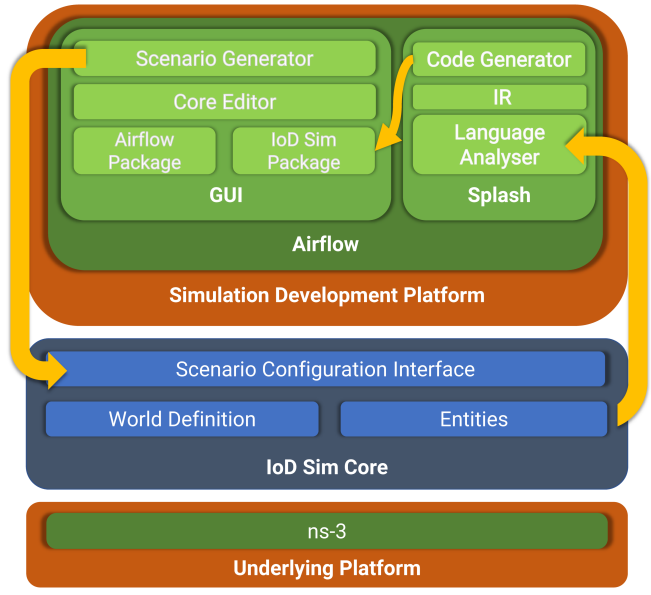


Figure 3.13: Airflow Architectural Design.

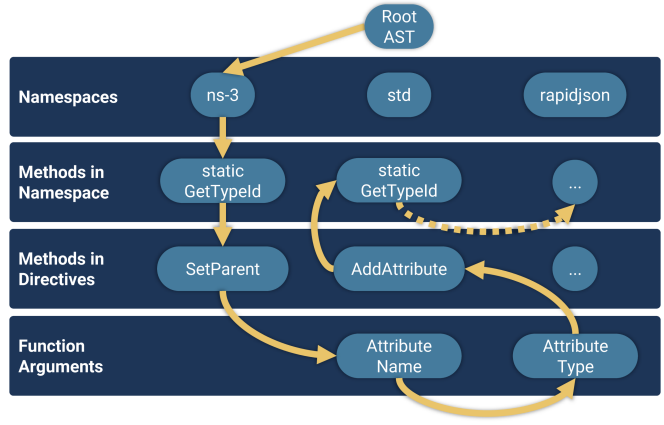


Figure 3.14: The tree-traversal search algorithm employed by Splash to extract the models from the IoD-Sim source code. The numerical ordering given on the edges reflects the algorithm logic used to extract model information.

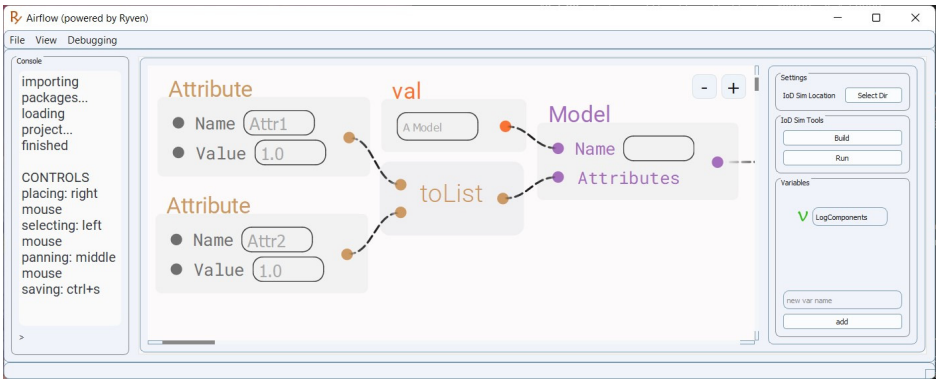


Figure 3.15: An overview of the configuration of a generic model in Airflow.

Splash

Splash is a middleware that analyses IoD-Sim source code and translates ns-3 models into visual block code used by Airflow. These blocks can be added to the editor as external packages. Splash enables the decoupling mechanism, able to ensure that Airflow and IoD-Sim can be developed asynchronously and updated when needed. In particular, it accomplishes the following tasks:

1. Parses the source code of IoD-Sim by relying on Clang lexical and syntax analysers, producing the abstract syntax tree (AST) that is stored into a binary file.
2. Scans the AST to find relevant simulation models, excluding internal structures and routines that are not relevant for the design of a scenario. This information is then encoded in an intermediate representation (IR).
3. optimises the IR by solving model hierarchies and removing redundancies.
4. Generates Python code that describes the models as Airflow visual blocks. This output can then be moved to the Airflow project folder for integration.

Concretely, this pipeline works as follows. The script `splash.sh` can be executed by passing the IoD-Sim project directory as an argument. The program then searches for any relevant C++ source code files in it. This process is eased by the ns-3 convention: models have the suffix `-model.cc`, `-manager.cc`,

`-mac.cc`, and `-application.cc` in their filenames. To this end, other files are filtered out to optimise parsing operations and to prevent the exposure the simulator's internal structures. For each file found, the `clang` command is used to analyse the source code and solve any include directives needed by the preprocessor. Finally, the output is an AST, which is encoded in an optimised binary file readable only through `clang`'s application programming interfaces (APIs). The file extension is named pre-compiled header (PCH). The PCH file is then passed into `splash` core executable. This program relies on `cxxopts` library to behave like an interactive command-line application, on `boost-json` to serialise C++ data structures in JSON, and on `libclang` to read the AST. The application requires the PCH file path as input with the output directory path in order to store the IRs. These IR files are encoded into JSON to ensure software interoperability and readability. Once the command-line program is executed, the entire translation unit of the AST is scanned in order to lookup for any model used in the simulator. A custom tree-traversal algorithm is used to optimise the parse time. It works as a hybrid implementation of the classical *Breadth-first* and *Depth-first* search algorithms. A high-level representation of the translation unit is given in Figure 3.14. The key feature of this approach is the speed up introduced by the algorithm. In fact, it first traverses the tree using *Depth-first* to find the depth at which one or more `ns3::TypeId` can be found, and then uses *Breadth-first* to analyse each model at the same depth. The same strategy is applied to extract all the attributes relevant to the simulator model. Each model is represented and exported into a JSON file having the following structure: the name of the parent model, the model name, and a list of attributes, each one described by a name, an optional description, and the ns-3 data type that characterises it. Once the entire model hierarchy is solved and optimised, the attributes are copied from parent to children, if any. Then, a code generator is executed to create the visual blocks for the editor GUI. Each block name reflects the model's one and the attributes are considered as block input parameters. The generated Python code is interpreted by the GUI to display a visual block with the model name as its title and model attributes as its inputs, as illustrated in Figure 3.15.

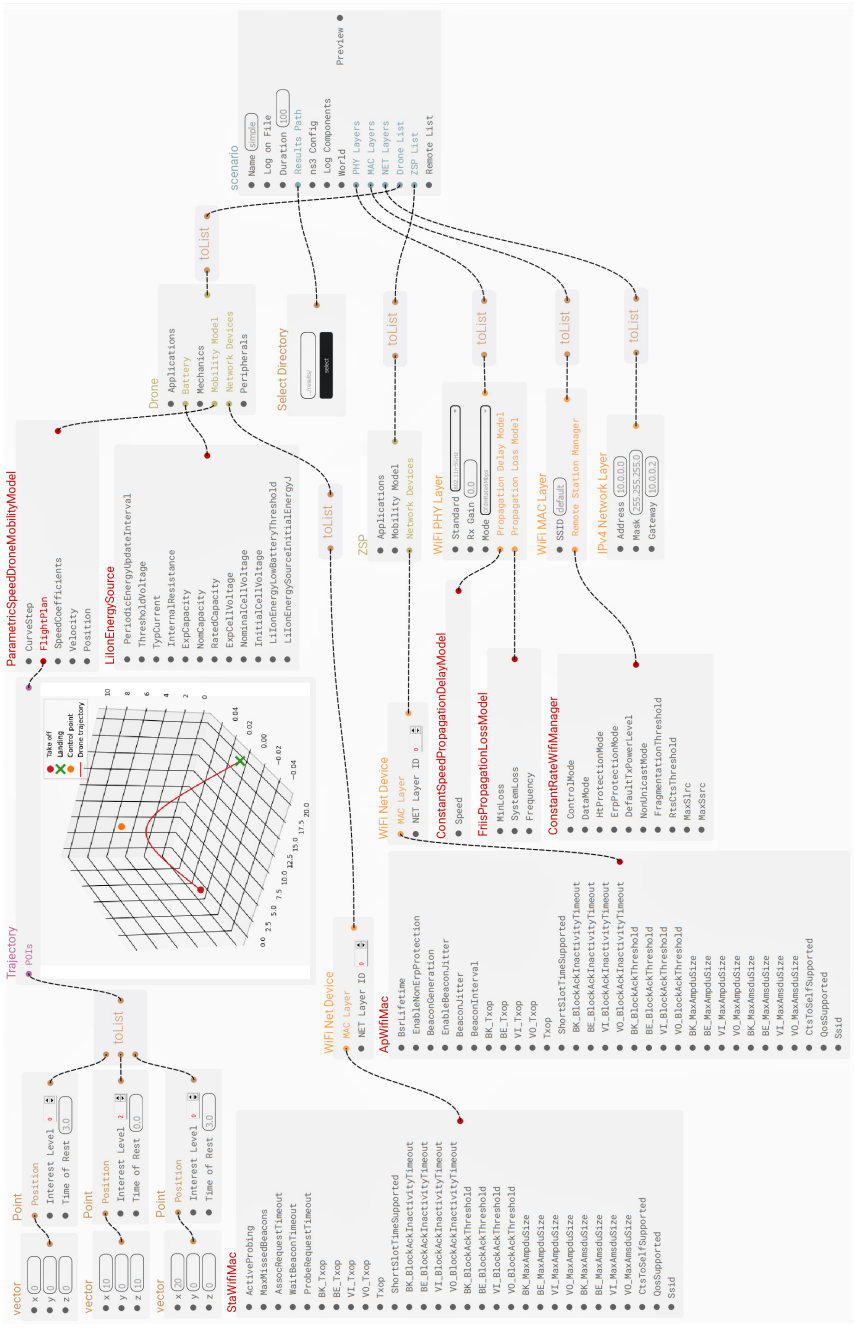


Figure 3.16: A simple scenario with one drone and ZSP designed from scratch in Airflow.

Graphical user interface

The Airflow GUI, shown in Figure 3.15, is based on the open-source engine named Ryven[7], which is a dynamic runtime, *flow-based* visual programming environment for Python scripts. It offers: (i) a central rendering view to place blocks and link them together, (ii) a settings area to customise options, (iii) a variable management section to include and store data that can be integrated with the flow, and (iv) a console to report errors. Ryven includes additional features to optionally debug internal routines with the help of console messages. Moreover, thanks to its modular design, it allows blocks generated by Splash to be aggregated into *packages*. Ryven has been deeply extended to inter-operate with IoD-Sim, especially for its compatibility with the *scenario configuration interface*. The user interface is organised into the following components:

1. A menu bar at the top of the GUI window.
2. A *Console* on the left in order to monitor errors and messages coming from Airflow or IoD-Sim. Informative messages are reported in blue, while errors are displayed in red.
3. A central workspace to design the scenario by placing blocks and connecting them together.
4. A settings panel on the right.

The menu bar is divided into three categories: with `File` it is possible to import Airflow packages to extend the user experience with third-party visual blocks. Moreover, it provides features to save the project or export it as an IoD-Sim configuration file. `View` offers graphical options, such as changing the theme, making a screenshot of the project, and tuning performance parameters. Finally, `Debugging` enables technical features to ease troubleshooting of the program, such as increasing verbosity level on the *Console*. The central workspace is the canvas where blocks and links are placed by the user to design a scenario. A block, as depicted in Figure 3.15, consists of a set of inputs and outputs. Each input and output can be connected to other outputs and inputs of other blocks, in order to create a tree. The root block is named *Scenario*. Each block has a different meaning and function. As a general overview, blocks can be divided

7 https://ryven.org/

into the following categories: operators, helpers, and IoD-Sim models. Operators are built-in blocks that can be used to work with values, constants, and data structures. Instead, helpers are special blocks that ease the configuration of a scenario, i.e., entities, Wi-Fi, and LTE configuration blocks. Usually, blocks provide a single output without a label. This output delivers the information of the block, along with all its inputs, to the next connected block. Blocks can be added to the workspace by a specific menu that is shown by clicking with the right mouse button. Moreover, each block can be right-clicked to show its contextual menu that can be used (i) to remove it, (ii) to refresh it (and hence to read all its inputs again,) and (iii) to use some particular features available in certain blocks. For instance, `toList` offers some additional controls to add or remove inputs. In the settings panel, it is possible to set the IoD-Sim path in order to enable interoperability features, such as checking the scenario configuration for errors, or running the scenario and reporting the status on the *Console*. These features can be used by clicking on the `Build` and `Run` buttons, respectively. Finally, a variable manager can be used to create, store, and reference values by their respective labels on the workspace. This allows to reduce redundancy and to make the block tree more compact.

3.6 The road ahead

The large-scale adoption of emerging 6G and integrated T/NTN services should be evaluated after a prototyping phase that can be time-consuming and may require unfeasible costs. To tackle this problem, simulators are an essential tool to facilitate the testing phase and state the readiness for real-world exploitation. At the same time, simulators can be a learning tool for young professionals, engineering students, and researchers to improve their knowledge and explore scenarios never considered before. In this landscape, IoD-Sim empowers users to explore a vast array of research questions and evaluate emerging applications and services. The simulator integrates some of the features discussed from the state of the art and it offers a configurable and extensible platform for ease of use and continuous improvement. Particularly, the simulator presents itself as a thorough and user-welcoming tool that can be used to evaluate the many facets of T/NTN scenarios, including trajectory design of flying network nodes, networking functionalities, mechanical characteristics, and data analytics. Nevertheless, IoD-Sim has been created as a modular tool that can be updated and upgraded as

needed. Moreover, a predictable build environment is used to ease the installation due to its dependencies that require careful set up and knowledge about the underlying simulator, libraries, and compilers. Even though IoD-Sim is a reliable solution, in the future more efforts will be focused on the improvement of the entire project, especially along the following research and development lines: (i) extend the support to design scenarios using technologies such as MAVLink, satellite communications, and 5G NR; (ii) speed-up Splash compilation with the use of parallel multiprocessing and optimised algorithms; (iii) develop interactive visual blocks to preview or design more accurate simulations in less time; (iv) improve the overall user experience of the visual editor; (v) allow the employment of multi-processing systems and clusters; (vi) directly compare the performance and the features with other network simulation platforms.

Regarding IRSs, these 6G emerging devices allow to control the environmental conditions of the radio channel, thus leading to noticeable improvements in communication quality. In this regard, the simulator enables the development of future communication systems where this two technologies can be integrated. Furthermore, the proposed solution provides general schedulers for IRS patches to maximise its utility and flexibility. Future endeavours for the simulation of 6G-enabled IRS-assisted communications should focus on: (i) accurate power consumption model of IRSs; (ii) performance assessment of this module with mmWave simulations, to assess the performance of systems that go beyond classical sub-6 GHz communications; (iii) comparison of these new emerging systems with AF solutions, employed in 5G network backhaul; (iv) real-time attitude controls for the IRS in order to follow a target mobile node; (v) enhanced configurators with feedback loop that choose to serve nodes depending on their channel conditions; (vi) channel model aware of the specific material obstructing the LoS, in order to choose the most suitable K-factor and path loss coefficient; finally, (vii) channel model able to consider multiple reflections made by the alignment of two or more IRSs, in compliance to the current standardisation efforts [146].

As for high-altitude and space communications, future implementation efforts should be focused on the modelling of (i) non-stationary satellite orbits, (ii) HAP and satellite power consumption, (iii) MAC layer protocols that take into account NTN propagation delays, and (iv) an Integrated T/NTN end-to-end communication stack for a comprehensive 6G simulation platform. Finally, the

birth of a thriving and empowering community on open-source collaboration platforms will be crucial in assessing the future development efforts of this work.

4 Experimental results from selected use cases

As introduced in Chapter 1, there is the need to extensively test the integration of T/NTN scenarios. To this end, a comprehensive selection of numerical examples and use cases are investigated and analysed hereby. While IoD-Sim has been tested and used for (i) general data communications, (ii) IRS-assisted UAVs scenarios, and (iii) HAP-to-satellite communications, MATLAB and Casadi were preferred to design, optimise, and assess T/NTN-based WPT applications.

4.1 IoD-Sim Simulation Campaign

Catering to user convenience, IoD-Sim, which is introduced and deeply analysed in Chapter 3, offers (i) pre-built scenarios to streamline research by leveraging sample use cases spanning from LTE to 6G and Integrated T/NTN; (ii) extensive customisation options to accommodate research across diverse fields, thus fostering user adoption while simultaneously facilitating the exploration of emerging applications and services; (iii) integration with conventional integrated development environments, such as Visual Studio Code, to ensure efficient workflow and minimal learning curve through simple executable tasks; (iv) comprehensive log and data generation for full insight into simulation behaviour, with detailed text traces capturing key events; and finally (v) a modular design to seamlessly extend the tool's capabilities in order to address diverse research needs by incorporating new mobility, communication, and application models.

An extensive and diversified simulation campaign is carried out to validate its manifold functionalities. To this end, three scenarios are conceived to evaluate different configurations of network topologies, communication technologies, drones' equipment, and software applications. Thoughtful insights are derived by analysing the obtained results in terms of SINR, throughput, power consumption, latency, and packet loss ratio (PLR). Moreover, a performance analysis is conducted to assess the computational load and its scalability.

Firstly, the discussion explains how the simulation can be designed. Secondly, three different scenarios with increasing complexity are presented. In particular,

the first scenario discusses the use case of telemetry with a few drones flying in a RoI, which follow customised trajectories while gathering data. The purpose of this scenario is to demonstrate that it is possible to monitor one or more variables with on-board sensors, while estimating the energy consumption associated with flight dynamics.

The second scenario has a wider perspective since it focuses on surveying and monitoring activities, further completed with the acquisition of multimedia signals by each drone. The possible applications include several real-world use cases in the fields of civil engineering, smart agriculture, or environmental monitoring, e.g., coastal erosion and other slow phenomena. In fact, in this scenario, drones are on a mission in neighbouring areas since it is assumed that the information of interest needs to be contextualised, i.e., must be gathered at the same time. Furthermore, this case investigates the possibilities enabled by different data storage capabilities of drones. Also, the offloading functionality of the acquired data avoids the overload / saturation of on-board available resources. Once data is gathered, they can be involved in offline post-processing, evaluation, and analysis.

The third scenario has been specifically designed to be the reference benchmark for NTN applications. It is set in the context of smart cities and involves clusters of low-power IoT sensors. This scenario models real-world applications and, hence, shadowing and path loss phenomena are included, thanks to the adoption of propagation models that are influenced by the presence of buildings. In order to guarantee a reliable communication, drones are in charge of relaying traffic to ensure coverage to all sensors in the city.

4.1.1 Scenario design

Airflow represents the foremost application for visual scenario development. To better understand how to design simulations, a simple configuration set up is provided hereby. The envisioned scenario considers a drone that follows an arc-like trajectory and communicates telemetry to a ZSP by means of Wi-Fi. Specifically, the drone acts as a *station* and the ZSP as an *access point*. The entire configuration is depicted in Figure 3.16, where all the visual components, encompassed in the Airflow workspace, are properly set up and linked together. Starting from the right, the block `Scenario` glues some configuration input values, e.g., `Name` and `Duration`, with more complex components, such as (i) `PHY/MAC/NET Layers`, (ii) `Drone List`, and (iii) `ZSP List`.

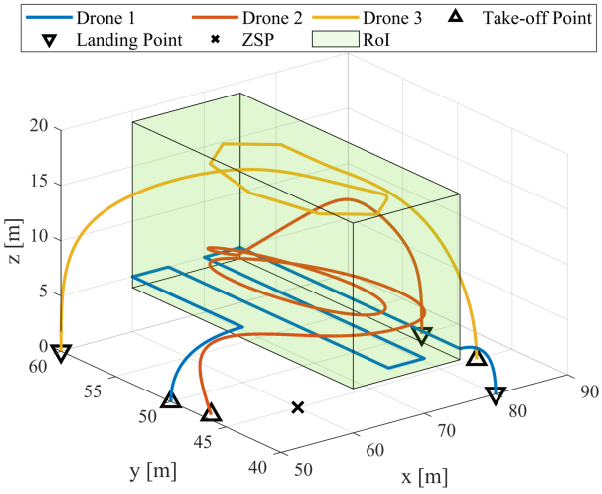


Figure 4.1: Scenario #1.

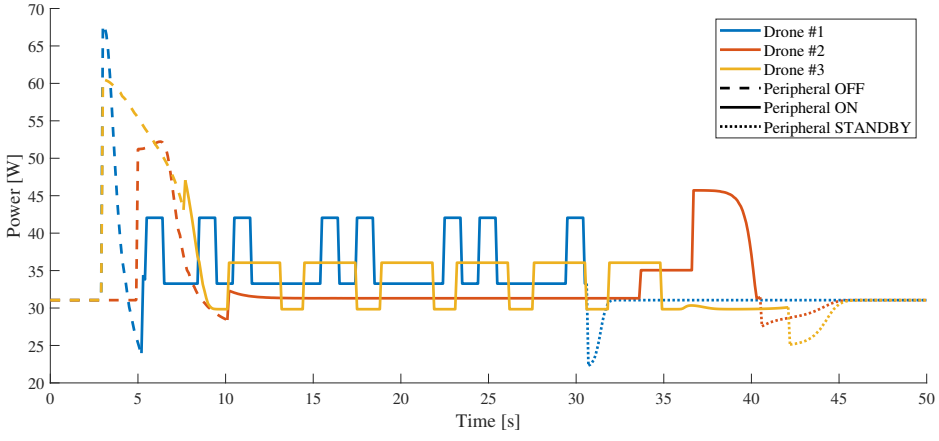
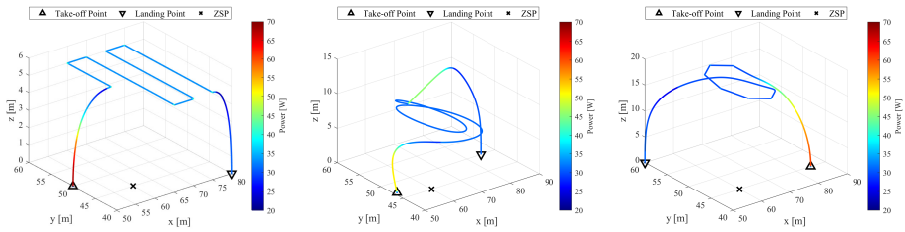


Figure 4.2: Power consumption and peripheral state for each drone, in the first scenario.



(a) Drone #1. **(b)** Drone #2. **(c)** Drone #3.

Figure 4.3: Drones' trajectories with their power consumption, in the first scenario.

In particular, the communication layers are configured to implement the Wi-Fi stack. The `WiFi PHY Layer` object defines the PHY layer to be used with particular propagation and loss models. The `WiFi MAC Layer`, instead, specifies the SSID of the network and the Wi-Fi Manager object that handles the MAC control plane. Further, the `IPv4 Network Layer` determines the address and mask of the overlying network. Both `Drone List` and `ZSP List` properties are connected to the simulated entities, namely `Drone` and `ZSP`. These two components share different properties such as `Applications`, `Mobility Model`, and `Network Devices`. However, the `Drone` block is also characterised by its unique features, i.e., `Peripherals`, `Mechanics`, and `Battery`. In this configuration, the `ConstantPositionMobilityModel` allows placing the `ZSP` at a fixed location, while the `ParametricSpeedMobilityModel` is employed to define the drone trajectory. In this regard, the `Trajectory` component, linked to the `FlightPlan` property of the mobility model, facilitates the design of the desired path. The `Network Devices` property of both drone and ZSP is linked to a `WiFi Net Device` block. While `StaWifiMac` characterises the device of the former, `ApWifiMac` is associated with the latter. Finally, a `LiIonEnergySource` defines the power supply of the drone.

The development strategy discussed above represents the common ground for the design of the following three scenarios.

4.1.2 Drone telemetry

The first scenario, as depicted in Figure 4.1, envisions three drones with the same mechanical characteristics, all equipped with an IMU. In this scenario, drones are flying in the same RoI at a constant speed, following different trajectories.

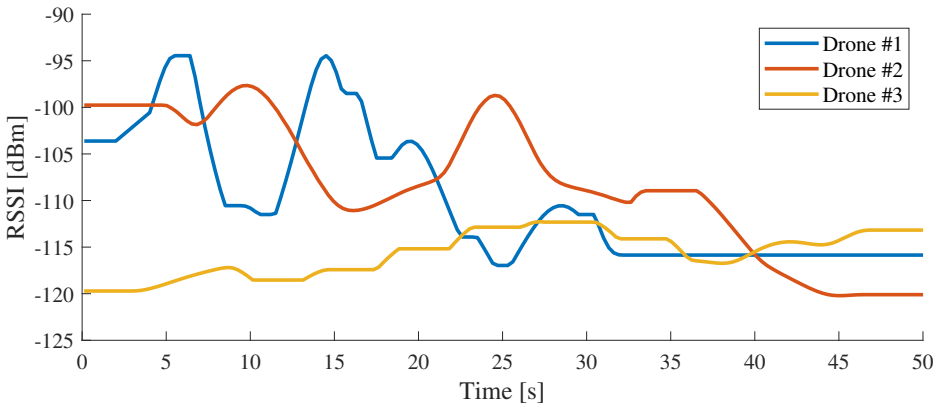


Figure 4.4: Measurement of the RSSI of each drone by the ZSP in the first scenario.

Moreover, a ZSP is deployed on the ground. The latter is released in $[60\ 45]^T$, which continuously monitors drones' operations by acquiring telemetry through Wi-Fi.

UAVs' trajectories are based on the object `ParametricSpeedMobilityModel`, which is configured to guarantee a constant speed of 5 m/s, 3 m/s, and 4 m/s, respectively. They are also equipped with IMUs, which are generic drone peripherals that provide basic telemetry data to the ZSP thanks to a dedicated application, as mentioned in Section 3.4.7. It is worth specifying that drones' IMUs have different power consumption, i.e., 12 W, 5 W, and 6 W.

The outcome of the simulation is hereby discussed. Figures 4.2 and 4.3 depict the power consumption trend wrt. time and trajectories. In the former, the three curves share an initial peak which corresponds to the energy required to take off. Indeed, acquiring altitude requires more power than flying along the xy plane, as highlighted. This phenomenon is further remarked in Drone #2 landing manoeuvrer. It includes a little parabola that yields a peak in the last part of the associated curve of Figure 4.2, which is also present in Figure 4.3 (b). After $\sim$ 10 s, the drones reach and almost maintain a target altitude. The corresponding power consumption, for Drones #1 and #3, is characterised by peaks due to hovering over the interest points for 1 s and 3 s, respectively. These points are identified by the vertices of the snake-like and octagon-shaped trajectories. Instead, this phenomenon is not present on Drone #2, since its trajectory describes a continuous curve. When the drones enter the RoI, the

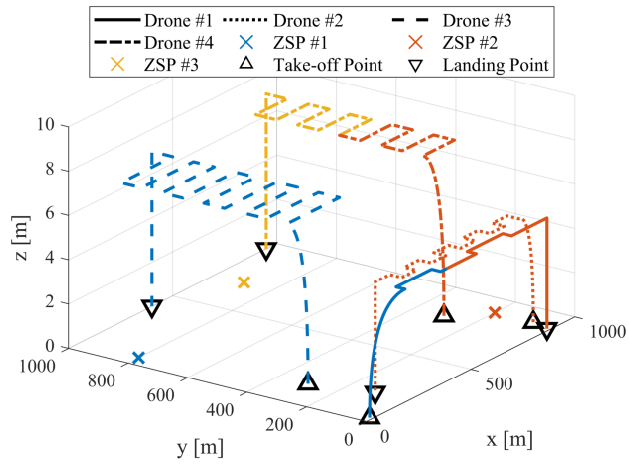


Figure 4.5: Trajectory design and eNB attachment for each drone, in the second scenario.

peripherals become active, and hence the IMUs power contribution is non-zero. Spikes can be noticed in the curves of Figure 4.2, especially in Drones #1 and #2, since they are equipped with two more energy-demanding peripherals. As soon as drones exit such a region, the peripherals go into standby mode, which preserves energy.

Figure 4.4 illustrates the measured received signal strength indicator (RSSI)[8] of each drone during the mission. Measurements are carried out by the ZSP. In general, such values can be conceived as an assessment of ranging operations carried out by a single node when its position is fixed. From this Figure, it clearly emerges that, on average, Drones #1 and #2 maintain a better signal quality wrt. the UAV #3. Obviously, the higher altitude, and hence the greater distance from the ZSP, worsens the communication quality due to the Friis propagation loss employed to model the fading effects in this scenario.

4.1.3 Multimedia signals acquisition

The second scenario is depicted in Figure 4.5. A swarm of four drones is in charge of acquiring multimedia signals in an operating area that is 10^6 m^2 wide. Acquired data are stored on-board and off-loaded to a remote server as soon as

8 The received signal power expressed in dBm.

the drone is able to communicate with a ground infrastructure. The latter, which allows data upload, is composed of three ZSPs, also referred to as eNBs, that are deployed on the ground in three different locations: $[50\ 800]^T$, $[900\ 200]^T$, and $[700\ 900]^T$, respectively. All the entities involved in the mission, which lasts 250 s, are equipped with LTE interfaces. Drones follow snake-like trajectories, each different from the other in terms of amplitude and frequency. Nevertheless, they adopt the same mobility model with a constant acceleration of $4\ m/s^2$ and a maximum velocity between 15 and 20 m/s. Each drone is equipped with cameras that operate at different data rates, 2 Mbps, 1.6 Mbps, 1.3 Mbps, and 1 Mbps, respectively. The communication between each UAV and the remote server is handled by *Generic Traffic Applications* (see Section 3.4.7,) with a payload size of 1024 bytes and a TCP Max Segment Size of 1380 bytes.

In the same figure, it can be further observed the attachment of the drones to the ZSPs. Throughout the mission, Drones #2 and #3 remain linked to the same eNB, i.e., ZSP #2 and #1. On the other hand, UAV #1 and #4 perform a handover procedure which changes the reference ZSP from #1 to #2 and from #2 to #3, respectively. It is worth noting that, despite Drone #1 takes off in the same area where Drone #2 lands, they are not attached to the same ZSP. Indeed, even if the two trajectories share the same direction, they have opposite verse: while one approaches an eNB, as the mission goes by, the other flies away from the ZSP without really getting closer to another one.

Figure 4.6 shows the throughput[9] for each drone on the associated ZSP over time. It is shown that UAV #1 experiences an average data rate of $\sim$ 1 Mbps until the handover procedure takes place, which increases this value by ~50%. Similarly, the average throughput of Drone #4 is also ameliorated since it increases from $\sim$ 800 kbps to $\sim$ 1.1 Mbps. It is worth noting that there exists a pattern correspondence between the throughput and the occupied storage curves (see Figure 4.7.) This is particularly evident for Drones #3 and #4: when the occupied memory lowers and goes to zero, the data rate decreases as well, and tends to zero. Indeed, for the information causality principle, it is not possible that a larger amount of information is transmitted wrt. the stored one. Notice that this happens as long as the acquisition rate remains lower or equal to the channel capacity which, for instance, is not the case of Drone #1.

9 Based on the evaluation of successful frame reception at each second.

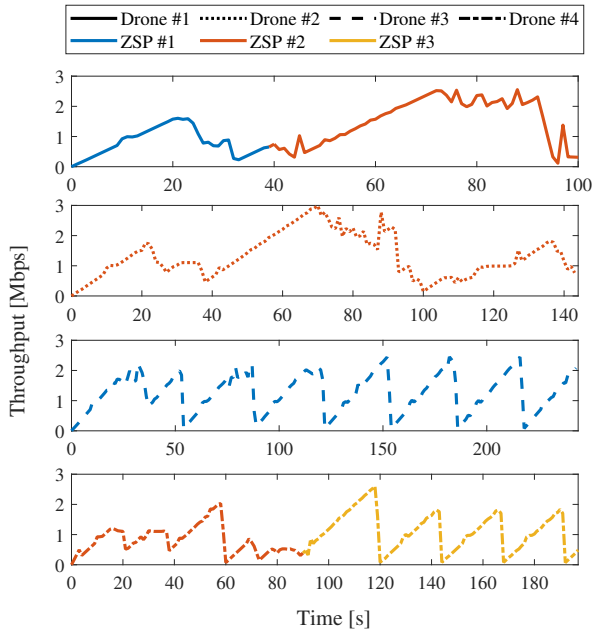


Figure 4.6: Drones' throughput, in the second scenario.

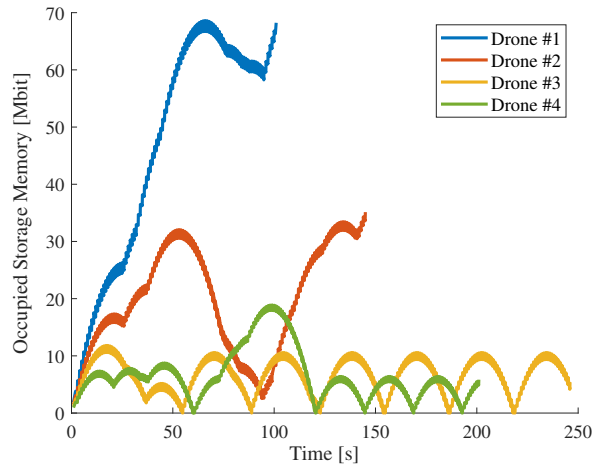


Figure 4.7: Memory occupancy for each drone, in the second scenario.

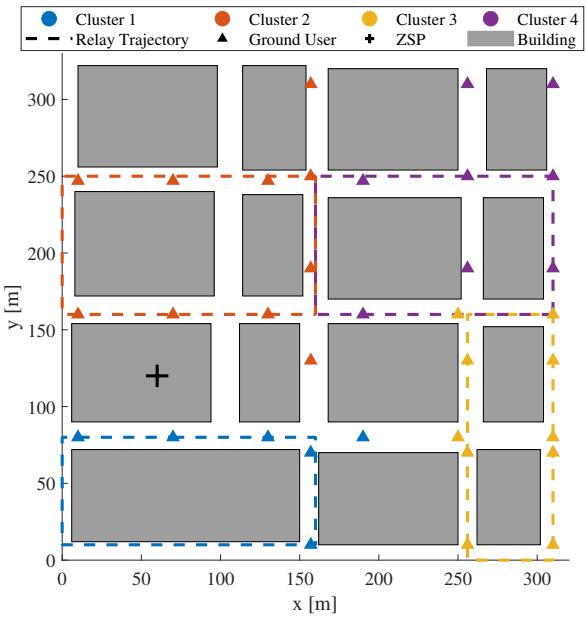


Figure 4.8: Scenario #3 simulation environment.

4.1.4 Flying base stations in smart cities

The third scenario reproduces a smart city context, in which drones are in charge of relaying traffic coming from clusters of GUs, using Wi-Fi technology, to a remote server over the Internet, through LTE. In this regard, the presence of buildings plays an important role both in trajectory design and in fading phenomena. The envisioned scenario is designed starting from the map of an urban area in the neighbourhood of the Central Station of Bari, Puglia, Italy.

As shown in Figure 4.8, four GU clusters of different size are present on the ground. Each of them is served by a drone, which relays the traffic by means of the NAT application discussed in Section 3.4.7. The entire simulation lasts 180 s and employs the `ns3::HybridBuildingsPropagationLossModel` to take into account the fading caused by the presence of buildings. Moreover, each building is characterised by a window per room and is assumed to be built with concrete walls. The Wi-Fi stack has been configured based on the 802.11ax standard operating at 2.4 GHz and is controlled by the `ns3::IdealWifiManager`, which

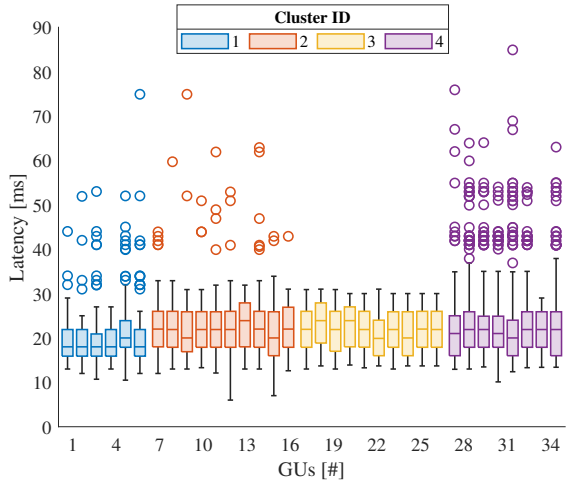


Figure 4.9: GUs application latency of link combined by Wi-Fi, relay drone, and LTE.

allows to keep track of the SINR, expressed as[10]

$$\Gamma = \frac{gP}{\eta k_B BL + I} \tag{4.1}$$

where P is the transmission signal power of the station/access point in W, g is an additional gain due to receiver antenna diversity in case of an additive white Gaussian noise (AWGN) channel, η describes the equivalent system noise temperature for both antenna and receiver noise (assuming 290 K), B is the channel bandwidth in MHz, L is the receiver noise figure[11] due to non-idealities of the receiver (assuming 7 dB), and I is the interference[12] on the channel band measured in W. Thanks to this mechanism, it is possible to always choose the best transmission mode to be used, i.e., a combination of modulation, coding scheme, and data rate.

As for the network level, each cluster is connected to its relay according to the $10.[1-4].0.0/24$ network address range, while LTE uses 7.0.0.0/8. Drones' trajectories are designed following the layout of the streets in order to minimise

10 Implemented by the ns-3 Wi-Fi module in `ns3::InterferenceHelper::CalculateSnr()`.
11 Defined by the ns-3 Wi-Fi module in `ns3::WifiPhy::GetTypeId()`.
12 Evaluated by `ns3::InterferenceHelper::CalculateNoiseInterferenceW()`.

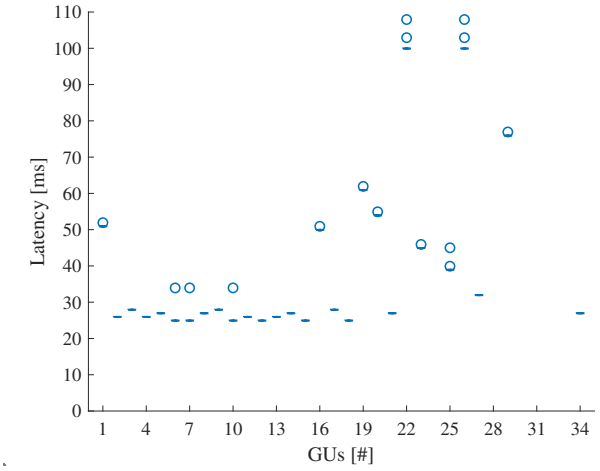


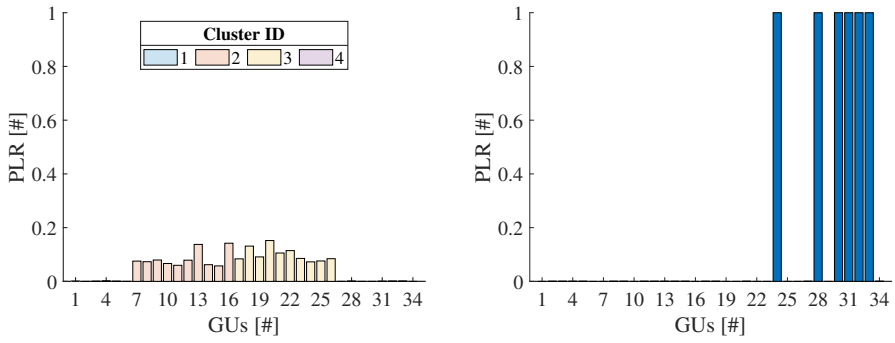
Figure 4.10: GUs application latency over LTE-only link.

the shadowing effects and maximise the LoS with the GUs. Furthermore, the path also maximises energy efficiency as the translation in the xy plane is less costly when compared to changes of altitude. At each angle of the trajectory, the drones pause for 1 s in order to simulate an accurate 90 degrees yaw.

Accordingly, each relay drone flies at a constant altitude of 50 m at 5 m/s. Drones are equipped with the `ns3::NatApplication`, which implements a simple Port-based NAT strategy for UDP communications. Each GU has a constant position and is equipped with a simple `ns3::UdpEchoClientApplication`, which periodically sends a packet of 1024 bytes to the remote address `200.0.0.1` at port `1337` with a frequency of 10 Hz. Each packet is equipped with an application header that reports an incremental sequence number and the time of creation. Finally, the remote has a `ns3::DroneServerApplication`, which records via log messages the received packets. The only ZSP, located at $[60, 120, 40]^T$, provides LTE access to the drones, thus allowing the communication with the remote host.

In order to allow the analysis of the latency[13] for the entire simulation, box plots are proposed in Figures 4.9 and 4.10, which clearly show the advantage brought by the relay activity by the drones. These box plots summarise the

13 The latency has been evaluated at the application layer of each transmitter and receiver host.



(a) Wi-Fi & LTE **(b)** LTE

Figure 4.11: GUs application PLR for Scenario #3.

Scenario #	Events [#]	real time [s]	Sim. Time [s]
1	57,437	9	50
2	18,226,323	761	250
3 LTE	37,178,812	4,620	180
3 Wi-Fi & LTE	28,903,306	2,858	180

Table 4.1: Comparison of the total number of events, the real time taken to execute, and the simulated time of each scenario.

latency distribution of GUs across the different clusters color-coded by the Cluster ID of reference, reported at the top of Figure 4.9. In the same figure it is possible to appreciate multiple distributions along their variability, particularly for understanding the spread, central tendency, and presence of outliers. The distributions are placed along the x axis and each distribution is shown as a box, which size reports the interquartile range from the 25^{th} to the 75^{th} percentile of the latency samples. The line inside the box marks the median latency for that distribution, thus providing a measure of central tendency. The vertical black lines extending from the box indicate the range of data within 1.5 times the interquartile range, both at the lower (first) and upper (third) quartiles. These lines, better known as whiskers, illustrate the main spread of the data, excluding outliers. Finally, the circles report the outliers, data points that significantly deviate from the typical range. These may highlight specific conditions affecting latency, such as retransmission of frames due to worse path loss.

In the relay case (Figure 4.9,) all the GUs experience an average latency of $\sim$ 25 ms, a result that is achieved also thanks to the proposed trajectory design. On the contrary, in absence of relay drones (see Figure 4.10,) while the GUs that are closer to the ZSP are affected by a latency similar to the previous case, the farther ones register a significant delay, which inevitably compromises the reliability of the link and, hence, the QoS. Nevertheless, this comes with a trade-off as highlighted in Figure 4.11, which shows the PLR[14] in both cases. In the former, all nodes are able to transmit data to the remote, but with a loss ratio of $\sim$10% for the cluster #2 and #3. It is worth noting that this result can be further improved by properly optimizing the trajectory design to target the desired trade-off. In the latter, instead, six nodes have 100% PLR, which means that there is no exchange of data.

4.1.5 Performance evaluations

To evaluate the performance of the simulator, and hence its scalability, the performance metrics of the simulated scenarios are analysed and compared hereby. The runtime environment is characterised by the following hardware and software specifications: (i) Intel® Xeon® Bronze 3106 at 1.70 GHz with 16 cores and no hyper-threading, (ii) RAM 92 GB DDR4 at 2666 MHz, (iii) 7200 RPM hard drives and (iv) OS Fedora 35 on LXD container [147]. It is worth specifying

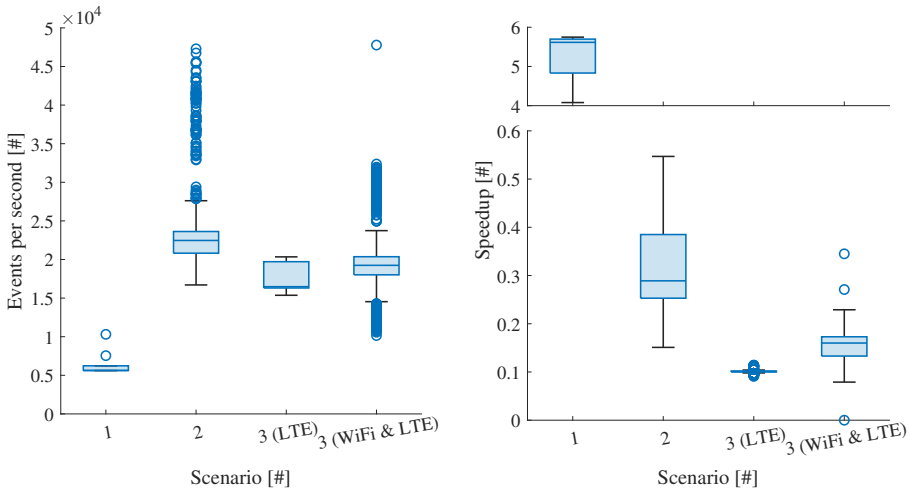
14 Evaluated at the application layer of each host in the network.

that the present assessment is made leveraging a single-core configuration, although multi-processing support is available. To fairly compare the simulations, two metrics are selected. The former takes into account the number of events processed per second for each simulation, thus providing an insight related to the scenario complexity. The latter considers the ratio between the simulated time and the real time, thus further addressing the complexity of the designed missions. Moreover, Table 4.1 summarises the total number of events, the time taken to simulate (real time,) and the simulated time of each scenario. It is worth noting that all scenarios are constructed differently and hence are difficult to compare. However, some clear indications can be derived from the following analysis. Indeed, Figure 4.12 shows that in Scenario #1 the employment of Wi-Fi technology slows the number of events processed per second, which means that the complexity is higher. On the contrary, the adoption of LTE (either mixed with Wi-Fi) reduces the overall computational complexity. However, in the first case (Scenario #1) the speed-up is greater wrt. the second case (remaining scenarios): this is due to the fact that the number of generated events is way lower. This is particularly evident in Scenario #3, where the simulation time and the number of GUs are the same, as shown in Table 4.1. Overall, even if the number of actors increases when drone relays are employed (LTE & Wi-Fi,) the lower number of events generated guarantees better performance.

Moreover, in order to further investigate the simulator performance and derive more insights regarding the required resources to run a computationally complex scenario, the following final evaluation is provided. A square area is partitioned into four quadrants, each one with a central drone relay. According to a uniform random distribution, a set of GUs is generated and symmetrically placed into the four regions wrt. the BS, which is placed in the center of the area. The number of considered GUs is then increased in accordance with the power of 2. Given this scenario, execution time and the maximum reserved memory are considered reference metrics and are reported in Figure 4.13. As it can be deduced, both exponentially grow with an increasing number of GUs, as confirmed by the regression performed on the obtained data. Indeed, it is possible to predict the time and memory required for a specific simulation as

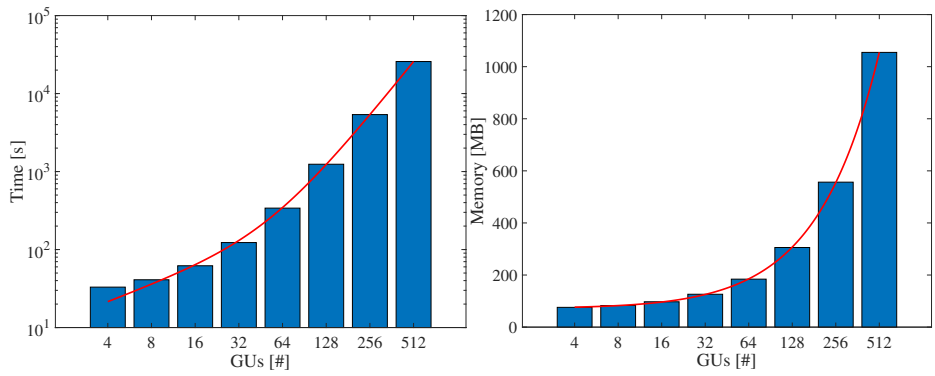
$$ae^{bx} + ce^{dx}, \tag{4.2}$$

where x is the number of GUs, and a, b, c, d are the fitted coefficients provided in Table 4.2.



(a) Number of events. **(b)** Simulation speed-up time.

Figure 4.12: Performance evaluation of the different simulated scenarios.



(a) Execution time. **(b)** Maximum reserved memory.

Figure 4.13: Performance evaluation of the simulator wrt. the number of GUs.

Metric	a	b	c	d
Time	117.2	1.197	87.45	3.961
Memory	64.35	-0.05182	85.62	1.717

Table 4.2: Coefficients of Equation 4.2.

Parameter	Value	Parameter	Value
Simulated Area	400×400 [m^2]	`KMax`	10 [dB]
`KMin`	6 [dB]	`KNlos`	0 [dB]
`AlphaLoss`	{3, 4} [#]	`NoIrsLink`	`false`
`OutageProbability`	0.01 [#]	`NoDirectLink`	`false`
`RotoAxis`	`["X_AXIS"]`	`RotoAngles`	[180.0] [deg]
`PruX`, `PruY`	0.01 m	UE, eNB Power	24, 49 [dBm]

Table 4.3: Parameter settings.

4.2 Evaluation of communications assisted by Intelligent Reflective Surfaces

Three different scenarios are designed and assessed hereby to validate the features related to the IRSs, adopting the parameters reported in Table 4.3, if not otherwise specified. Furthermore, all the scenarios are tested using one communication technology only, i.e., LTE, with a fixed bandwidth of 5 MHz and 25 resource blocks. In particular, all these scenarios consider an eNB, acting as a ZSP / BS, and a set of UEs, acting as nodes / GUs, that experience different SINR levels due to path loss and LoS conditions. To this end, IRS-equipped UAVs are employed to assist the communication links. The overall performance achieved through the aid of the IRS are compared, analysed, and discussed via several KPIs, such as REM, SINR, maximum achievable rate, and average throughput. The proposal guarantees ease of use to reliably simulate advanced T/NTN systems, with the goal of thoroughly testing new proposals and applications, especially related to their employment in densely populated urban environments. The numerical results obtained from the proposed IRS extension indicate that the presence of IRS-equipped drones enhances the channel quality of the GUs. Moreover, the possibility to organise the IRS in patches is an effective solution to uniformly assist multiple nodes. This in turn demonstrates the unique potential of the simulation platform to assess and prototype complex IRS-enabled NTNs.

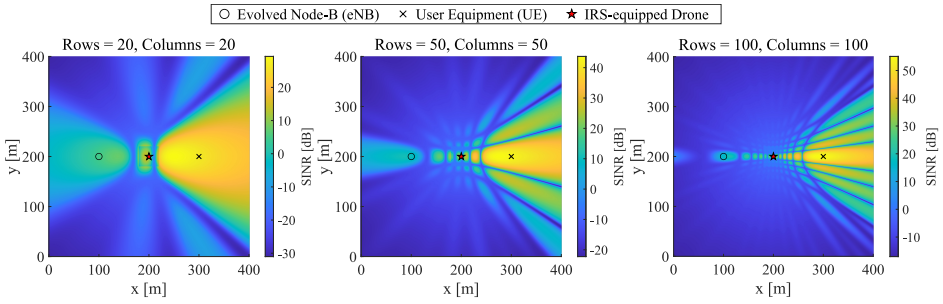


Figure 4.14: Comparison among downlink REMs for different IRS sizes.

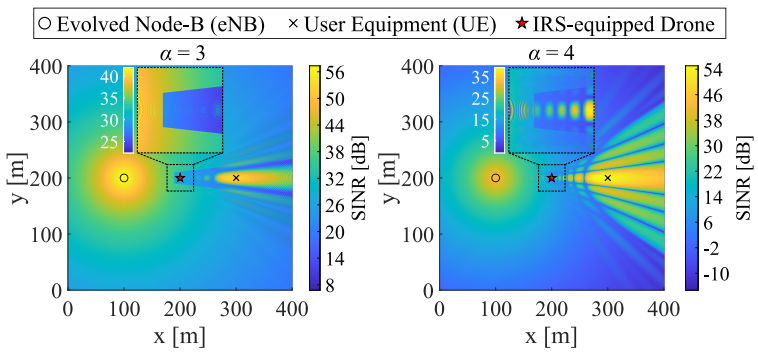


Figure 4.15: Downlink REMs for different attenuation factors.

4.2.1 eNB–UE obstructed communications

The first Scenario considers an area with a building of $20 \times 20 \times 25\,\mathrm{m}^3$, placed at $[200, 200, 0]^\mathsf{T}$, that obstructs the direct link between an eNB and a UE, located at $[100, 200, 30]^\mathsf{T}$ and $[300, 200, 0]^\mathsf{T}$, respectively. To support the communication between these two nodes, an IRS-equipped UAV hovers 50 m over the building, thus re-establishing the LoS. The overall Scenario is depicted in Figure 4.14, which also illustrates the downlink REM[15] at the ground level with a resolution of 16 samples/m^2. It is worth specifying that, for the sake of the analysis, the contribution of the eNB-UE direct link is temporarily neglected. The radiation fingerprints exhibit two main properties: as the number of PRUs increases (i) the main lobe pointing at the target node becomes narrower and (ii) the perceived SINR increases as well. However, these benefits come at the price of a larger IRS, which implies higher costs and footprint.

Figure 4.15 depicts the same scenario from a different point of view. The direct link is not neglected anymore and it is considered an IRS of fixed size 100×100 elements, with a varying attenuation factor $\alpha = \{3, 4\}$ adopted for the direct link. Clearly, this case highlights the shadowing effect due to the presence of the building, which is more evident with $\alpha = 3$, since the direct link is less attenuated. At the same time, with $\alpha = 4$, it is more noticeable a weaker shadow surrounding the building, which is its projection on the ground as a result of the IRS reflections, i.e., UAV-UE NLoS link. Moreover, it can be noticed a slight ripple effect due to fast-fading phenomena as a consequence of multipath interference.

Lastly, Figure 4.16 shows the channel conditions between the eNB and UE in terms of SINR and maximum achievable data rate in uplink. This time, multiple configurations investigate the presence and also the absence of the building, labelled with *LoS* and *NLoS*. Moreover, the total absence of the eNB-UE direct link is considered, marked as *No direct link*. In terms of SINR, illustrated on the left, the first obvious observation is that, for a given α, the LoS cases are always better than the NLoS ones. Moreover, for a low number of elements, the curves with $\alpha = 3$ start with a better SINR wrt. the ones characterised by $\alpha = 4$. However, as the IRS becomes larger, the less attenuated case in NLoS conditions, i.e., $\alpha = 3$, is characterised by a significant destructive interference

15 The REM is a uniform 2D or 3D grid of values representing the downlink SINR relative to the eNB with the strongest signal at each point.

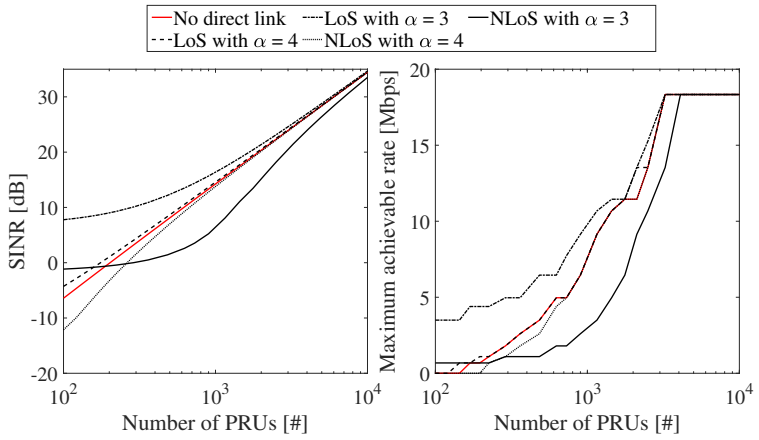


Figure 4.16: Uplink maximum achievable rate and SINR under different channel conditions.

phenomena, as it can be noticed by comparing it with the *No direct link* curve. These unwanted effects can be prevented with a proper design of the scenario geometry, i.e., the communication actors should be correctly aligned. Lastly, for a very large number of elements, all cases converge to "No direct link", since the reflected link overshadows the direct one. For what concerns the maximum achievable rate, depicted on the right, it follows a trend that is similar to the SINR. It can be observed that, as the number of PRUs grows, the curves overlap due to modulation and coding scheme (MCS) switching [148], until the rate saturates at 18.336 Mbps.

4.2.2 Enhancing crowd communications with far-away eNB

This scenario considers the presence of three clusters, each with an increasing number of UEs placed according to Table 4.4. All the UEs exchange data with an eNB with the support of an IRS-equipped UAV. The direct UE-eNB link is characterised by the path loss exponent $\alpha = 4$. The goal is to fairly serve each cluster through an IRS of 100×100 elements. To this end, the drone follows a circular trajectory of radius 150 m, at a constant speed of 10 m/s, that intersects the center of each cluster. The circumference is equally divided into three arcs, for which a suitable IRS configuration is set to serve

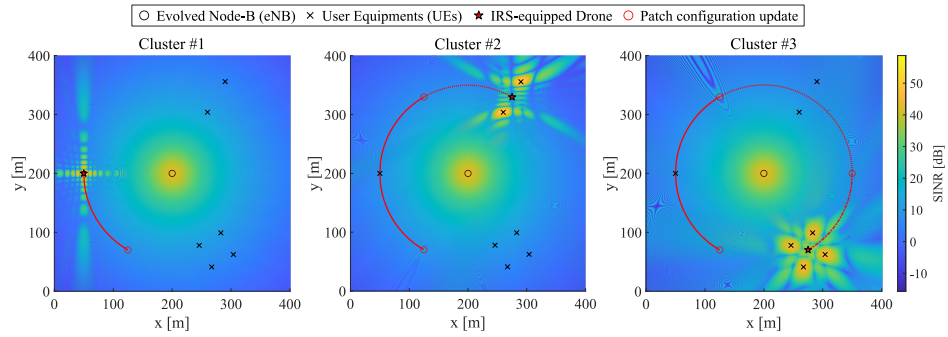
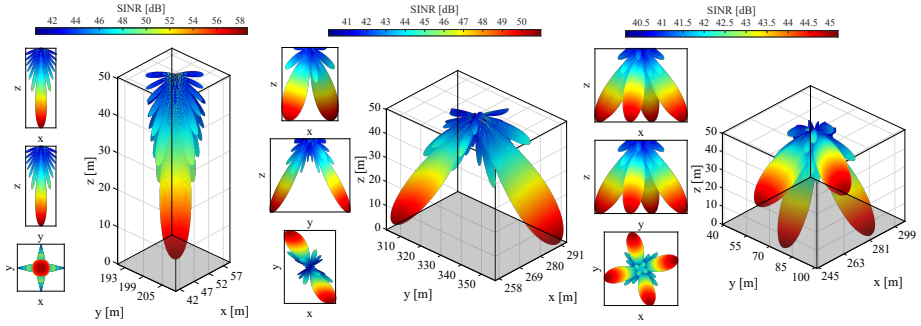


Figure 4.17: REMs taken exactly when the drone results orthogonal to each cluster.



(a) Cluster #1. (b) Cluster #2. (c) Cluster #3.

Figure 4.18: The 3D radiation patterns of the IRS serving different clusters.

the UEs of interest for ~ 31.416 s. Such trajectory is implemented through the `ns3::ParametricSpeedDroneMobilityModel` class.

The described scenario is depicted in Figure 4.17, which also shows the downlink REM at three different instants corresponding to the UAV being orthogonal to the center of each cluster. As it can be seen, the reflected signal power yields a different radiation pattern on the ground, depending on the adopted IRS configuration. As already seen in Scenario #1, this case is subject to the interference between direct and reflected links, i.e., multipath. Additionally, this effect is exacerbated by the presence of patches configured to serve different members of the same cluster, i.e., the IRS self-interference. Furthermore, the SINR is inversely proportional to the number of users to be served. This behaviour is more evident in Figure 4.18, where the signal beams produced by the IRS are depicted, with a

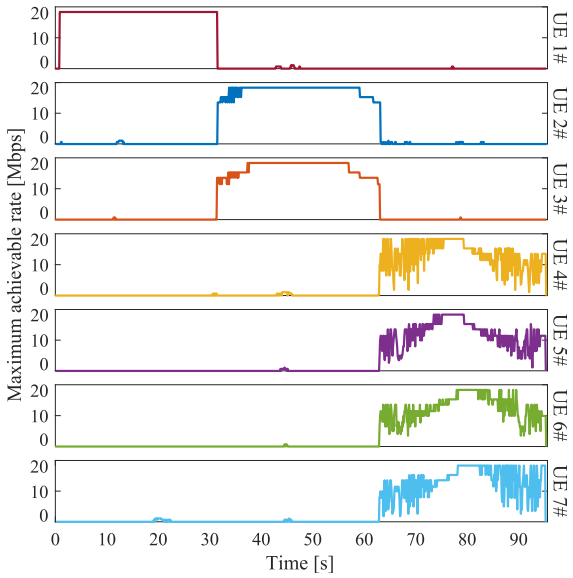


Figure 4.19: Uplink maximum achievable rate.

Cluster #	UE #	Position
1	1	$[50,\ 200,\ 0]^{\mathsf{T}}$
2	1	$[260,\ 303.923,\ 0]^{\mathsf{T}}$
	2	$[290,\ 355.885,\ 0]^{\mathsf{T}}$
3	1	$[282.765,\ 99.074,\ 0]^{\mathsf{T}}$
	2	$[303.978,\ 62.331,\ 0]^{\mathsf{T}}$
	3	$[267.235,\ 41.118,\ 0]^{\mathsf{T}}$
	4	$[246.022,\ 77.86,\ 0]^{\mathsf{T}}$

Table 4.4: UEs positions in different clusters.

peak SINR of ∼ 58.61 dB for Cluster #1, ∼ 50.98 dB for Cluster #2, and ∼ 45.19 dB for Cluster #3. Specifically, the SINR lowers since the surface is equally divided among the nodes of the target cluster. It is worth noting that, differently from the Clusters #1 and #3, the two beams depicted in Figure 4.18b are not symmetrical, as can be seen in the xz and yz projections, due to the rectangular shape of the patches.

For the sake of completeness, in Figure 4.19 it is investigated the uplink maximum achievable rate for all the UEs, since it is more critical wrt. the downlink one. The contribution of the UAV is crucial to allow the communication between these nodes and the eNB. Indeed, when the UEs are no longer supported by the drone, the data rate drops to zero due to the high loss characterizing the direct link. Otherwise, it can be observed time-discrete variations of the rate, which are caused by the MCS switching. This is due to (i) the variation of the UAV-GUs distance and (ii) the fast-fading effect, which clearly worsens as the number of served UEs increases. Moreover, when the UAV is closest to a UE, the theoretical maximum rate of 18.336 Mbps is reached, as already seen in Figure 4.16 of Scenario #1.

4.2.3 Swarm-aided IRS-enhanced smart city

The last scenario investigates all the available *Serving Configurators* described in Section 3.4.4, in the context of a smart city. Indeed, the urban environment is particularly useful to analyse both LoS and NLoS cases. As depicted in the left of Figure 4.20, multiple buildings and an uniform grid of 25 UEs are considered. Each UE communicates with an eNB placed on the top of the bottom-left building, at 30 m of height. As it can be noted, in the right of Figure 4.20, the downlink REM shows the shadowing effect due to the presence of buildings, which obstruct the LoS between some UEs and the eNB. As a consequence, there are nodes that cannot communicate, since the SINR is under the threshold, according to the `ns3::MiErrorModel` [149]. To cope with this issue, the communication system is enhanced with one and then four IRS-equipped UAVs. In the former case, the UAV is placed in $[200, 200, 50]^\mathsf{T}$, whereas in the latter the UAVs are located at $\{[100, 200, 50]^\mathsf{T}, [200, 300, 50]^\mathsf{T}, [300, 200, 50]^\mathsf{T}, [200, 100, 50]^\mathsf{T}\}$.

In order to saturate the LTE capacity, a live streaming traffic is simulated for a mission that lasts 75 s. With this aim, the `ns3::UdpEchoClientApplication` is employed, which leverages the UDP protocol to transmit a packet of 64 KiB, i.e., the maximum possible size, every 0.03 s. All the three proposed *Serving*

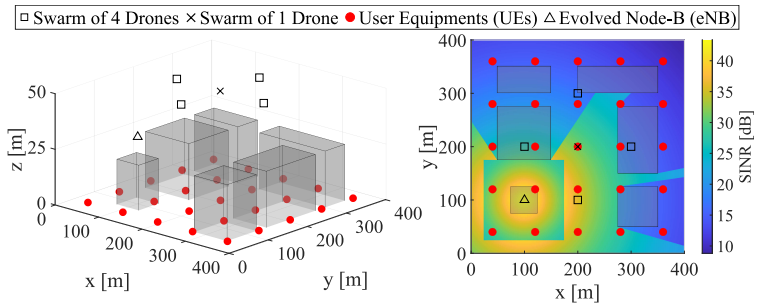


Figure 4.20: Simulated smart city scenario.

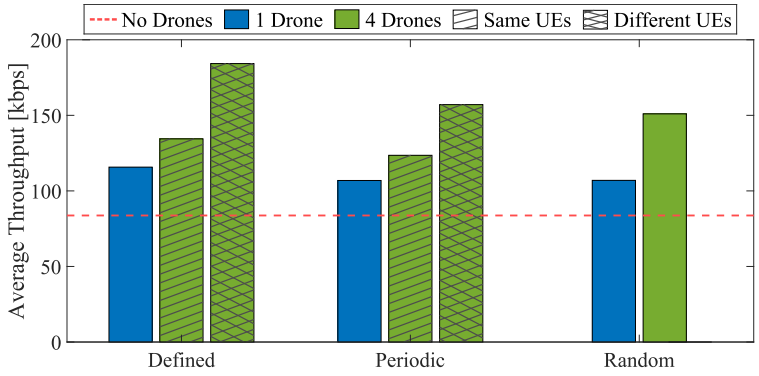


Figure 4.21: UEs average throughput adopting different *Serving Configurators*.

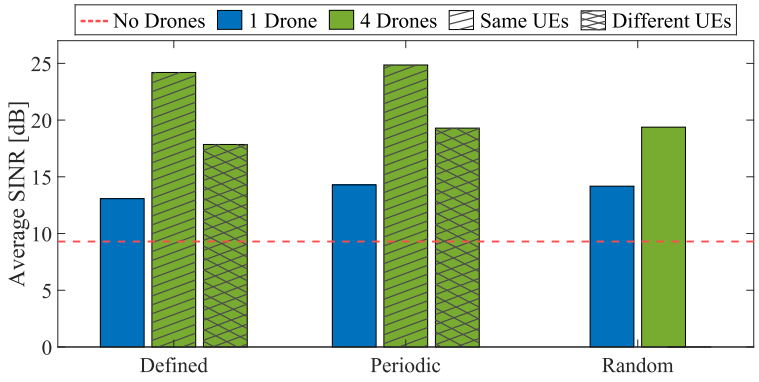


Figure 4.22: UEs average SINR adopting different *Serving Configurators*.

Configurators are tested, labelled as *Defined*, *Periodic*, and *Random*. Specifically, the former is set up to exclusively serve the nodes that have an SINR below the threshold. The last two, instead, serve all the nodes. Moreover, a baseline is also considered (the red dashed line) in which no drones assist the UEs, which have to rely solely on the direct link with the eNB. The results, in terms of average throughput and SINR, are reported in Figures 4.21 and 4.22, respectively. In both Figures, the blue bars indicate the case where only one IRS-equipped drone is employed, whereas the green ones consider a swarm of 4 UAVs. In the latter case, patterns are used to distinguish two different approaches: *Same UEs* refers to the case in which all drones simultaneously assist a given UE in each time interval; vice versa, *Different UEs* indicates that all drones serve distinct UEs. For obvious reasons, the *Random* case do not discuss such a difference.

It is evident that, wrt. the baseline approach, the employment of IRSs leads to an improvement in both the average throughput and SINR, of at least $\sim 27.66\%$ and $\sim 40.6\%$, respectively. Of course, these benefits become more prominent as the number of drones increases. Among the adopted configurators, it can be noted that there are no major differences in terms of SINR. Indeed, even if in different orders, UEs are served for about the same time and with the same bandwidth. Nonetheless, the *Periodic* presents slightly better performances. However, when the average throughput is considered, the *Periodic* configurator (which performs very similar to the random one) does not guarantee the same benefits brought by the *Defined* one. Indeed, the latter focuses on serving those nodes which demand more signal power to reach the required minimum SINR, which in turn produces an higher overall system throughput. A similar rationale can be applied when, given a configurator, *Same UEs* and *Different UEs* are compared. In fact, serving distinct UEs at the same time allows them to use a higher MCS, which yields a greater average throughput, even if the corresponding SINR are comparable.

4.3 Satellite-to-high altitude platform communication link

In this section, the channel link between a GEO satellite and a HAP is evaluated to assess the preliminary implementation of the satellite feature. The satellite is

16 effective isotropic radiated power (EIRP)

Table 4.5: Simulation parameters and settings.

Parameter	**Value**
Mission duration (T)	284 [h]
3GPP Environment	NTN Rural [51, Section 6.1.2]
Update period	1 [s]
Frequency (f)	20 [GHz]
Shadowing	Disabled
Time resolution	1000 [s^{-1}]
Bandwidth	400 [MHz]
EIRP[16] density	40 [dBW/MHz]
Antenna noise figure	1.2 [dB]
HAP speed	24 [m/s]
GEO antenna gain	58.5 [dBi]
HAP antenna gain	39.7 [dBi]
GEO antenna radius	2.5 [m]
HAP antenna radius	0.3 [m]
GEO antenna inclination	180.0 [deg]
HAP antenna inclination	0 [deg]
1st PoI (Take off/Landing)	[78.244789°, 15.4843571°, 20 km]
2nd PoI (Iran PoI)	[35.7074505°, 51.1498211°, 20 km]
3rd PoI (GEO Satellite)	[0.04°, −4.95°, 20 km]
4th PoI (Iceland PoI)	[64.133542°, −21.9348416°, 20 km]

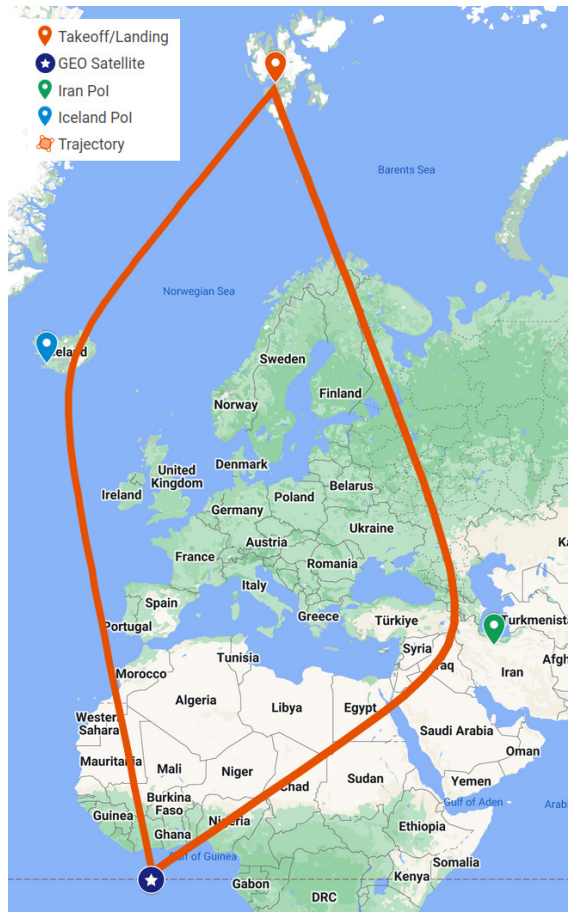


Figure 4.23: An overview of the trajectory of the HAP, its PoIs, and the satellite position over the Earth.

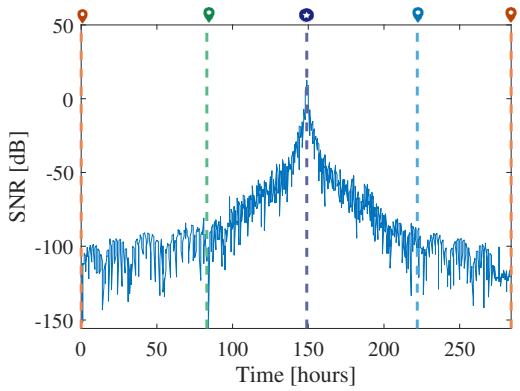


Figure 4.24: The evolution of the SNR during the mission.

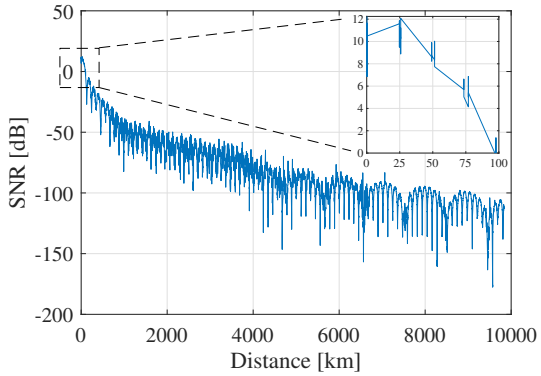


Figure 4.25: SNR vs. the distance between the HAP and the GEO satellite, projected on the Earth.

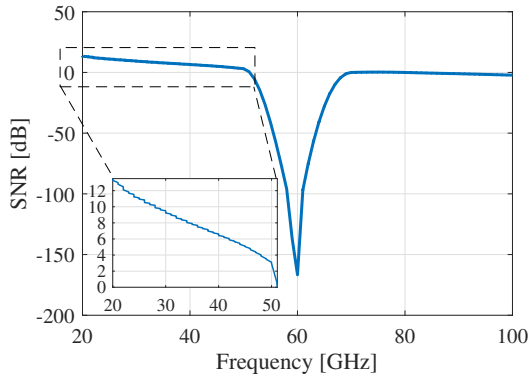


Figure 4.26: SNR in the PoI of maximum link gain vs. the frequency.

located at $[0.04°, -4.95°, 35\,770.88\,\text{km}]$, which corresponds to the actual position of Eutelsat 5 West B. The HAP follows a curvilinear trajectory generated with 4 PoIs. Besides, the HAP adopts the mobility model described in Section 3.4.1, with a constant speed of 24 m/s. This leads to a total mission duration of $T = 1\,023\,160\,\text{s} \simeq 284\,\text{h}$. A comprehensive overview of the described scenario and mobility pattern is illustrated in Figure 4.23, while simulation parameters are listed in Table 4.5. The HAP and the GEO satellite are equipped with a circular aperture antenna operating at 20 GHz. This antenna, also known as reflector, is modelled based on the `ns3::CircularApertureAntennaModel` class in the `ns3-ntn` module [112]. The HAP (GEO satellite) antenna has a maximum gain of 39.7 dB (58.5 dB), a diameter of 0.6 m (5 m), and an inclination angle of 0° (180°.) The evaluation is focused on downlink communication, where signals are sent from the satellite to the HAP with a transmission power of 37.5 dBm and a bandwidth of 400 MHz. The channel model described in Section 3.4.5 is considered with a rural environment [51, Section 6.1.2] and with the assumption of LoS visibility. Given that the HAP flies in the stratosphere, i.e., at a fixed altitude of 20 km, it is assumed that the impact of shadowing as well as of tropospheric scintillation is negligible[17]. Moreover, the impact of atmospheric absorption is considered through all the layers of the atmosphere,

17 Even though 3GPP works do not dimension these kind of links, the implementation of the TR 38.821 in IoD-Sim accounts for this extension by neglecting tropospheric phenomena described in Section 3.4.5.

even though the HAP flies in the stratosphere, so as to obtain worst-case results. Considering the above setup, Figure 4.24 illustrates the evolution of the signal-to-noise ratio (SNR) over time of the link between the GEO satellite and the HAP. The markers refer to those in Figure 4.23, and indicate when the HAP reaches a certain PoI according to the given level of interest. As expected, as the HAP approaches the geographical position of the GEO satellite (i.e., the starred marker, corresponding to the area in the Gulf of Guinea,) the SNR increases, thus reaching a maximum value of 13.0584 dB. With a bandwidth of 400 MHz as per 3GPP TR 38.821 [52], this corresponds to a PHY-layer capacity of approximately 1.78 Gbps, which is enough to realise HAP-to-satellite communication. However, the SNR drops below 0 dB as the HAP moves farther away from the GEO satellite, i.e., as the length of the link between the two endpoints increases. For additional insights, Figure 4.25 shows the SNR as a function of the distance between the HAP and the GEO satellite projected over the Earth. As expected, the SNR is positive only for distances lower than ~100 km, which roughly corresponds to the service area of the HAP, and then drops below 0 dB everywhere else. This is due to (i) the high directivity of reflector antennas, which poses a limit to the coverage radius of the HAP, and (ii) the higher path loss as the distance between the HAP and the GEO satellite increases, and the elevation angle between the two decreases accordingly. Finally, another scenario is analysed in which the HAP hovers below the GEO satellite in the PoI of maximum link gain (i.e., the starred marker in Figure 4.23,) and the frequency varies from 20 to 100 GHz. Figure 4.26 shows that the SNR decreases as the frequency increases, as the L^{FS} in Equation (3.16) increases, with a significant drop around 60 GHz due to the impact of oxygen absorption in the atmosphere (in the order of 15 dB/km.) Still, the SNR is consistently above 0 dB as $f_c \leq 50$ GHz, where the very large bandwidth at these frequencies can support high-rate transmissions. In conclusion, the above results demonstrate that NTN communication between a GEO satellite and a HAP can be effectively established, at least from a PHY-layer standpoint, and simulated using IoD-Sim.

4.4 Wireless power transfer applications

At the time of writing, there is a lack of contributions that design and evaluate the performance of a UAV-powered IoT network that relies on a LEO CubeSat for information transmission. Therefore, this work proposes an optimisation

strategy to fairly distribute energy via WPT operation across GNs, while maximizing the transmitted sensed data. Specifically, the main contributions are as follows:

- An integrated NTN is designed to enable the uplink data transmission of GNs to a LEO CubeSat, within its visibility window. These IoT nodes are deployed in a specific area and are recharged by a UAV, equipped with an array antenna, that employs WPT. Accordingly, a mathematical model is developed to characterise the UAV-GN channel and CubeSat-GN communication link.

- Two mixed-integer non-linear programming (MINLP) problems are formulated to fairly maximise (i) the harvested energy of the GNs by jointly optimizing the UAV kinematics and the array antenna beamforming vectors, and (ii) the total transmitted data by fine-tuning the transmission plan of the nodes communicating with a LEO CubeSat. Both problems are non-convex and hence intractable. Therefore, the first is divided into two sub-problems, which are alternatively solved by leveraging also the successive convex approximation (SCA) technique, until convergence to a quasi-optimal solution is achieved. Following a similar strategy, also the second problem is solved by adopting the two aforementioned techniques.

- A lower-bound mathematical expression for the harvested energy is derived. The stochastic nature of the UAV-GN channel model represents a challenge, which is addressed by imposing a maximum out-of-service probability. This leads to a non-linear energy-harvesting model that can be employed also for system design and assessment.

- Multiple scenarios are simulated, analysed, and discussed under different parameter configurations, which include transmission power, number of GNs, and array antenna size. The performance of the conceived algorithm is then compared with a baseline approach, where the drone follows a snake-like trajectory and periodically recharges the nearest node, while adopting an optimal transmission scheduling.

WPT applications were designed and analysed outside IoD-Sim, given the experimental nature of a wireless technology that differs from traditional communication networks and the optimisation involved. Numerical results demonstrate

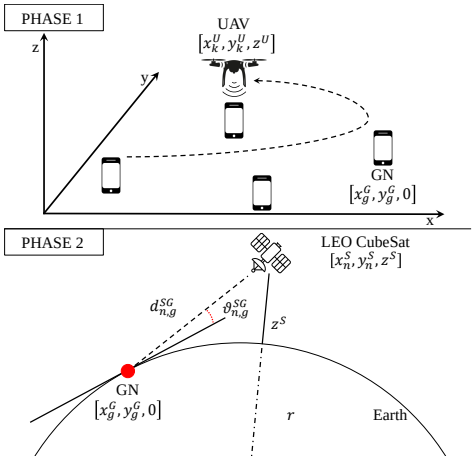


Figure 4.27: Reference scenario.

that the proposed strategy outperforms the baseline in terms of total transmitted data.

For the entire Section, the following notation is employed: boldface lower case letters refer to vectors; $j = \sqrt{-1}$ is the imaginary unit; $\mathbf{x}^{\mathsf{T}}$ is the transpose of a generic vector $\mathbf{x}$; $\mathbf{x}^{\mathsf{H}}$ is the Hermitian of a generic vector $\mathbf{x}$; $\mathbf{x} \otimes \mathbf{y}$ denotes the Kronecker product between two generic vectors; $x \sim \mathcal{CN}(\mu, \sigma^2)$ defines a circularly symmetric complex Gaussian distribution x with mean μ and variance σ^2; $\mathbf{I}_x$ represents the identity matrix of dimension x; $J_x(\cdot)$ denotes Bessel function of the first kind of order x; $\mathcal{O}(x)$ denotes the time-complexity of an algorithm of input size x, i.e, big O notation. The most significant parameters used in this work are summarised in Table 4.6.

4.4.1 System model

The entire mission, depicted in Figure 4.27, is divided into two phases. The first one considers a UAV wirelessly charging a set of G low-power GNs, while the second comprises the transmission of sensed data from the nodes to a LEO CubeSat.

The first phase is uniformly split into K timeslots of duration δ seconds each. The UAV flies at a fixed height z^{U} and follows a discretized trajectory, denoted by $\mathbf{q}_k^{\mathrm{U}} = \left[x_k^{\mathrm{U}}, y_k^{\mathrm{U}}\right]^{\mathsf{T}} \in \mathbb{R}^2$, at a velocity of $\mathbf{v}_k^{\mathrm{U}} \in \mathbb{R}^2$, with $k = 1, \ldots, K$. The

Symbol	Description	Symbol	Description
N	Number of timeslots for transmission.	$\chi^{\mathrm{SG}}_{n,g}$	CubeSat–GN link budget.
K	Number of timeslots for charging.	$\Upsilon^{\mathrm{SG}}_{n,g}$	CubeSat–GN link noise power.
G	Number of GNs.	$\Gamma^{\mathrm{SG}}_{n,g}$	CubeSat–GN SNR.
δ	Duration of each timeslot [s].	R^{SG}_{n}	Max data rate for CubeSat-GN link.
$\mathbf{q}^{\mathrm{U}}_{k}$	UAV position in cartesian coordinates.	B	GN-CubeSat channel bandwidth.
$\mathbf{v}^{\mathrm{U}}_{k}$	UAV velocity [m/s].	MCL	Maximum coupling loss.
$\mathbf{q}^{\mathrm{S}}_{n}$	CubeSat position in cartesian coordinates.	$\Delta^{\mathrm{SG}}_{n,g}$	CubeSat–GN uplink coupling loss.
$d^{\mathrm{UG}}_{k,g}$	UAV–GN distance.	$\mathbf{v}^{\mathrm{S}}$	CubeSat orbital speed.
$d^{\mathrm{SG}}_{n,g}$	CubeSat–GN distance.	G^{E}	Earth's gravitational constant.
$\theta^{\mathrm{SG}}_{n,g}$	CubeSat–GN inclination angle.	M^{E}	Earth's mass [kg].
$\varphi^{\mathrm{SG}}_{n,g}$	CubeSat–GN azimuth angle.	v^{F}	CubeSat speed footprint [m/s].
$\vartheta^{\mathrm{SG}}_{n,g}$	CubeSat–GN elevation angle.	F^{S}	CubeSat footprint diameter [m].
r	Earth radius [m].	T^{V}	CubeSat visibility time [s].
$\mathbf{\Lambda}$	Transmission schedule.	$\theta^{\mathrm{UG}}_{k,g}$	UAV–GN inclination angle.
$\mathbf{\Omega}$	Recharge schedule.	$\varphi^{\mathrm{UG}}_{k,g}$	UAV–GN azimuth angle.
$L^{I}_{n,g}$	CubeSat–GN comm. loss.	$\mathbf{h}_{k,g}$	UAV-GN WPT channel vector.
$G^{\mathrm{S}}_{n,g}, G^{\mathrm{G}}_{n,g}$	CubeSat and GN antenna gain.	κ	Rician K-factor.
ζ	Free space phase constant.	$\gamma_{k,g}$	UAV-GN WPT channel gain.
ϱ	Effective CubeSat antenna radius.	$E_{k,g}$	Energy harvested by GN [J].
f^{SG}, f^{UG}	Carrier frequencies [Hz].	$P, \overline{P}$	UAV and GNs transmission power [W].

Table 4.6: Main notations used in this work.

GNs are uniformly deployed over an area of interest with a diameter equal to d^{A} and can be in either one of these three states: energy harvesting, data upload, and idle. Moreover, each one is placed at known coordinates denoted by $\mathbf{q}_g^{\mathrm{G}} = \left[x_g^{\mathrm{G}}, y_g^{\mathrm{G}}\right]^{\mathsf{T}} \in \mathbb{R}^2$, with $g = 1, \ldots, G$. Therefore, it is possible to define the inclination and azimuth angles, i.e, $\theta_{k,g}^{\mathrm{UG}}$ and $\varphi_{k,g}^{\mathrm{UG}}$, between the g-th GN and the UAV as

$$\theta_{k,g}^{\mathrm{UG}} = \arccos \frac{z^{\mathrm{U}}}{d_{k,g}^{\mathrm{UG}}}, \quad \varphi_{k,g}^{\mathrm{UG}} = \arctan 2 \frac{y_k^{\mathrm{U}} - y_g^{\mathrm{G}}}{x_k^{\mathrm{U}} - x_g^{\mathrm{G}}}, \tag{4.3}$$

and corresponding distance as:

$$d_{k,g}^{\mathrm{UG}} = \sqrt{\|\mathbf{q}_k^{\mathrm{U}} - \mathbf{q}_g^{\mathrm{G}}\|^2 + (z^{\mathrm{U}})^2}. \tag{4.4}$$

Similarly to the former, also the second phase is split into N equal timeslots of duration δ seconds. The LEO CubeSat is assumed to be at constant altitude z^{S}, following a sun-synchronous circular orbit, denoted by $\mathbf{q}_n^{\mathrm{S}} = \left[x_n^{\mathrm{S}}, y_n^{\mathrm{S}}\right]^{\mathsf{T}} \in \mathbb{R}^2$, with $n = 1, \ldots, N$, at steady speed $\mathbf{v}^{\mathrm{S}} \in \mathbb{R}^2$. Thus, the inclination $\theta_{n,g}^{\mathrm{SG}}$ and azimuth $\varphi_{n,g}^{\mathrm{SG}}$ angles, between the g-th GN and the LEO CubeSat read:

$$\theta_{n,g}^{\mathrm{SG}} = \arccos \frac{z^{\mathrm{S}}}{d_{n,g}^{\mathrm{SG}}}, \quad \varphi_{n,g}^{\mathrm{SG}} = \arctan 2 \frac{y_n^{\mathrm{S}} - y_g^{\mathrm{G}}}{x_n^{\mathrm{S}} - x_g^{\mathrm{G}}}, \tag{4.5}$$

where the CubeSat-GN distance $d_{n,g}^{\mathrm{SG}}$, also known as slant range [150], can be expressed as

$$d_{n,g}^{\mathrm{SG}} = \sqrt{r^2 \sin^2 \vartheta_{n,g}^{\mathrm{SG}} + \left(z^{\mathrm{S}}\right)^2 + 2 z^{\mathrm{S}} r} - r \sin \vartheta_{n,g}^{\mathrm{SG}}, \tag{4.6}$$

with r representing the Earth's radius and $\vartheta_{n,g}^{\mathrm{SG}} = \frac{\pi}{2} - \theta_{n,g}^{\mathrm{SG}}$ being the elevation angle. Note that, $0 \leq \vartheta_{n,g}^{\mathrm{SG}} \leq \pi/2$ and specifically $\vartheta_{n,g}^{\mathrm{SG}} = 0$ at the sunrise and the sunset, while $\vartheta_{n,g}^{\mathrm{SG}} = \pi/2$ when the CubeSat is over the GNs. Since the altitude of the CubeSat is fixed, $d_{n,g}^{\mathrm{SG}}$ depends only on the elevation angle. Finally, according to the scheduling plan $\mathbf{\Lambda} = (\lambda_{n,g}) \in \{0, 1\}^{N \times G}$, if the energy harvested in the first phase is sufficient, then a GN can transmit the sensed data in the second one.

4.4.2 Drone model

The UAV is equipped with an uniform planar array (UPA), with $S = L \times W$ antenna elements, which works in one of the resonant frequencies of the GNs' monopole antenna. Beamforming is adopted in order to maximise the power transfer to the GN of interest. Typically, the air-to-ground links are characterised by a strong LoS component. However, the multi-path fading caused by reflections on the ground is not negligible. Therefore, the Rician distribution is adopted to capture both LoS and NLoS components [151], thus granting a realistic representation of the UAV-GN channel. Consequently, following Rician fading, the channel vector for the multiple-input-single-output (MISO) link between the UAV and the g-th GN, characterised by the Rician K-factor κ, can be modelled as:

$$\mathbf{h}_{k,g} = \sqrt{\frac{\kappa}{\kappa+1}}\bar{\mathbf{h}}_{k,g} + \sqrt{\frac{1}{\kappa+1}}\tilde{\mathbf{h}}_{k,g} \in \mathbb{C}^{S\times 1}, \tag{4.7}$$

where

$$\bar{\mathbf{h}}_{k,g} = \left[1, e^{-j\ell d \sin\theta^{\mathrm{UG}}_{k,g}\cos\varphi^{\mathrm{UG}}_{k,g}}, \ldots, e^{-j(W-1)\ell d \sin\theta^{\mathrm{UG}}_{k,g}\cos\varphi^{\mathrm{UG}}_{k,g}}\right]^{\mathsf{T}} \otimes \left[1, e^{-j\ell d \sin\theta^{\mathrm{UG}}_{k,g}\sin\varphi^{\mathrm{UG}}_{k,g}}, \ldots, e^{-j(L-1)\ell d \sin\theta^{\mathrm{UG}}_{k,g}\sin\varphi^{\mathrm{UG}}_{k,g}}\right]^{\mathsf{T}},$$

is the LoS deterministic component, which describes the large-scale fading phenomena, and $\tilde{\mathbf{h}}_{k,g} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_S)$ is the NLoS stochastic fluctuation due to multi-path propagation. Moreover, d is the distance between each element of the UAV's UPA, $\ell = \frac{2\pi}{c} f^{\mathrm{UG}}$, c is the speed of light, and f^{UG} the carrier frequency. Given the channel model description, the gain between the UAV and each GN can be expressed as:

$$\gamma_{k,g} = \left| \sqrt{\beta \left(d^{\mathrm{UG}}_{k,g}\right)^{-2}} \mathbf{w}_k^{\mathsf{H}} \mathbf{h}_{k,g} \right|^2, \tag{4.8}$$

where β denotes the channel power gain at the reference distance of 1 m, and $\mathbf{w}_k \in \mathbb{C}^{S\times 1}$ is the beam-forming vector.

The energy harvested by each g-th GN from the UAV can be non-linearly modelled [80, 83] as:

$$E_{k,g} = \frac{\alpha_0 P \delta \gamma_{k,g}}{\alpha_1 P \gamma_{k,g} + \alpha_1^2}, \tag{4.9}$$

where P is the transmission power of the UAV, and $\alpha_0 = 0.399$, $\alpha_1 = 0.826$ are positive constants determined in [80, 152]. Note that the adopted non-linear model is preferred wrt. a linear one, since it improves the overall accuracy and better estimates the time required to recharge each GN.

4.4.3 Satellite model

This Section discusses the model adopted to describe the uplink communication between the GNs and the LEO CubeSat, to derive an expression of the link budget, which is then employed to obtain the reciprocal visibility time, and hence the mission duration. Among the possible channel models available in the scientific literature [153], the proposed one aligns with the specifications outlined in 3GPP TR 38.811 [51]. It is worth mentioning that, since the locations of GNs and the trajectory of the satellite are known, a compensation of the frequency shift introduced by the Doppler effect can be always performed, and hence it is not taken into account. Moreover, the considered uplink channel is typically characterised by a large elevation angle of the LEO CubeSat wrt. GNs [154], thus leading to a communication link dominated by a strong LoS component and hence a negligible slow fading. Besides, the satellite is distant several hundred kilometers from the nodes, and hence the channel is subject to a significant path loss which makes the multi-path effect negligible [155]. To avoid interference among different nodes, the communication system has been designed in a TDMA fashion, such that at most one node per timeslot can communicate with the LEO CubeSat. This comes with the advantage, differently from frequency division multiple access (FDMA), that the GNs can effectively exploit all the available bandwidth. Each GN employs a COTS horizontally-oriented monopole antenna, assumed to be lossless, with linear polarisation that operates at frequency f^{SG} in the S-band [156]. In particular, the antenna gain [156] can be expressed[18] solely as function of the elevation angle $\vartheta^{\mathrm{SG}}_{n,g}$:

$$G^{\mathrm{G}}_{n,g} = 4\,\frac{\cos^2\left(\frac{\pi}{2}\cos\vartheta^{\mathrm{SG}}_{n,g}\right)}{\sin^2\vartheta^{\mathrm{SG}}_{n,g}\int_0^{\pi}\frac{\cos^2\left(\frac{\pi}{2}\cos\vartheta\right)}{\sin\vartheta}d\vartheta}. \tag{4.10}$$

18 It is worth remarking that, differently from the aforementioned 3GPP standard, the antenna gains are evaluated according to the referenced antenna theory principles. It is possible to check Table 4.7 for numerical differences wrt. the standard.

Likewise, the LEO CubeSat is equipped with a lossless circular patch antenna, whose gain [156] can be expressed as:

$$G_{n,g}^{\mathrm{S}} = 4\,\frac{\cos^2\varphi_{n,g}^{\mathrm{SG}} J_{02}'^2 + \cos^2\theta_{n,g}^{\mathrm{SG}} \sin^2\varphi_{n,g}^{\mathrm{SG}} J_{02}^2}{\int_0^{\pi/2} (J_{02}'^2 + \cos^2\theta J_{02}^2)\sin\theta d\theta}. \tag{4.11}$$

Specifically, J_{02}' and J_{02} read:

$$J_{02}' = J_0(\zeta\varrho\sin\theta_{n,g}^{\mathrm{SG}}) - J_2(\zeta\varrho\sin\theta_{n,g}^{\mathrm{SG}}), \tag{4.12}$$

$$J_{02} = J_0(\zeta\varrho\sin\theta_{n,g}^{\mathrm{SG}}) + J_2(\zeta\varrho\sin\theta_{n,g}^{\mathrm{SG}}), \tag{4.13}$$

with ζ being the free space phase constant and ϱ the effective radius. Furthermore, the channel is characterised by different impairments [157] which can be modelled as follows:

$$L_{n,g}^{\mathrm{I}} = L_{n,g}^{\mathrm{A}}\, L_{n,g}^{\mathrm{R}}\, L_{n,g}^{\mathrm{Sc}}\, L^{\mathrm{P}}. \tag{4.14}$$

In particular, $L_{n,g}^{\mathrm{I}}$ is estimated by taking into account the air attenuation and the atmospheric gas absorption $L_{n,g}^{\mathrm{A}}$ [158–160], the rainfall droplet $L_{n,g}^{\mathrm{R}}$ [161, 162], the scintillation attenuation $L_{n,g}^{\mathrm{Sc}}$ [156], and the polarisation attenuation L^{P} [156]. The transmitted signal of each GN undergoes polarisation rotation during the propagation in the ionosphere. It means that the signal may be polarized differently than intended on the satellite side. This phenomenon can be mitigated by using a circular-polarized signal, causing a maximum misalignment of $\pi/4$, which leads to $L^{\mathrm{P}} = 2$. Therefore, the combination of (4.10), (4.11), and (4.14) leads to the definition of the link budget [157]:

$$\chi_{n,g}^{\mathrm{SG}} = \frac{\overline{P}\, G_{n,g}^{\mathrm{S}}\, G_{n,g}^{\mathrm{G}}}{L_{n,g}^{\mathrm{FS}}\, L_{n,g}^{\mathrm{I}}}, \tag{4.15}$$

where $\overline{P}$ defines the transmission signal power of the GN. Further, $L_{n,g}^{\mathrm{FS}}$ [51] describes the free space propagation loss, which depends on the carrier frequency[19] f^{SG} and the GN-CubeSat distance $d_{n,g}^{\mathrm{SG}}$. Moreover, the receiver sensitivity [157]

19 It is assumed that f^{UG} and f^{SG} are different carriers defined in the S-band, such that the GN can employ the same monopole antenna for both information transmission and energy harvesting without interference.

represents the noise power of the link and is defined as

$$\Upsilon_{n,g}^{\text{SG}} = k_B \, \eta_{n,g}^{\text{SG}} \, B, \tag{4.16}$$

with k_B being the Boltzmann constant, B the channel bandwidth, and η^{SG} describing the equivalent system noise temperature for both antenna and receiver noise.

Once the link budget and the receiver sensitivity are defined, it is possible to obtain the SNR as:

$$\Gamma_{n,g}^{\text{SG}} = \frac{\chi_{n,g}^{\text{SG}}}{\Upsilon_{n,g}^{\text{SG}}}. \tag{4.17}$$

To evaluate the coverage of a radio access technology, the 3GPP introduced the maximum coupling loss (MCL) [163], which expresses the maximum loss in conducted power level, that a system may tolerate to properly establish a connection:

$$MCL = \frac{\overline{P}}{\hat{P}}, \tag{4.18}$$

where $\hat{P}$ is the minimum power required by the CubeSat to correctly decode the received signal. Therefore, it is possible to express the current uplink coupling loss for the g-th GN in the n-th timeslot as:

$$\Delta_{n,g}^{\text{SG}} = \frac{\overline{P}}{\chi_{n,g}^{\text{SG}}}. \tag{4.19}$$

Specifically, the GN is able to communicate with the CubeSat if and only if

$$\Delta_{n,g}^{\text{SG}} \leq MCL. \tag{4.20}$$

It is worth noting that $\Delta_{n,g}^{\text{SG}}$ is inversely proportional to $d_{n,g}^{\text{SG}}$, and hence to the elevation angle $\vartheta_{n,g}^{\text{SG}}$. The minimum elevation angle able to satisfy (4.20) is denoted as $\vartheta_{\text{MIN}}^{\text{SG}}$ and the period during $\vartheta_{\text{MIN}}^{\text{SG}} \leq \vartheta_{n,g}^{\text{SG}}$ is called reciprocal visibility time. Although the function of the coupling loss is dependent on non-invertible components [158–162], $\vartheta_{\text{MIN}}^{\text{SG}}$ can be obtained by intersecting the coupling loss curve with the MCL thresholds defined by the standard, also called coverage classes, shown in Figure 4.28.

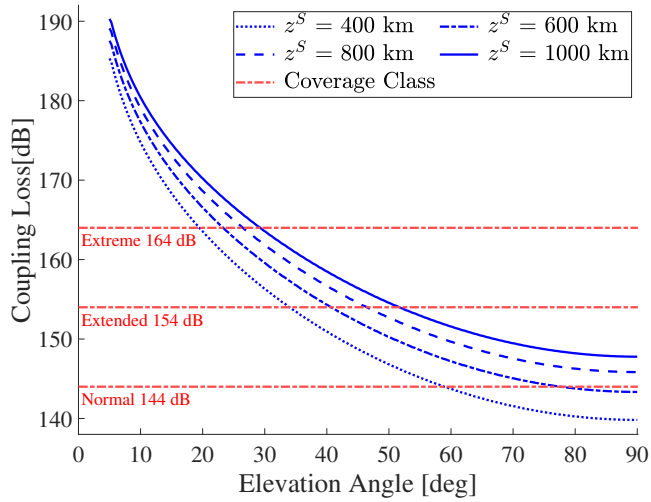


Figure 4.28: Maximum Coupling Loss thresholds of the coverage classes for different CubeSat's altitudes [164].

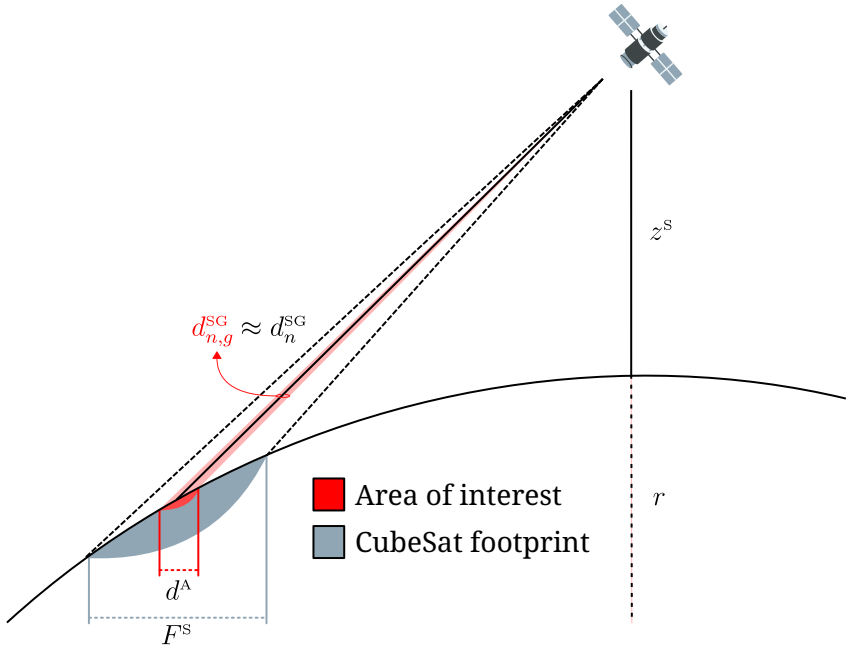


Figure 4.29: Illustration of Proposition 4.1.

► **Proposition 4.1.** As illustrated in Figure 4.29, without loss of generality, if the diameter of the area of interest is much smaller than the footprint of the CubeSat, i.e., $F^{\mathrm{S}} \gg d^{\mathrm{A}}$, its distance from each GN, defined in (4.6), is approximately the same and can be expressed as d_n^{SG}. ◄

As a consequence of Proposition 4.1, it is considered the same reciprocal visibility period for each GN, which in turn defines the total duration of the second phase δN, as derived hereby. The orbital period T^{S} of the CubeSat is denoted by the following equation:

$$T^{\mathrm{S}} = 2\pi\sqrt{\frac{(r+z^{\mathrm{S}})^3}{G^{\mathrm{E}}\,M^{\mathrm{E}}}}, \tag{4.21}$$

where M^{E} and G^{E} denote the mass and the gravitational constant of the Earth, respectively. Furthermore, the orbital speed v^{S} of an Earth's satellite is:

$$v^{\mathrm{S}} = \sqrt{\frac{G^{\mathrm{E}}\,M^{\mathrm{E}}}{(r+z^{\mathrm{S}})}} = \frac{2\pi(r+z^{\mathrm{S}})}{T^{\mathrm{S}}}. \tag{4.22}$$

Once defined the orbital speed v^{S}, it is necessary to compute the speed v^{F} of the covered area at the Earth side, namely footprint, as follows:

$$v^{\mathrm{F}} = v^{\mathrm{S}} - 2\pi\frac{z^{\mathrm{S}}}{T^{\mathrm{S}}} = 2\pi\frac{r}{T^{\mathrm{S}}}. \tag{4.23}$$

Moreover, the footprint diameter F^S can be expressed by:

$$F^{\mathrm{S}} = 2\,z^{\mathrm{S}}\,\tan(\theta_{\mathrm{MIN}}^{\mathrm{SG}}). \tag{4.24}$$

with $\theta_{\mathrm{MIN}}^{\mathrm{SG}} = \frac{\pi}{2} - \vartheta_{\mathrm{MIN}}^{\mathrm{SG}}$. Therefore, the duration δN, which corresponds to the visibility time T^{V} can be obtained as:

$$T^{\mathrm{V}} \triangleq \delta N = \frac{F^{\mathrm{S}}}{v^{\mathrm{F}}} = \frac{2\,z^{\mathrm{S}}\,\tan(\theta_{\mathrm{MIN}}^{\mathrm{SG}})}{r}\sqrt{\frac{(r+z^{\mathrm{S}})^3}{G^{\mathrm{E}}\,M^{\mathrm{E}}}}. \tag{4.25}$$

Finally, as a further consequence of Proposition 4.1, it is also possible to consider the same channel condition for each GN, and hence approximate Equation 4.17 as follows:

$$\Gamma_n^{\mathrm{SG}} \simeq \Gamma_{n,g}^{\mathrm{SG}} \forall g. \tag{4.26}$$

Consequently, recalling the well-known Shannon formula [165], the maximum achievable data rate of a CubeSat-GN link is

$$R_n^{\mathrm{SG}} = B \log_2(1 + \Gamma_n^{\mathrm{SG}}). \tag{4.27}$$

4.4.4 Wireless power transfer optimisation

Define $\mathbf{Q} = \{\mathbf{q}_k^{\mathrm{U}}\}_{k=1}^K$, $\mathbf{V} = \{\mathbf{v}_k^{\mathrm{U}}\}_{k=1}^K$, and $\mathbf{W} = \{\mathbf{w}_k\}_{k=1}^K$. The first phase concerning the kinematics of the drone and the node battery charging can be optimised by solving the following problem:

$$\max_{\eta_1, \mathbf{W}, \mathbf{Q}, \mathbf{V}} \eta_1 \quad \textbf{s.t.} \tag{4.28a}$$

$$\eta_1 \leq \sum_{k=1}^{K} E_{k,g}, \quad \forall g : 1, \ldots, G, \tag{4.28b}$$

$$\mathbf{q}_{k+1}^{\mathrm{U}} = \mathbf{q}_k^{\mathrm{U}} + \delta \mathbf{v}_k^{\mathrm{U}}, \quad \forall k : 1, \ldots, (K-1), \tag{4.28c}$$

$$\mathbf{q}_1^{\mathrm{U}} = \mathbf{q}_K^{\mathrm{U}} = \mathbf{q}_0^{\mathrm{U}}, \tag{4.28d}$$

$$\mathbf{v}_1 = \mathbf{v}_K = \mathbf{0}, \tag{4.28e}$$

$$\|\mathbf{v}_k\| \leq v_{\mathrm{MAX}}, \quad \forall k : 1, \ldots, K, \tag{4.28f}$$

$$\|\mathbf{v}_{k+1} - \mathbf{v}_k\| \leq a_{\mathrm{MAX}} \delta, \quad \forall k : 1, \ldots, (K-1). \tag{4.28g}$$

Problem (4.28) aims at fairly maximizing the battery charge of all the nodes through constraint (4.28b). Equation (4.28c) describes the kinematics of the drone, with the given initial and final point of the trajectory $\mathbf{q}_0^{\mathrm{U}}$ and the correspondent speed imposed by (4.28d) and (4.28e), respectively. Moreover, (4.28f) and (4.28g) limit the maximum speed and acceleration of the flight by v_{MAX} and a_{MAX}, respectively. However, (4.28) is a MINLP problem, and hence intractable in the present form. In particular, the stochastic formulation of the channel gain requires a dedicated strategy to derive an optimal solution. Therefore, the original problem is divided in two sub-problems, which are then alternately solved until convergence to a quasi-optimal solution is achieved.

Sub-problem 1: charge plan optimisation

The first sub-problem aims at optimizing the beamforming vectors $\mathbf{W}$, such that the amount of energy harvested by the GNs is maximised in a fairly manner, as follows:

$$\max_{\eta_1, \mathbf{W}} \eta_1 \quad \textbf{s.t.} \quad (4.28b). \tag{4.29}$$

Still, (4.29) is non-convex in $\mathbf{W}$ due to constraint (4.28b), which is affected by the stochastic nature of the energy term $E_{k,g}$. To tackle this issue, the Maximum Ratio Combining [165] approach is adopted as beamforming strategy, which is indeed the optimal solution to maximise the energy harvested by a single GN:

$$\mathbf{w}_k = \frac{\mathbf{h}_{k,g}}{\left\|\mathbf{h}_{k,g}\right\|}. \tag{4.30}$$

Therefore, it is necessary to define a charging plan $\Omega = (\omega_{k,g}) \in \{0,1\}^{K \times G}$ describing which node is charged in each timeslot[20]. Hence, when a GN is selected, i.e., $\omega_{k,g} = 1$, the energy term $E_{k,g}$ can be rearranged combining (4.9) and (4.30) as

$$\widetilde{E}_{k,g} = \frac{\alpha_0 P \delta \beta \left\|\mathbf{h}_{k,g}\right\|^2}{\alpha_1 P \beta \left\|\mathbf{h}_{k,g}\right\|^2 + \alpha_1^2 \left(d_{k,g}^{\mathrm{UG}}\right)^2}, \tag{4.31}$$

which, however, maintains a stochastic nature. Given an Out-of-Service probability ε, the minimum guaranteed energy $\overline{E}_{k,g}$ harvested by a GN, i.e, $\omega_{k,g} = 1$, can be obtained as follows:

$$\begin{aligned} \mathbb{P}\left(\widetilde{E}_{k,g} < \overline{E}_{k,g}\right) &= \mathbb{P}\left(\left\|\mathbf{h}_{k,g}\right\|^2 < \frac{\alpha_1^2 \left(d_{k,g}^{\mathrm{UG}}\right)^2 \overline{E}_{k,g}}{P\beta\left(\alpha_0 \delta - \alpha_1 \overline{E}_{k,g}\right)}\right) \\ &= F\left(\frac{\alpha_1^2 \left(d_{k,g}^{\mathrm{UG}}\right)^2 \overline{E}_{k,g}}{P\beta\left(\alpha_0 \delta - \alpha_1 \overline{E}_{k,g}\right)}\right) \leq \varepsilon, \end{aligned} \tag{4.32}$$

20 In this work, the side lobes that can eventually point to/illuminate other GNs are not considered, since their contribution is negligible.

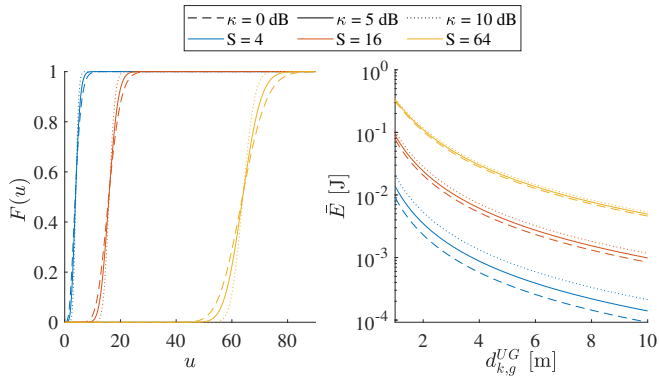


Figure 4.30: The CDF of the stochastic harvested energy $\widetilde{E}_{k,g}$ (left) and the minimum harvested energy $\overline{E}_{k,g}$ (right) with $\varepsilon = 0.01$ and $P = 49$ dBm, for different number of antenna elements S and K-factor κ.

with $F(\cdot)$ describing the cumulative distribution function (CDF) of the stochastic energy expression $\widetilde{E}_{k,g}$ in (4.31). It is worth noting that the latter follows a non-central chi-squared distribution, and the correspondent CDF is

$$F(u) = 1 - Q_S\Big(\sqrt{2S\kappa}, \sqrt{2(\kappa+1)u}\Big), \tag{4.33}$$

where $Q_S(\cdot)$ is the Marcum Q-function of order S. Considering the worst-case scenario, in which (4.32) holds with equality, the final energy term $\overline{E}_{k,g}$ can be derived as

$$\overline{E}_{k,g} = \frac{\alpha_0 P \delta \beta \overline{Q}_S^2}{\alpha_1 P \beta \overline{Q}_S^2 + 2\alpha_1^2 \Big(d_{k,g}^{\mathrm{UG}}\Big)^2 (\kappa+1)}, \tag{4.34}$$

$$\overline{Q}_S \triangleq Q_S^{-1}\Big(\sqrt{2S\kappa}, 1-\varepsilon\Big), \tag{4.35}$$

where $Q_S^{-1}(\cdot)$ is the inverse Marcum Q-function, which can be computed numerically or via analytical approximation. Figure 4.30 shows the CDF of $\widetilde{E}_{k,g}$ (left) defined in (4.33) and the derived energy term of $\overline{E}_{k,g}$ (right) obtained in (4.34).

Therefore, problem (4.29) can be rearranged as

$$\max_{\eta_1,\Omega} \eta_1 \quad \textbf{s.t.} \tag{4.36a}$$

$$\Omega \in \{0,1\}^{K\times G} \tag{4.36b}$$

$$\eta_1 \leq \sum_{k=1}^{K} \omega_{k,g}\overline{E}_{k,g}, \quad \forall g : 1,\dots,G, \tag{4.36c}$$

$$\sum_{g=1}^{G} \omega_{k,g} \leq 1, \quad \forall k : 1,\dots,K,, \tag{4.36d}$$

where $E_{k,g}$ in (4.28b) has been substituted with (4.34). Moreover, constraints (4.36b) and (4.36d) impose that the drone can only recharge one sensor per timeslot. Still, (4.36) is non-convex due to the presence of the binary charging plan Ω. To cope with this issue, several works in the scientific literature employ only the relaxation of the integer constraint followed by a rounding procedure of the obtained values, which often results to be infeasible or far from optimal in the best case. To avoid such a scenario and to derive a quasi-integer solution, it is jointly employed (i) the relaxation of (4.36b), and (ii) an additional term in the objective function which encourages the adoption of a binary solution:

$$\max_{\eta_1,\Omega} \eta_1 + \rho_1 \sum_{k=1}^{K}\sum_{g=1}^{G}\left(\omega_{k,g} - \frac{1}{2}\right)^2 \quad \textbf{s.t.} \tag{4.37a}$$

$$0 \leq \omega_{k,g} \leq 1, \quad \forall k : 1,\dots,(K-1),\ \forall g : 1,\dots,G, \tag{4.37b}$$

$$(4.36c), (4.36d).$$

In the above formulation, $\rho_1 \in \mathbb{R}^+$ acts as a weight that, if too low makes the additional term ineffective, otherwise if too high causes η_1 to become irrelevant. In the next Section, an empirical rule for the problem scaling, including ρ_1, will be discussed. Still, the objective function is non-convex wrt. Ω. To cope with this issue, the SCA technique is employed. Recalling that the first-order Taylor expansion is a global understimator for convex functions, it is possible to lower-bound the objective function for the local point $\overline{\omega}_{k,g}$, thus leading to the

final formulation:

$$\max_{\eta_1,\Omega} \eta_1 + \rho_1 \sum_{k=1}^{K} \sum_{g=1}^{G} \omega_{k,g}(2\overline{\omega}_{k,g} - 1) \quad \textbf{s.t.} \tag{4.38}$$
$$(4.36c), (4.36d), (4.37b).$$

It can be verified that (4.38) is convex. The scheduling plan is obtained by updating the value of the local point with the solution of the previous iteration, until convergence to a prescribed accuracy ξ is achieved.

Finally, since $\Omega \in [0,1]^{K \times G}$, a round operation is performed. In particular, for each timeslot k, only the $\omega_{k,g}$ that has the maximum value is set to 1, while the others to 0. Therefore, constraints (4.36b) and (4.36d) are satisfied.

Sub-problem 2: drone kinematics optimisation

Given the charging plan Ω and the other results derived above, the trajectory-related parameters and the transmission plan are hereby optimised. Note that η_1 is re-optimised to derive a fair solution from the energy-harvesting perspective. The second sub-problem reads:

$$\max_{\eta_1,\mathbf{Q},\mathbf{V}} \eta_1 \quad \textbf{s.t.} \tag{4.39}$$
$$(4.36c), (4.28c) - (4.28g),$$

which however is non-convex due to the presence of the squared distance term $d_{k,g}^{\mathrm{UG}}$ at the denominator of $\overline{E}_{k,g}$ in constraints (4.36c). To tackle this issue, it is first necessary to introduce a set of slack variables $\mathbf{B} = \{b_{k,g} \geq 0\}$. Then, recalling the definition of the distance in (4.4), the slack variables are lower-bounded such that

$$\left(d_{k,g}^{\mathrm{UG}}\right)^2 \leq b_{k,g}. \tag{4.40}$$

Similarly to the previous sub-problem, the SCA technique is employed which leads to the following inequality chain

$$\begin{aligned}\overline{\overline{E}}_{k,g} &= \frac{\alpha_2}{\alpha_3 + \alpha_4 \overline{b}_{k,g}} - \frac{\alpha_2 \alpha_4}{\left(\alpha_3 + \alpha_4 \overline{b}_{k,g}\right)^2}(b_{k,g} - \overline{b}_{k,g}) \\ &\overset{(a)}{\leq} \frac{\alpha_2}{\alpha_3 + \alpha_4 b_{k,g}} \overset{(b)}{\leq} \overline{E}_{k,g},\end{aligned} \tag{4.41}$$

where (a) is due to the first order Taylor expansion, (b) follows from (4.40), $\alpha_2 \triangleq \alpha_0 P \delta \beta \overline{Q}_S^2$, $\alpha_3 \triangleq \alpha_1 P \beta \overline{Q}_S^2$, and $\alpha_4 \triangleq 2\alpha_1^2(\kappa + 1)$. Hence, problem (4.39) is equivalent to

$$\max_{\eta_1, \mathbf{Q}, \mathbf{V}, \mathbf{B}} \eta_1 \quad \textbf{s.t.} \tag{4.42a}$$

$$\eta_1 \leq \sum_{k=1}^{K} \omega_{k,g} \overline{\overline{E}}_{k,g}, \quad \forall g : 1, \ldots, G, \tag{4.42b}$$

$$(4.28c) - (4.28g),$$

because in order to maximise the objective function it is necessary to maximise the new energy term in (4.41) and hence minimise $b_{k,g}$, until (4.40) holds with equality. Therefore, problem (4.42) is convex wrt. $\mathbf{Q}, \mathbf{V}$, and $\mathbf{B}$ and it is iteratively solved until a prescribed accuracy ξ is achieved.

Overall optimisation procedure

A quasi-optimal solution for the original problem (4.28) is derived by iteratively solving the two discussed sub-problems. It is worth specifying that, to avoid a waste of irradiated power, at the end of the entire procedure, the recharging plan Ω is further improved by setting to zero the entries which do not satisfy a minimum harvested energy threshold ν, which typically takes place when the drone is too far from a specific node (as can be seen in Figure 4.30.) For what concern the time complexity, the first sub-problem is in the order of $O\big(I_1(KG+1)^{3.5})\big)$, where I_1 is the number of iterations required by SCA. Similarly, the second sub-problem has a complexity of $O\big(I_2(4K+KG+1)^{3.5}\big)$. Therefore, the joint complexity is given by $O\big(M_1(I_1(KG+1)^{3.5} + I_2(4K+KG+1)^{3.5})\big)$, where M_1 is

denotes the number of iterations required to converge. More details can be found in Algorithm 1.

Algorithm 1: WPT Optimisation.

Set $\mathbf{q}_g^{\mathrm{G}}$ and $\mathbf{q}_k^{\mathrm{S}}$;
Compute R_k^{SG} and $\overline{Q}_S$;
Initialize $\overline{b}_{k,g}$, $\overline{\omega}_{k,g}$;
for $m : 1, \ldots, M$ **do**
  for $i : 1, \ldots, I$ **do**
    Solve (4.38) to obtain the objective function $s_{1,i}$, η_1, and $\mathbf{\Omega}$;
    $\overline{\omega}_{k,g} \leftarrow \omega_{k,g}\ \forall k, g$;
    if $i > 1$ **and** $|s_{1,i} - s_{1,i-1}|/|s_{1,i}| < \xi$ **then**
      $s^*_{1,m} \leftarrow s_{1,i}$; **break**;
  Round the optimised $\mathbf{\Omega}$;
  for $i : 1, \ldots, I$ **do**
    Solve (4.42) to obtain the objective function $s_{2,i}$, η_1, $\mathbf{Q}$, and $\mathbf{V}$;
    $\overline{b}_{k,g} \leftarrow b_{k,g}\ \forall k, g$;
    if $i > 1$ **and** $|s_{2,i} - s_{2,i-1}|/|s_{2,i}| < \xi$ **then**
      $s^*_{2,m} \leftarrow s_{2,i}$; **break**;
  if $m > 1$ **and** $|s^*_{1,m} - s^*_{1,m-1}|/|s^*_{1,m}| < \xi$ **and** $|s^*_{2,m} - s^*_{2,m-1}|/|s^*_{2,m}| < \xi$ **then**
    break;
  Round and process the optimised $\mathbf{\Omega}$;

4.4.5 Ground nodes-satellite transmission optimisation

Leveraging the results obtained in the previous optimised phase, i.e., the energy $E_{k,g}$ harvested by the GNs, the second phase encompassing the GNs' transmission scheduling can be optimised by deriving the optimal solution of the following problem:

$$\max_{\eta_2, \mathbf{\Lambda}} \quad \eta_2 \quad \textbf{s.t.} \tag{4.43a}$$

$$\mathbf{\Lambda} \in \{0, 1\}^{K \times G} \tag{4.43b}$$

$$\eta_2 \leq \sum_{n=1}^{N} \lambda_{n,g} R_n^{\mathrm{SG}} \quad \forall g : 1, \ldots, G, \tag{4.43c}$$

$$\delta\overline{P}\sum_{n=1}^{N}\lambda_{n,g} \leq \sum_{k=1}^{K}E_{k,g}, \quad \forall g : 1,\ldots,G, \tag{4.43d}$$

$$\sum_{g=1}^{G}\lambda_{n,g} \leq 1, \quad \forall k : 1,\ldots,K. \tag{4.43e}$$

Problem (4.43) focuses on fairly maximizing the sum-rate of all CubeSat-GN links through constraint (4.43c). Moreover, (4.43d) states that a GN can transmit only if enough energy has been harvested. Constraints (4.43b) and (4.43e) impose that only a GN can transmit in each timeslot. However, also (4.43) is a MINLP problem and hence non-convex due to (4.43b), which describes the binary nature of the transmission plan $\mathbf{\Lambda}$. Following the same rationale adopted to convexify problem (4.36), the binary constraint (4.43b) is relaxed and a new constraint is added to the formulation. Again, to encourage the adoption of an integer solution, one more addendum is introduced in the objective function employing the SCA technique, thus leading to:

$$\max_{\eta_2,\mathbf{\Lambda}} \eta_2 + \rho_2\sum_{n=1}^{N}\sum_{g=1}^{G}\lambda_{n,g}(2\overline{\lambda}_{n,g}-1) \quad \textbf{s.t.} \tag{4.44a}$$

$$0 \leq \lambda_{n,g} \leq 1, \quad \forall n : 1,\ldots,N,\ \forall g : 1,\ldots,G, \tag{4.44b}$$

$$(4.43c) - (4.43e).$$

The above formulation is convex and it is iteratively solved until a prescribed accuracy ξ is achieved. Similarly to (4.38), $\mathbf{\Lambda}$ needs to be rectified to satisfy (4.43b). However, in this case, $\lambda_{k,g}$ is rounded to 1 only if the value is ≥ 0.99, otherwise it is set to zero. This operation guarantees that the transmission takes place only if the scheduled GN has sufficient energy.

The time complexity associated with the reference problem is in the order of $O\big(I_3(KG+1)^{3.5}\big)\big)$, where I_3 is the number of iterations required by SCA. More details of the overall proposed algorithm can be found in Algorithm 2.

4.4.6 Numerical results and discussion

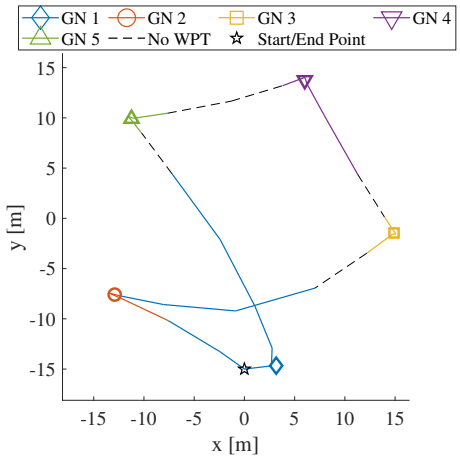
In this Section, a simulation campaign is carried out to assess the effectiveness of the proposed solution, which consists in sequentially executing Algorithm 1 and Algorithm 2. The investigated scenarios involve different area sizes, i.e.,

Algorithm 2: Transmission Optimisation.

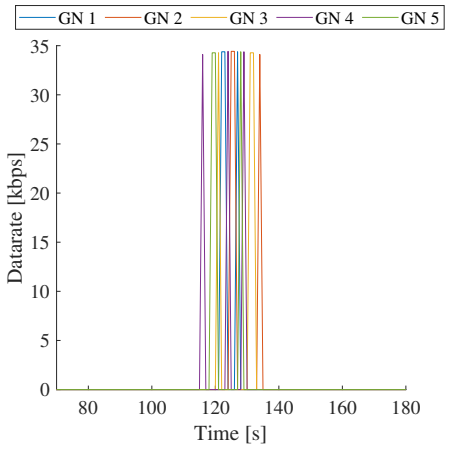
Set $\mathbf{q}_g^{\mathrm{G}}$ and $\mathbf{q}_k^{\mathrm{S}}$;
Compute R_k^{SG} and $\overline{Q}_S$;
Initialize $\overline{\lambda}_{k,g}$;
for $i : 1, \ldots, I$ **do**
    Solve (4.44) to obtain the objective function $s_{3,i}$, η_2, and $\mathbf{\Lambda}$;
    $\overline{\lambda}_{n,g} \leftarrow \lambda_{n,g}\ \forall n, g$;
    if $i > 1$ **and** $|s_{3,i} - s_{3,i-1}|/|s_{3,i}| < \xi$ **then**
        $s_3^* \leftarrow s_{3,i}$; **break**;
Rectify the optimised $\mathbf{\Lambda}$;

Parameter	Value	Parameter	Value
K	$\{30, 60\}$ [#]	N	250 [#]
G	$\{5, 10, 15\}$ [#]	B	180 [kHz]
L, W	$\{15, 20\}$ [#]	δ	1 [s]
$\mathbf{q}_0^{\mathrm{U}}$	$[0, -15]$ [m]	$\mathbf{q}_0^{\mathrm{S}}$	$[-795, 0]$ [km]
z^{U}	1 [m]	z^{S}	1000 [km]
v_{MAX}	15 [m/s]	$\mathbf{v}^{\mathrm{S}}$	$[6353, 0]$ [m/s]
a_{MAX}	3 [m/s^2]	$\vartheta_{\mathrm{MIN}}^{\mathrm{SG}}$	52 [deg]
f^{UG}	2.4 [GHz]	f^{SG}	1995 [MHz]
ζ	0.42 [#]	ϱ	3.05 [cm]
η^{SG}	615 [K]	ν	0.005 [#]
ρ_1	0.01[#]	ρ_2	2000 [#]
P	$\{46, 49\}$ [dBm]	$\overline{P}$	23 [dBm]
κ	10 [dB]	MCL	154 [dB]
ξ	10^{-3} [#]	ε	10^{-2} [#]

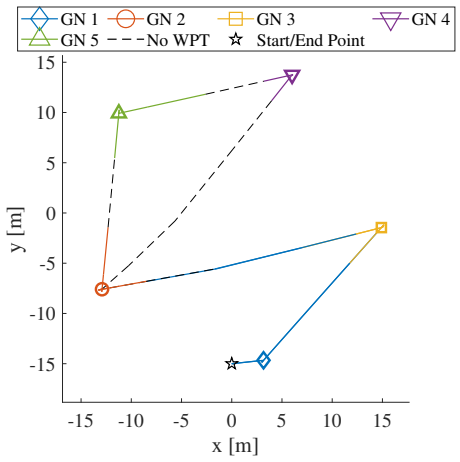
Table 4.7: Parameter settings.



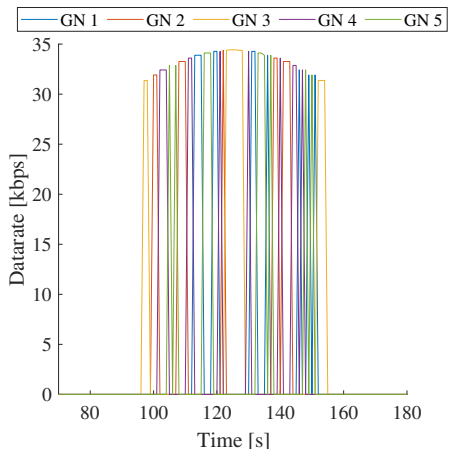
(a) Trajectory and served GNs with $K = 30$.



(b) Data rate with $K = 30$.



(c) Trajectory and served GNs with $K = 60$.



(d) Data rate with $K = 60$.

Figure 4.31: Analysis of the scenario with $G = 5$, $S = 225$, $P = 49\,\text{dBm}$, and $\delta = 1\,\text{s}$.

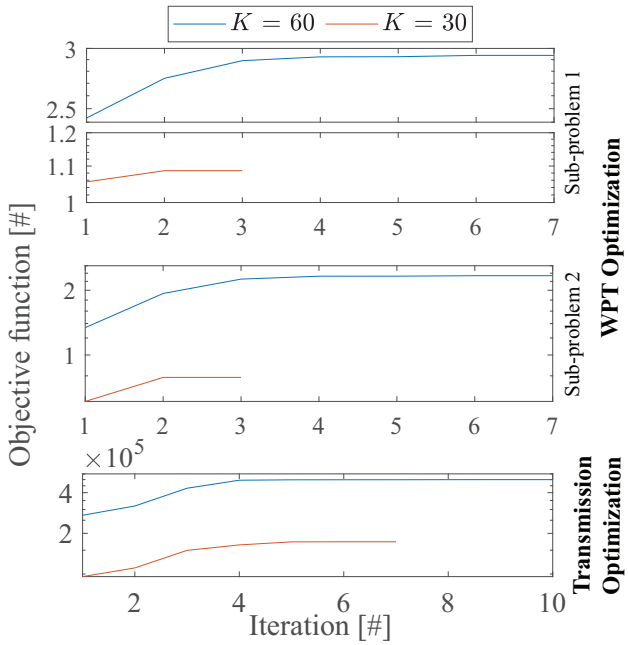


Figure 4.32: Convergence of the algorithms with $G = 5$, $S = 225$, $P = 49$ dBm, and $\delta = 1$ s.

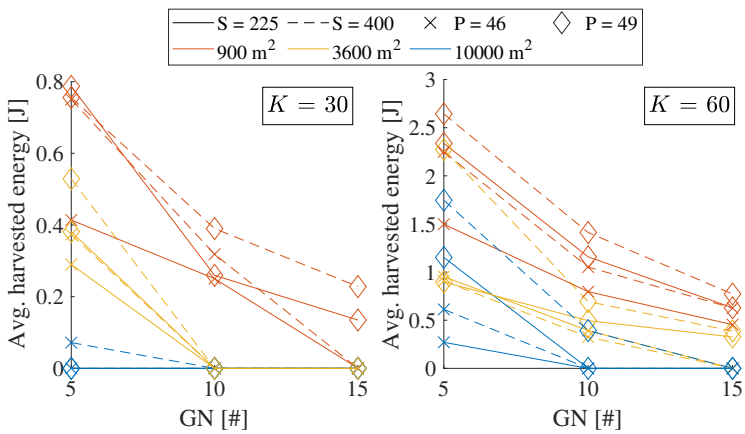
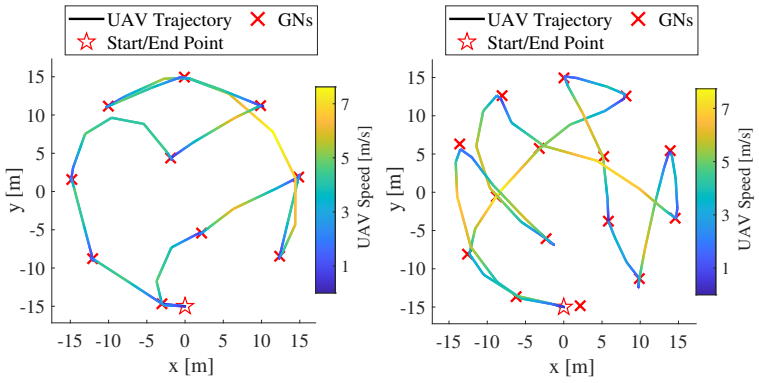
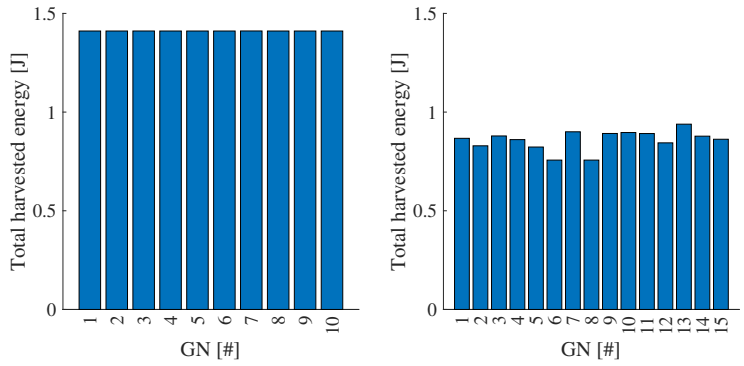


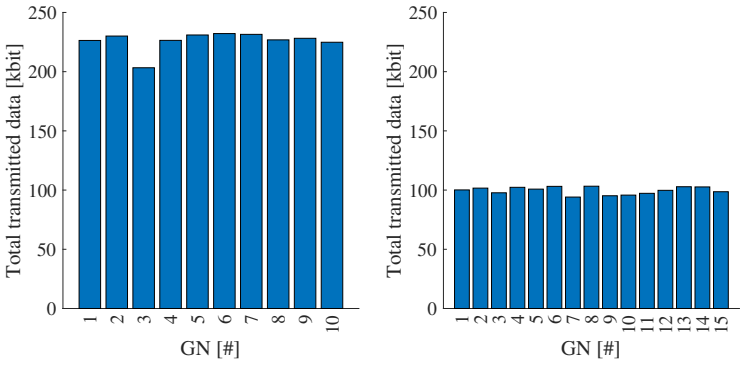
Figure 4.33: Average harvested energy for different parameters with $\delta = 1$ s.



(a) Trajectory and speed with $G = 10$. **(b)** Trajectory and speed with $G = 15$.



(c) Harvested energy with $G = 10$. **(d)** Harvested energy with $G = 15$.



(e) Transmitted data with $G = 10$. **(f)** Transmitted data with $G = 15$.

Figure 4.34: Analysis of the scenarios with $G = \{10,\ 15\}$, $S = 400$, $P = 49$ dBm, $K = 60$, $\delta = 1$ s.

30x30 m^2, 60x60 m^2, and 100x100 m^2, in which {5, 10, 15} GNs are uniformly deployed. The LEO CubeSat pursues a trajectory that follows a uniform linear motion, starting from position $\mathbf{q}_0^S$ at velocity $\mathbf{v}^S$. The parameters characterizing the CubeSat-GN link are set according to [70]. Furthermore, considering a noise figure of 5 dB, the equivalent system noise temperature $\eta^{SG} \simeq 615\,K$ [156] is the sum of the antenna noise temperature and the receiver noise temperature, which correspond to 290 K and 150 K, respectively. Moreover, the UAV is equipped with a squared UPA of {255, 400} elements to wirelessly recharge each GN at {46, 49} dBm. The transmission power is set in compliance with the ITU-R M.2135-1 Report [166, 167] for a LTE macro-cell deployed in urban and rural areas. All the simulation parameters are summarised in Table 4.7. A detailed discussion of the results, obtained by varying the aforementioned parameters, is followed by a comparison between the proposed solution and a baseline approach.

Objective function scaling

The normalisation of the objective functions of both problems is deemed necessary, since their components have different orders of magnitude, which affect the optimisation process, and hence the final solution. The possible values of η_1, given the involved parameters of problem (4.28), range from $\sim 10^{-1}$ to ~ 1, as can be verified by solving the relaxed problem (4.38), with $\rho_1 = 0$. Accordingly, to keep the fairness factor slightly above the additional term introduced to encourage integer solutions, the following $\rho_1 = 10^{-2}$ is adopted. The same rationale is applied for problem (4.44), thus leading to $\rho_2 = 2 \cdot 10^3$.

Analysis of the results

The first scenario considers $G = 5$ GNs recharged by a drone equipped with a UPA of $S = 225$ elements, with $P = 49$ dBm and $K = \{30, 60\}$. Indeed, the goal is to highlight how the duration of the first phase affects the second one in terms of obtained GNs' data rate. In this regard, Figures 4.31 (a) and 4.31 (c) jointly depict (i) the trajectory followed by the drone, and (ii) the GN recharged during the flight, and (iii) the initial position of the UAV, which is coincident with its final one. As it can be seen, in both scenarios, the trajectory paths tend to be straight to save time, which is convenient to reduce the distance between the UAV and the served GN, thus maximizing the energy income. It is worth noting that, thanks

to the procedure performed at the end of the recharging phase, the drone does not irradiate power when is too far from the served GN, thus saving energy that would be not efficiently harvested by the nodes. Furthermore, Figures 4.31 (b) and 4.31 (d) represent the data rate of nodes-satellite communications. It can be observed that the data rates are subject to the path loss which affects the satellite link. Indeed, the satellite trajectory is designed to firstly approach and then leave the reference area, thus leading to increasing and then decreasing data rates, that visually resemble a parabola. In both configurations, the latter is centered around the mid-point of the mission to maximise the overall sum-rate. However, for higher K, the effective transmission time of the second phase increases and, as a consequence, the shape of the parabola changes. This phenomenon is due to the fact that more energy is harvested in the recharging phase. Therefore, the maximum data rate of $\sim$ 35 kbps is achieved always at $\sim$ 125 s, which corresponds to the instant where the distance is minimised, i.e., the satellite is almost orthogonal to the area. Instead, the minimum data rates achieved are $\sim$ 34 kbps for $K = 30$ and $\sim$ 31 kbps for $K = 60$. Moreover, according to the constraints (4.43b) and (4.43e) which model the TDMA protocol, the peaks of the curves never overlap. For the sake of completeness, Figure 4.32 shows the convergence curves of the proposed optimisation algorithms, specifically related to the first and second phases. It is noteworthy that in the two chosen configurations, Algorithm 1 achieves convergence after 7 iterations at most, while Algorithm 2 after 10 iterations, both with a prescribed accuracy of ξ.

To further investigate the impact on the harvested energy when the parameters involved in the scenario vary, Figure 4.33 shows the average amount collected by a GN. As expected, most of the unfeasible, i.e., no harvested energy, configurations involve a 100x100 m^2 area. Indeed, the speed and acceleration limits of the drone, together with the maximum duration of the second phase, play the most important role in the mission feasibility. Clearly, also the transmission power and the number of antenna elements are aspects that can also zero out the gathered energy, especially for a significant number of GNs. This result is of fundamental importance for the following analysis, since it provides a solid indication of which other configurations can be studied. It is worth mentioning that, across all the examined scenarios, the energy consumption of the UAV, which can be calculated with [168, Eq. 12], is significantly lower than the commonly used commercial drones. To provide further insights, two more configurations are investigated with $G = \{10, 15\}$, $S = 400$, and $P = 49$ dBm.

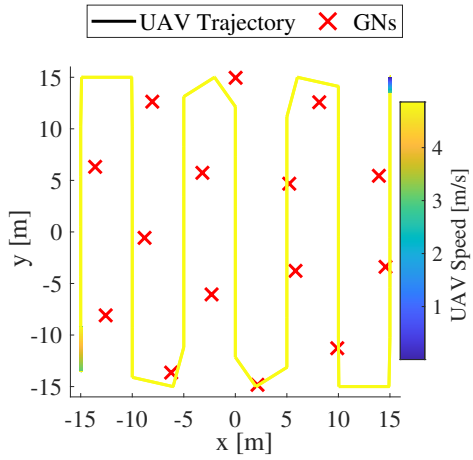
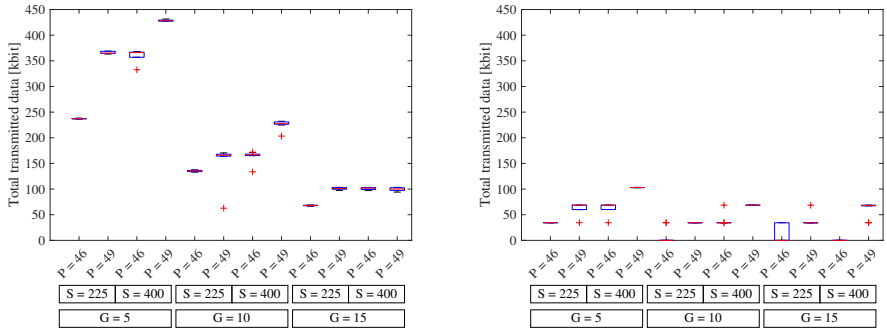


Figure 4.35: Example of the UAV trajectory and speed in the baseline scenario with $G = 15$ with $K = 60$, $\delta = 1$ s.

Figures 4.34 (a) and 4.34 (b) illustrate the trajectory and the speed of the drone. Clearly, in both setups, the UAV slows down and approaches the GNs as close as possible to increase the amount of harvestable energy. Indeed, the speed of the drone reaches a maximum of ~ 8 m/s.

This behaviour is reflected in Figures 4.34 (c) and 4.34 (d), where the total harvested energy per GN in both configurations is depicted. As a matter of fact, the proposed approach presents satisfactory results in terms of fairness. Moreover, it can be observed that the average amount of harvested energy decreases from ~ 1.4 J with $G = 10$ to ~ 0.9 J with $G = 15$. This in turn leads to a different amount of transmitted data (Figures 4.34 (e) and 4.34 (f)) with a mean of ~ 225 kbit and ~ 100 kbit, with a coefficient of variation of 0.037 and 0.031, respectively.

In conclusion, to prove its effectiveness, the proposed solution is compared to a baseline approach across all previously investigated scenarios. Specifically, the baseline foresees the drone covering the interest area by following a sampled snake-like trajectory at the minimum possible speed which satisfies the mission duration. The drone periodically recharges the battery of the nearest node throughout the flight. Subsequently, the final state of charge for each GN serves as input for Algorithm 2, which will endeavor to fairly distribute the available



(a) Proposed algorithm.

(b) Baseline algorithm.

Figure 4.36: Comparison of the total transmitted data between the proposed solution and the baseline with $K = 60$, $\delta = 1$ s.

transmission resources. An instance of the UAV trajectory and its speed is shown in Figure 4.35. The results of the comparison between the proposed solution and the baseline, with $K = 60$ and $\delta = 1$ s, is illustrated in Figure 4.36. Each bar reports the distribution of the total transmitted data per each GN, for all the possible combinations of transmission power P, number of antenna elements S and number of GNs G. As it can be seen, the proposed solution achieves great performance when the number of GNs is small enough, i.e., $G = 5$, allowing the UAV to hover over each one as much as possible. Indeed, more energy harvested by each GN corresponds to a greater amount of data transmitted. The same holds true when P and S increase. Finally, the baseline does not always provide sufficient energy to the GNs for the transmission. Instead, the proposed method demonstrates a higher total transmitted data volume compared to the baseline approach, ranging from a minimum of 1.5 to a maximum of 7 times higher, due to its fairly optimised energy distribution.

4.4.7 Remarks

In this work, a Satellite-IoT network powered by a UAV via WPT has been investigated. Starting from the system model, two MINLP problems have been formulated to fairly maximise the harvested energy of the GNs and the total transmitted data towards a LEO CubeSat. This requires a joint optimisation approach that encompasses recharge and transmission scheduling plans while

accounting for drone kinematics. To handle the non-convexity of both problems, the former has been decomposed into two sub-problems and then reformulated by also applying SCA technique. Meanwhile, the latter is tackled by employing the same two aforementioned techniques.

A simulation campaign has been conducted in order to evaluate the algorithm performance over different (i) number of GNs, (ii) number of antenna elements of the UAV, (iii) WPT transmission power levels, (iv) area sizes, and (v) mission duration. Finally, the proposed solution is compared with a reference baseline approach, demonstrating a substantial performance improvement, ranging from 1.5 to 7 times, in terms of the total amount of transmitted data.

In the future, the research efforts will be focused on the following directions:

- Extend the algorithm to multiple drones for a thoroughly integrated NTN cooperative network.
- Investigate inductive and capacitive WPT models to improve the overall efficiency.
- Employ 3D antenna arrays for a more flexible beam-steering and -forming strategies.
- Expand the trajectory optimisation and GN placement considering different heights to address different terrain conditions.
- Consider the presence of a IRS that can enhance the energy harvested by GNs.
- Investigate the adoption of other multiple access protocols, such as random access scheme, FDMA, and NOMA.

Finally, the proposal will lay the groundwork for the realisation of a test bed that will be used for experimentation and measurements in the context of 6G technologies.

5 Explainable artificial intelligence for cyber-physical-aware drones

Among the key challenges that hinder the realisation of an integrated T/NTN, UAVs are limited by their trustworthiness and spatial awareness required in order to access complex environments, especially the ones subject to strong winds and cluttered urban areas. In this regard, AI is emerging as a key enabler in solving this issue, as it represents a native component in the design of 6G communications and it plays a key role in the realisation of the intelligent sensing layer, the data mining layer, the intelligent control layer, and the smart application layer [169], as illustrated in Fig. 5.1.

Currently, UASs access to the airspace is severely limited, even neglected for BVLoS flights and for autonomous missions. This is due to the risks and hazards currently associated with UAVs, as they can (i) collide with buildings, sensitive infrastructure, and other objects, (ii) lose stability, (iii) succumb to external interference, (iv) disrupt air traffic and essential services, (v) violate privacy, and (vi) damage property and people.

At the same time, AI is being employed to solve critical challenges in autonomous networks [12]. One of the main problems, especially affecting DL models, is their opacity in providing insights and relations of the output with the given input, and hence they must be treated as black boxes. While DL has provided breakthroughs in language and computer vision, their use in mobile networks is hindered by: (i) data discrepancy due to time diversity, space diversity, and various features of networking data; (ii) feasibility especially in the data plane due to scarce resources, (iii) robustness due to AI systems vulnerable to possible attacks and manipulations; (iv) trust, as the decisions made by AI models involve complex parameters and non-linear transformations, which are not understandable to humans [169, 170].

It is clear that the fusion of these two key-enabling technologies is vital to ensure the realisation of 6G mobile communications. However, the new frontier of XAI can overcome the AI issues by interpreting the inference processes of learning models, boosting performance with simpler model structures, and improving the robustness against adversarial attacks, thus achieving an overall

improvement in its trustworthiness [170]. Specifically, XAI-based models unlock the opportunity to design and implement a transparent AI-driven decision system at any stage of the UAV mission, especially to guarantee safety at mission planning and during the flight.

Indeed, when planning for a mission it is vital to rely on accurate and precise data of the environment where the mission will take. A high-resolution geospatial digital twin of the landscape coupled with the terrestrial network coverage, a suited channel model simulation, and the weather forecast allow the implementation of XAI-oriented models able to suggest an optimal path plan to successfully complete the mission. By referring to Fig. 5.1, the models are able to provide effective decisions to the different layers of the architecture. The model inferences can be effectively understood, and thus possible issues and authorisations can be mitigated and requested, respectively.

During flight, it is critical for the autonomous BVLoS UAV to have similar information and updated in real time, in order to ensure its cyber-physical spatial situational awareness, and thus achieving a safe mission. The current perception of the surrounding environment, connection performance metrics, and weather conditions are part of the intelligent sensing layer depicted in Fig. 5.1. They enable XAI-driven model inference explanations, influencing the intelligent control layer through the adoption of safer strategies. Such decisions and inferences, on the one hand, can be sent back to a ground control station as telemetry, in order to keep control of the situation and ensure mission safety at all times; on the other hand, the inferences history can be analysed by forensics in case of accident and hence determine its root cause.

To this end, this chapter envisions and introduces a key concept for mobile 6G network architectures: cyber-physical spatial situational awareness. In this way, highly-mobile network devices can safely operate in complex environments, realizing the use cases envisioned for the next generation of wireless communications.

5.1 Toward 6G spatial safety estimation

Cyber-physical spatial situational awareness for UAVs represents an important key-enabler for BVLoS flights [171], as it fuses data of different nature to provide a quantitative risk analysis for the geographical zone of interest, e.g., local

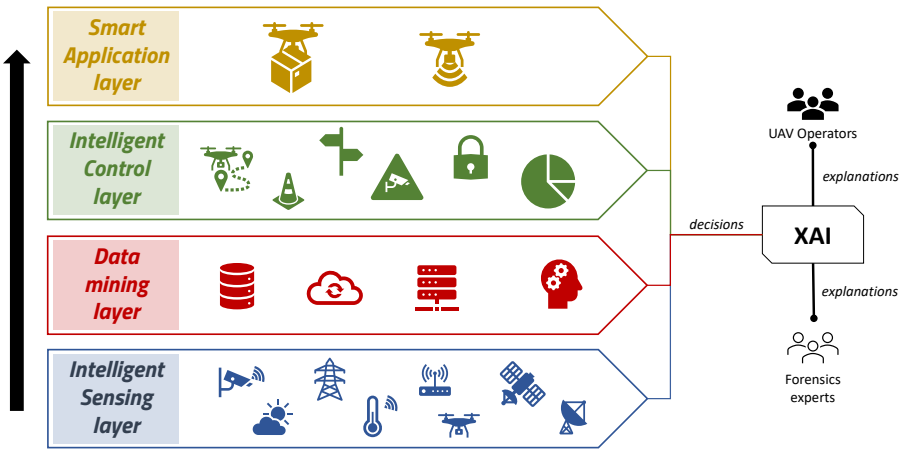


Figure 5.1: The architecture of AI-powered 6G networks for cyber-physical spatial safety estimation through UAVs.

orography, 3D city survey, mobile coverage, and weather historical and forecast data.

Static risk frameworks were initially proposed to assess UAV flight conditions for global [172] and local [171] weather conditions. However, the presence of skyscrapers influences wind turbulence [173] and mobile coverage, thus impacting on the risk of UAVs flights. Moreover, given the high number of features coming from different datasets and domains, an AI pipeline would be preferred to create a model that accurately estimates the flight risk. At the same time, AI models, especially ML and DL models, lack sufficient transparency that would allow operators to understand the key factors that characterise a risk over a certain area. To this end, XAI models are emerging to address such issue [169, 174].

At the time of writing, ML/DL models are mainly focused on coverage prediction and, even more specific, on terrestrial entities. A taxonomy of existing coverage prediction models is proposed in [175]. There are different categories of models that predict the coverage of wireless communication systems. The traditional ones are focused on statistical analysis, the study of physical electromagnetic wave propagation, or a mix of the two. Regarding the traditional models, the contributions in [176] and [177] focus on evaluating the coverage map of terrestrial network infrastructure for low-altitude and terrestrial UEs.

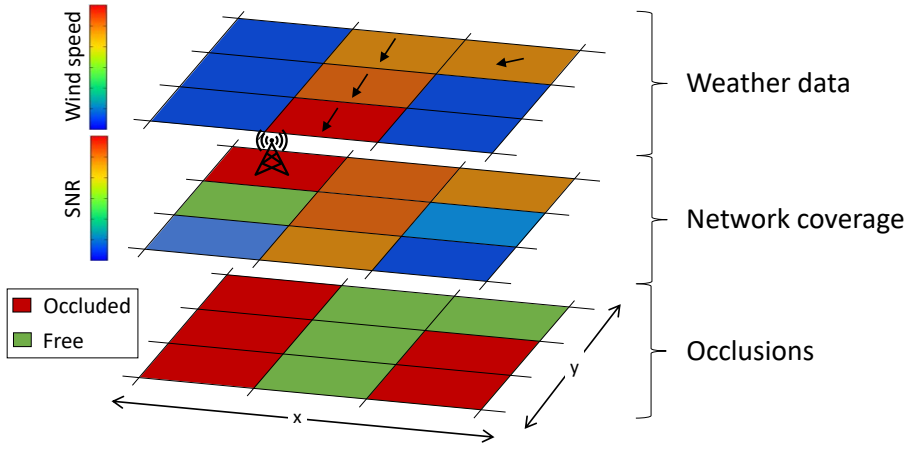


Figure 5.2: An overview of the layers of data to be considered toward XAI-enabled Cyber-Physical Spatial Safety Estimation for UAVs.

These works employ a high-resolution digital terrain model of 1 m and below combined with a georeferenced raster image layer of the Earth and a multi-class clutter layer to distinguish between the different types of environments. As for the BS-UE link, the authors considered the evaluation of free-space path loss from standardised models, taking into account diffraction loss, gas, and rain attenuation factors.

To compensate for the issues of these models related to efficiency, robustness, and efficacy, model-based and data-driven predictors are favoured [175]. While the former are empirical, propagation-based, and ray-tracing models, the latter are the emerging ML and DL ones. On the one hand, ray-tracing models are popular among commercial tools, as they provide accurate simulations, even though they are time-consuming and expensive to use [178]. On the other hand, RFs are known as the most suitable emerging supervised learning models for this task. In [179] and [180] the authors predict the received signal strength based on the transmission power of the cell, the channel model loss according to 3GPP standards, and a loss estimated by the ML model to address the discrepancies between empirical observations and channel model predictions.

First XAI-based prediction models are presented in [178] for different contexts. Specifically, a Light Gradient Boosting Machine (LightGBM) model, an implementation of the Gradient Boosting Decision Tree, is proposed for self-driving

terrestrial networks that, based on ray-tracing data, provide high-accuracy and faster results than classic ray-tracing models. Additionally, the model is enhanced with shapley additive explanations (SHAP), a well-known XAI approach, to provide explainability. The dataset is based on (i) calibrated ray-tracing data, (ii) a digital terrain model, digital height model, and digital land use map of 1 m of resolution, and (iii) UE traces. A key task during the data pre-processing phase is the *data gridding*, which maps UEs traces into spatial bins of 10 m of resolution, that is more coarse than the geographical surface model. *data gridding* compensates (i) the variability of fast fading in a small zone of interest and (ii) the positioning error of UEs traces, which are subject to human or GNSS error. This process simplifies the dataset and its variability, at the cost of quantization error and hence its accuracy.

Moreover, simulators also play a pivotal role for studying specific conditions, in order to gather more insights beyond the offered dataset resolutions. In this regard, ML-based path loss estimation is of recent interest in ns-3 [181] for more accurate path loss estimations, while a wind dynamics simulation is critical for evaluating wind turbulence in urban areas [173].

XAI methods will enhance cyber-physical spatial safety estimation for UAVs, by solving AI shortcomings. As shown in Fig. 5.2, which illustration has been fixed to a certain altitude for the sake of simplicity, the cyber-physical spatial situational awareness would benefit from the following data:

1. *Occlusions* – A reliable up-to-date digital twin of the surrounding environment, in order to know the obstacles, the orography, and relevant points of interest with geospatial attributes and metadata.

2. *Network coverage* – A detailed map of the radio access network coverage and SNR, in order to know blind spots and places where the cells are too overcrowded. Furthermore, the map can be enhanced with historical QoS, other performance metrics, and the presence of non-terrestrial access networks, maintained by UAVs, HAPs, and satellites.

3. *Weather data* – A detailed current and forecast data of the weather conditions, together with wind speeds and directions. A fine-grained processing of this information would ease the XAI-based model to quickly mitigate the situation.

5.2 Challenges

UAVs and AI emerge as key actors in 6G networks. However, they have several challenges that can be solved through the introduction of XAI, which are discussed hereby.

5.2.1 Mobility and dynamic environments

UAVs operate in highly dynamic and mobile environments, leading to frequent handovers and interruptions in connectivity due to obstacles. AI helps in predicting mobility and next UAV actions through a reliable and up-to-date digital replica of the surrounding environment. But, in this dynamic and changing context, characterised by the diversity of scenarios, XAI solves data discrepancy and allows valid and explained decisions on routes.

5.2.2 Resource management

Since UAVs have limited battery life, they require optimal energy management strategies, among limited network resources, e.g., in terms of bandwidth, transmission power, and energy consumption. Moreover, propagation delays between UAVs and UEs may compromise the real-time performance and responsiveness of the network. To this end, XAI can be essential for feature engineering and simplification, thus reducing the overall complexity of the model and, consequently, its execution.

5.2.3 Signal quality

Under complex and cluttered environments, it is necessary to consider interference and consequent negative effects on SNR and other QoS indicators. In this regard, XAI can take advantage from coverage maps and historical data to propose better UAV trajectories.

5.2.4 Weather conditions

Weather conditions, together with rapidly changing wind speeds and directions, impact the on decision-making processes and influence UAV missions. To this end, UAVs should quickly adapt to new scenarios. Weather predictions based on XAI are essential for planning UAV patterns and ensuring safe operations.

5.3 Compelling use cases and opportunities

UAVs are touching people's everyday life through experimental services and novel solutions, such as logistics, traffic monitoring and management, and first emergency response. Such services are decoupling from the traditional UAS: the pilot is becoming even more remote, which is not able to directly see the UAV, and must rely on the presence of modern communication network coverage to intervene when necessary. To this end, it is imperative to ensure that BVLoS flights are safe, reliable, and secure. When planning a mission, it is crucial to have a reliable knowledge of the environment the UAV will fly in. Is it a rural area, an industrial zone, or a smart city? How tall and close together the buildings are? Are there any relevant restrictions imposed by local and aviation authorities? Moreover, how will be the weather at that time of the day? How is the network coverage in that area for real-time control and high-quality telemetry? Are there any dead spots and crowded or high-latency areas? Is GNSS signal reliable? Are there any nearby assisted GNSS stations to ensure high-precision navigation? These are all questions that rely on historical and forecast data during both the planning phase and in-flight. For instance, was the forecast data a good estimation of the actual situation of the environment? Are there any changes of plans that the UAV should take to ensure a safe flight and a successful execution of the mission plan?

This context leads to complex non-linear problems to solve, for which DL and specifically neural networks (NNs) are key in providing an acceptable trajectory plan and mission design, as well as taking flight decisions. It becomes relatively easy for the UAS operator to not trust the NN decision flow, considering the resulting output as "abnormal behaviour". This leads to a loss of trust in the model by the UAV mission designers and operators. This is due to the fact that the opacity of NN models does not allow to understand what input and data are actually influencing the final output decision taken by the model itself.

As anticipated, XAI represents a key enabler for safe UAV flights in such complex environments. In these complex situations, which span multiple areas of vast knowledge, data, parameters, and expertise, it is crucial that the planning operators, as well as the remote UAV pilot, get an indication of what causes a particular UAV trajectory, which may require proper authorisations to access the air space of interest.

XAI positively affects UAV spatial safety estimation under both aerial traffic

management conditions: structured and unstructured. On the one hand, structure traffic management is the basis of the Internet of Drones theorised by Gharibi et al. [22], which proposes a structured and regulated airspace through the definition of "airways", i.e., aerial paths that the UAV must take to ensure proper regulatory compliance. A path plan optimised with the help of XAI feedback may improve the airways to take in order to safely reach the destination. On the other hand, unstructured traffic management relies on the suitability of the mission path planning, which should still respect all the applicable laws that regulate the airspace, and the precision of the UAV in following such plan, even under external disturbances.

Last-mile delivery is being treated as the playground for these ideas, as drones are effectively used to access urban air space in order to deliver small and lightweight packages. At the time of writing, commercial services cover only sub-urban environments, where is it quite common to find typical American family houses with a green just outside the door. This ensures spatial safety by nature, as small UAV navigational errors do not impair the overall success of the mission. Furthermore, being in an area with low-altitude houses allows easier wind mitigation and facilitates the LoS with nearby cell towers, ensuring optimal communications anytime.

On the contrary, extending such service to city centres requires even more stringent spatial limitations. UAVs must be extremely precise in their movements, as they have to deliver packages to specific parts of the city and, as a worst-case scenario, on balconies, which may be bounded by nearby buildings and other obstacles. Wind gusts are harder to predict and mitigate, requiring precise control of the UAV, and connectivity may be affected by shadowed and/or overcrowded zones. In this situation, being able to quickly understand what hinders UAVs spatial safety is key in taking a prompt response. XAI represents a compelling opportunity to ensure that such perturbations are promptly identified and help to justify the adoption of a behaviour by the decision maker, be a NN model as well as a remote UAV pilot.

A FBS is another promising use case that unlocks the integration of TNs with NTNs. The final goal is to employ aerial actors, such as UAVs, in zones whose connectivity is poor, overwhelmed, or simply lacking, ranging from big events to rural zones and disaster areas. The FBSs act as a BS, or they just employ intelligent reflective surfaces, to serve traffic on the ground, while repeating the signal to other far-reaching entities, such as terrestrial BSs, HAPs, or satellites.

Given that the UAVs serve as a critical component for the radio access network, cyber-physical spatial awareness is vital to ensure optimal connectivity and service continuity to ground UEs. The fusion of information between the environment, coverage, weather maps, and their forecasts plays a key role in finding spots that are best suitable for the UAVs, guaranteeing flight energy savings and improved network coverage and quality, as long as it is safe to do so.

5.4 The road ahead

Cyber-physical spatial safety for UAVs represents a critical component for future 6G mobile communications, as the upcoming network infrastructures are intrinsically related both to the cyber world, as in network coverage and its integration, and the physical one, as in environmental occlusions and interference by the weather conditions. To this end, UAVs necessitate of reliable and trustworthy decision-making models to guarantee safety throughout the mission. As current AI models are affected by their opacity and inability to insights into their inferences, XAI-based models promise to overcome such issues. To this end, their application to UAV mission would greatly benefit safe flights and troubleshooting.

Given the early research of this work, it is suggested to provide reference XAI models integrated with the 6G architecture and related evaluation metrics, along with reliable and accurate data from environmental obstacles, network coverage, and weather forecasts. Furthermore, as network planning requires accurate forecasting, network simulators play a pivotal role for their execution. Differently, hardware-in-the-loop simulations, emulations, and novel network software are required to apply the same concepts during flights.

Finally, for what concerns the standardisation efforts in AI, 3GPP activities are focusing on enhancing existing specifications from Release 18 onwards [91, 96, 98, 99, 182–203]. Specifically, work is being done (i) to manage the capabilities of AI models across the 5G core network and its RAN [204]; (ii) to transfer a model across a 5G core system [93]; (iii) to employ AI models for enhancing channel state information feedback, beam management, and improving position accuracy [205].

6 Counter-unmanned aircraft systems

This chapter presents an initial study aimed at gaining a deeper understanding of C-UAS, including their operational context, characteristics, and challenges as identified in the state-of-the-art literature. Additionally, it offers insights into potential future research directions.

Drones are sometimes referred to as UAVs or remotely piloted aircraft systems, but the term used in European Union (EU) legislation [206] is UAS. According to EU legislation, UAS refers to the whole system comprising pilot, operator, data-link, and unmanned aircraft (UA).

UAS are disrupting sectors of the aviation industry with affordable and easy access to airspace, influencing and impacting a wide range of applications [207], e.g., urban air mobility, logistics, pharmaceuticals, last-mile delivery, emergency reconnaissance, agriculture, mapping, and surveying. UAS have unlocked an easy and affordable access to low-altitude airspace, further improving communications in overcrowded areas and in smart cities, where high-altitude NTNs, i.e. HAPs and satellites, are not as effective as drones and TNs. To this end, UAS are also becoming key actors for the realisation of future generation networks, 6G goals, as they blend together NTN services with TNs, as illustrated in Figure 1.1.

However, drones can also be used for illicit and hazardous activities, such as smuggling, harassment, trespassing private properties, accessing public sensitive areas, cyber-attacks, and terrorism [208]. Addressing non-cooperative drone threats is crucial for legitimate drone usage. Drones can be classified according to their non-compliance nature (criminal, unlawful, or amateur.) The growing use of drones in warfare theatres, recent terrorist attack plans, and the use of malicious drones demonstrate these risks. Suspicious drones near critical infrastructure and public spaces indicate potential hostile intelligence gathering. Criminals employ drones for smuggling and facilitating illicit operations such as drug trafficking and prison contraband.

EU legislation [206, 209] has been implemented to ensure safe flights and identify responsibilities. BVLoS flights are emerging as a key step towards autonomous flights, that is, without human intervention. These flights represent

a stepping stone for the adoption of UAS at scale, beyond the current use cases for both private and enterprise use, monitored by unmanned traffic management systems. However, this technology can easily be exploited by malicious actors, causing serious disruption to public spaces, crowded places, and civil buildings [210].

To this end, C-UAS solutions must be deployed in sensitive areas, in order to protect society and critical infrastructure. C-UAS implementations carry out DTI of unauthorised UAs accessing the protected area, also known as the "scan area". They are able to do so thanks to a combination of different technologies, ranging from electromagnetic spectrum monitoring to audio sensing. While civil C-UAS implementations typically focus on DTI, military ones normally extend this concept to provide counterattack features for UA neutralisation and forensics procedures for post-hoc analysis and intelligence [211]. While C-UAS is a topic in its infancy, multiple problems and solutions can be translated from other domains, such as traditional aviation, satellite imaging [212], biomedical, automotive, and IoT [213]. C-UAS solutions deal with vast amount of data; they handle a large data volume across a variety of sensors, which can be generated at different velocities, provide a form of value toward the success of DTI, also influenced by their quality and accuracy, i.e., veracity [212]. Furthermore, data are (i) from multiple sources of different nature, (ii) high-dimensional, especially for hyper-spectral sensors, (iii) subject to multiple scales and different spatial resolutions, and (iv) isomeric, i.e., drones may be represented under different data structures, such as vector and raster. Moreover, their relationship can be non-linear. These aspects introduce networking and synchronisation problems to guarantee sufficient bandwidth and timing for the elaboration of big data. Furthermore, for distributed and wireless sensors, C-UAS inherits the issues related to remote sensing, further discussed in [212].

In the context of EU policy, the EU has adopted common rules in the field of civil aviation that include the use of UAS in European skies [214]. Further measures were established under the Commission Delegated Regulation (EU) 2019/945 on unmanned aircraft systems [206] and Commission Implementing Regulation (EU) 2019/947 on the rules and procedures for the operation of unmanned aircraft [209]. Moreover, the EU has taken significant steps to address the security threats posed by drones, including the Action Plan to support the protection of public spaces [215], the EU Security Union Strategy [216] and the Counter-Terrorism Agenda [217]. To boost research and development in

this regard, EU funds have been allocated for drone technology improvement, including C-UAS [208]. Finally, in 2023, the Commission adopted a new Communication specifically addressing the unlawful and dangerous use of civil drones [218].

6.1 Related work

While C-UAS research and commercialisation are rapidly expanding [213, 219–224], current state of the art lies its foundations in [225], which has been of great use in the aviation industry to DTI multiple aircraft. Furthermore, such principles can be extended to UAS vulnerabilities and their potential intrusion strategies [226] which are key for their DTI [227].

The European Commission's Joint Research Centre (JRC) is developing C-UAS knowledge and capability via implementation of its DRONE project [228] and has published two handbooks on physical hardening [229] and protection of critical infrastructure and public spaces [230]. In addition to setting up a Living Lab[21] in Geel, Belgium, the JRC has a framework contract for C-UAS services for research and protection of infrastructure of Commission premises in Geel and Brussels.

The COURAGEOUS project[22], launched in 2021, focuses on developing a standardised test methodology to DTI nefarious drones using countermeasure systems to protect the low-altitude airspace. The project aims to create a fair qualitative and quantitative comparison between different C-UAS by establishing a series of standard user-defined scenarios representing a wide range of use cases. By addressing the need for a standardised test methodology, the COURAGEOUS project significantly contributes to advancing the field and ensuring effective protection of low-altitude airspace from unauthorised drone usage. In [223] the authors provide an overview of the trends for the design and the implementation of C-UAS solutions. RF, Radio Detection and Ranging (RADAR), and visible-light sensors are preferred, whereas acoustic, Light Amplification by Stimulated Emission of Radiation (LASER)-based, and IR ones are more niche. While RF

21 Living labs are a modern way of creating user-centred environments that enable innovation, co-creation and start-up development. https://joint-research-centre.ec.europa.eu/living-labs-jrc_en

22 The COURAGEOUS project is funded under EU's Internal Security Fund Police (ISFP) under Grant Agreement 101034655, https://courageous-isf.eu

and RADAR are more likely to be used stand-alone, some solutions prefer the combination of RADAR, visible-light, and thermal sensors.

A comprehensive overview of cooperative scenarios, in which a C-UAS implementation is augmented with the use of UASs for patrolling, can be found in [231]. This contribution also addresses neutralisation techniques to mitigate airspace breaches. A crucial aspect for this extended system is a robust wireless network that is able to not interfere with the counter system and is able to defend itself from external attacks. The challenges in these aspects can be found in [23, 227, 232]. More generally, in [233] Laudias et al. provide a review on network localisation, tracking, and navigation, in particular on cellular localisation systems and solutions based on wireless local area networks.

DL is also enhancing DTI in recent C-UAS implementations, which techniques are studied in [234–236]. Among the several DL architectures, the multi-layer perceptron neural network, well known for its classical input, several hidden, and output layers, is the most used for this application [237], especially for RADAR-based DTI [238].

It is critical for such DL systems to be as efficient and as accurate as possible for the DTI of unauthorised UAs. To this end, the development of algorithms able to do DTI in challenging conditions is a must to ensure the reliability of the system [212]. Currently, it is challenging to have reliable and fast DL models if the data is subject to imbalanced and noisy labels. Furthermore, the data must be partitioned, loaded, and the pipeline analysis task properly scheduled to balance the computing and networking resources, and thus keep the entire system under the real-time requirement for a prompt protection of the scan area. For an efficient use of the DTI models, proper feature selection has to be made to filter, embed, and wrap data before processing. Consequently, ML and/or DL models need to be adapted and tuned to the necessities of C-UAS solutions, especially in terms of fast data ingestion, their processing, and the ability to work with a continuous data stream. Data may be missing or noisy, as not all sensors are in their ideal conditions in a certain time frame, especially caused by natural phenomena, e.g., fog, mist, or direct sunlight. Furthermore, imbalanced data impact classifiers, for which proper techniques must be implemented to mitigate the issue. Finally, on-premises solutions can be part of a distributed and/or federated system of C-UAS implementations all-over the world, e.g., the ARTAS system in the case of airspace monitoring for traditional aviation [239],

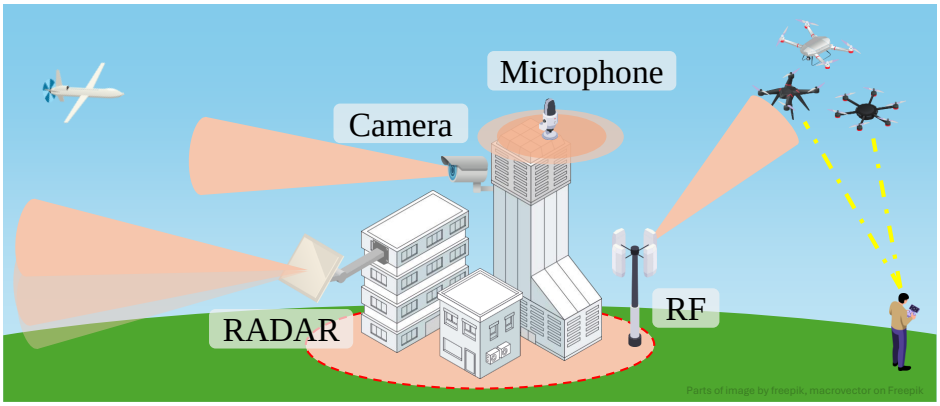


Figure 6.1: Overview of C-UAS sensors employed to defend a sensitive area, made by buildings and highlighted in red with dashed red border. Apart for the *RADAR*, a generic antenna has been chosen for *RF analysis*, while a camera as an example of an *electro-optical* sensor and a microphone for *mechanical wave analysis*.

in order to improve models through transfer learning and/or provide a broader early warning system to protected areas that are close together.

Given the implications of such systems in public and sensitive areas, regulations are being developed around the world, particularly in the EU, with the UAS regulatory framework [206, 209] and the United States with the Federal Aviation Administration (FAA) regulations [240]. Furthermore, preliminary information can be seen for the integration with the upcoming Air Mobility market [241]. Given all these aspects, C-UAS design and sensor implementation are critical aspects for their DTI of UAS.

6.2 Overview of C-UAS sensors: taxonomy and challenges

Sensors can be categorised in different ways [212], as they (i) sense electromagnetic waves or mechanical ones; (ii) work on specific bands of the spectrum; (iii) analyse a signal being generated by the sensor itself or scan for signals propagating in the environment, and thus be classified as active or passive, respectively; (iv) operate in different working conditions; and (v) be installed on different platforms. A general overview of a C-UAS system is depicted in Figure 6.1, while

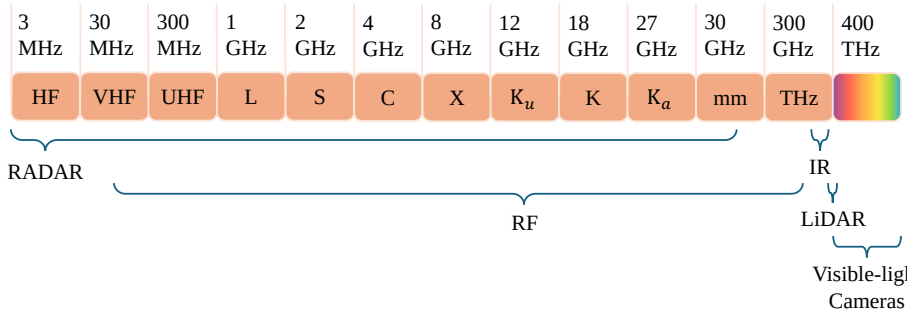


Figure 6.2: Overview of the types of sensors used to monitor the electromagnetic spectrum. Spectrum bands according to IEEE standard [242], with the application of Table I Note 9 for brevity purposes. Spectrum bands are extended beyond the standard to include visible light.

a sensor categorisation employed to monitor the electromagnetic spectrum is illustrated in Figure 6.2. Each combination of these characteristics designates a subset of sensors that brings a trade-off in terms of detection accuracy, cost of implementation, latency, and cost of computational resources, and thus it affects the quality of the data and influences the overall performance of the C-UAS solution.

Finally, sensors can be classified by their platform of reference [212]. In-situ sensors can be fixed, mounted on a tower, or placed on the ground. They can also be mobile, as they can be mounted on cars or carried by humans. Sensors can also be airborne, as they can be mounted on collaborative UAs, i.e., drones that extend and enhance the C-UAS DTI task. Given their variable altitude, their data acquisition may be subject to less clutter and interference.

6.2.1 RADAR

There are several types of RADARs, which offer different trade-off in terms of performance, technological readiness, economic feasibility, and limitations. These sensors analyse a radio signal that has been initially transmitted by the sensor itself and then backscattered by the environment of the scan area and, possibly, by the targets of interest. In typical scenarios, the amplitude and phase shift of the returned signal describes the intensity and distance of the target from the sensor, respectively. Furthermore, the Doppler effect can be analysed,

providing an estimation of the target velocity and its distance from the sensor. In [243, 244], the authors presents a specific study made for the application of RADARs for C-UAS implementations, current mmWave experimentation's are summarised in [245] and micro-Doppler signatures are discussed in [246].

A key advantage of RADARs is that they tend to work better than the other types of sensors at multiple weather conditions, even though they are dependent to the attenuation of rain, i.e., the heavier the rainfall and the higher the operating frequency, the higher is also the signal attenuation experienced [223].

UASs can be of various size, with a wingspan that greatly ranges from 61 m to 1 mm [247]. To this end, the sensor choice would be firstly focused on their covered wavelength, which typically varies between 1 mm, i.e., 300 GHz, and 100 m, i.e., 300 MHz [212].

Current market trends reports that a significant number of C-UAS implementations do not report the range capability of their RADAR, i.e., 45%. The solutions focus more on DTI UASs in the $3 \sim 5$ km and $7.5 \sim 10$ km ranges.

Even though RADARs are compelling sensors to be included in a C-UAS implementation, considerations related to the spectrum regulation, the maximum power allowed in a certain area, interference, blind spots, resolution, and position are key aspects that can heavily influence their overall performance [223].

Among several RADARs types and techniques to DTI UASs, automatic target recognition (ATR) [248] emerges as a theoretical proposal to identify UAS characteristics from the echo signal and then further classify it. Even though ATR has achieved significant progress thanks to DL-based approaches, there are open challenges to overcome and real-world testing needed, among them the need for a dataset large enough for proper detection and recognition. Furthermore, with Explainable AI (XAI), explainable models are in development for secure, trustworthy, and robust DTI.

Conversely, synthetic aperture RADARs (SARs) represent the most common RADARs. Their main advantage is the possibility of being installed on movable platforms, and thus it provides an enhanced aperture of the acquisition, hence its name. Furthermore, they provide high-resolution acquisitions independent of the time and weather condition. Among SARs, ATR is also being considered [249–251]. Another major challenge highlighted is the real-time requirement of C-UAS, which is often neglected in research and imposes strict limits on the complexity of the neural network, the ability to be compressed in smaller forms, and the ability to quickly elaborate data fragments acquired in a brief time

interval. Furthermore, it is possible to obtain high-resolution images through static synthesis of aperture and by imposing a circular movement to the sensor to obtain the best results [252].

6.2.2 RF analysis

An UA is typically part of a much broader and rich system of UAS, with the intent to offer remote control, telemetry, and application data stream to a remote pilot or a ground control station [213]. To this end, UASs are subject to electromagnetic radiation for the transmission and reception of signals from / to the remote pilot, as well as Internet and localisation services, such as GNSS. To this end, multiple software-defined radios (SDRs) can be employed to detect the radio activity of UASs and determine its location through the direction of arrival. Signal processing algorithms have been proposed for an accurate direction finder requiring just two pairs of antennas [253].

Traditional RF-based analysis techniques and Wi-Fi signal interception are discussed in [254], where it is highlighted that it is a challenge to eavesdrop Wi-Fi traffic in open field, especially if the scan area is larger than 3 km of radius, requiring expensive high-gain antennas or ad-hoc networks. Furthermore, there is the need to build a comprehensive dataset of communications data recorded from different models of drones, in order to ease their detection and classification. The current state of the art focuses on laboratory tests with one single drone being analysed, thus severely limiting the research for a C-UAS solution able to DTI multiple UASs at long distances and to discriminate between legit drone communications, other types of communications, and fake traffic to confuse the C-UAS detectors. Finally, circumvention of Remote ID [255–257] represents another major challenge, in particular in relation to privacy safeguarding and regulation.

Passive RADAR methods for detection of drones are possible with SDRs and other receivers integrated in a heterogeneous bistatic RADAR setup. This is most useful in case of autonomous drones with no radio activity. These methods process reflections from non-cooperative sources of illumination, e.g., Wi-Fi, radio, digital video broadcasting - terrestrial (DVB-T) signals, in the environment to detect or localise moving objects. However, it should be noted that these methods do not specifically discriminate drones from other moving objects, which can result in false alarms. In this case, third-party radio activity such as nearby Wi-Fi stations, radios, and cellular network can be analysed, along

with their interference, to understand the presence of autonomous UAs. In this regard, for example Deng et al. demonstrated that their solution is able to DTI drones using the channel state information from Wi-Fi with interference to such signals caused by shifting, moving, and spinning propellers of a drone [258].

One signal that is present all around the globe is GNSS [212], emitted from a constellation of satellites to provide GNSS-based location. Reflectometry techniques can help to analyse the surrounding environment, and thus to detect moving objects, i.e., UAs.

Simultaneous transmission and reception at the same frequency would ease RF monitoring deployments, as they allow less SDRs to transmit and then analyse the surrounding interference, acting as full duplex radios with filtered self-interference [259]. For this case, tests were conducted in the military sector in the 225 $\sim$ 400 MHz range up to 100 W of transmission power and in the civil cellular infrastructure for in-band 5G communications at 2.4 GHz [260].

DL-augmented neural models provide superior DTI accuracy [261], as it has been demonstrated that it is possible to differentiate between different drone types, as well as their flight setup. This is due to the fact that it is possible to extrapolate frequency-related features that contain more information about the incoming UASs in respect to other types of sensors.

6.2.3 Electro-optical sensors

Active sensors in this area are typically dominated by the presence of Light Detection and Ranging (LiDAR) devices, which employ pulsed LASER beams for ranging [212]. Inversely, passive sensors of this category are further distinguished between common visible-light cameras, IR cameras, multi-spectral, and hyper-spectral, with the last two focusing on a few discrete bands and many narrow and contiguous bands, respectively [212]. Moreover, these cameras may be augmented in a pan-tilt-zoom (PTZ) setup, allowing three degrees of freedom in terms of movement and zooming [223].

Electro-optical sensors may benefit from the recent advances in DL-aided computer vision systems. A comprehensive overview of the models applied to these sensors can be found in [262]. These proposals are severely limited in C-UAS solutions [223] due to (i) their dependence on light, especially at night, (ii) the lens choice, which imposes a certain focal length and the maximum zoom distance, (iii) the sensor resolution, which limits their detection range, (iv) adverse weather conditions, and (v) dirt on the sensor. Regarding light sensitivity

and adverse weather conditions, short-wave IR are promising solutions [231]. Moreover, it is challenging to accurately determine the UAS coordinates, which may require at least two cameras in stereo configuration, a recent DL model for monocular ranging [263], or the coupling of a LASER for ranging. Finally, the PTZ benefits are limited by their speed of movement and variable blind spot when zooming [223]. Among these difficulties, there is also the need to find the right trade-off in terms of timing and accuracy. Furthermore, the need for a large dataset may be compensated with few-shot learning, allowing the models to detect and classify new drone models via a few examples [262].

6.2.4 Mechanical wave analysis (acoustic sensors)

Sound Navigation and Ranging (SONAR) is not suitable for detecting and tracking targets using sound waves. Sound waves travel much more efficiently in water than in air, as the former is denser than the latter and it provides better acoustic coupling. Consequently, sound waves quickly lose energy in air, significantly limiting the effective range of SONAR.

However, microphone arrays have been successfully used for DTI given the typical rotor blade noise of UAs [213]. An array of microphones is deployed inside and around the scan area to evaluate the Time Delay of Arrival (TDoA) in real time. Even though the system offers high accuracy detection and localisation in ideal conditions, it is severely limited by adverse weather conditions and environmental noise, which typically surpasses the rotor noise produced by UAs. To this end, acoustic-based DTI, both in its active and passive forms, are not typically favoured for C-UAS implementations [213] but should be used in specific conditions.

6.3 Multi-sensor data fusion for counter-unmanned aircraft systems

Given the limitations of individual sensor types, a C-UAS solution that relies solely on one type of sensor would be significantly constrained in its monitoring capabilities [264]. A more effective approach involves adopting a system architecture that benefits from multiple sensor types and implements a multi-sensor data fusion algorithm. This approach integrates the features of different sensors

to enhance the overall accuracy of DTI, even when individual indicators are incomplete or uncertain [265].

Data fusion can be applied at different levels: (i) at feature level, aiming to integrate different sensors data; (ii) at the decision level; and (iii) at the rule level, which is the simplest approach for integrating a variety of sensors but sacrifices flexibility and adaptability [264, 266]. Multi-sensor data fusion presents an ongoing challenge, with the goal of providing a unified representation of these data within C-UAS implementations [236, 267]. Major challenges in multi-sensor data fusion solutions include addressing sensor heterogeneity, managing misalignments between data acquisitions, and handling the high volume, variety, velocity, and veracity of multi-modal data, which often exhibit complex inter-modal correlations [266].

Moreover, it is essential to identify a combination of sensors and classifiers that can produce consistent results or support DTI, even if some sensors fail to detect certain targets (e.g., autonomous UAs immune to passive RF monitoring.) The goal is to avoid combinations that weaken the model's effectiveness for UAS DTI. Currently, C-UAS solutions on the market combine RADAR with electro-optical cameras for independent tasks and integrate different types of cameras or RADARs with passive RF devices to improve accuracy on the same tasks. Although qualitative analyses have been conducted by the research community to evaluate the overall system's reliability [223], quantitative analyses that accurately dimension an effective C-UAS solution remain scarce.

Despite the potential benefits, current C-UAS solutions on the market often lack transparency regarding the specific sensor fusion techniques employed. While some may briefly mention the use of ML or traditional statistical methods, the limited disclosure hampers a comprehensive understanding of their limitations and potential areas for improvement. Although the combination of multiple sensors is still a niche practice, it is hindered by the complexities involved in merging and fusing multi-sensor data within such sophisticated systems [223].

A meta-study analysis of the scientific literature has been conducted to evaluate the state of the art and current challenges regarding multi-sensor data fusion algorithms for C-UAS. The study has been limited to the DTI task, considering also the classification of UAS, as it fits the scope of C-UAS employed for civil purposes. Furthermore, the study was further limited to "static C-UAS" tech-

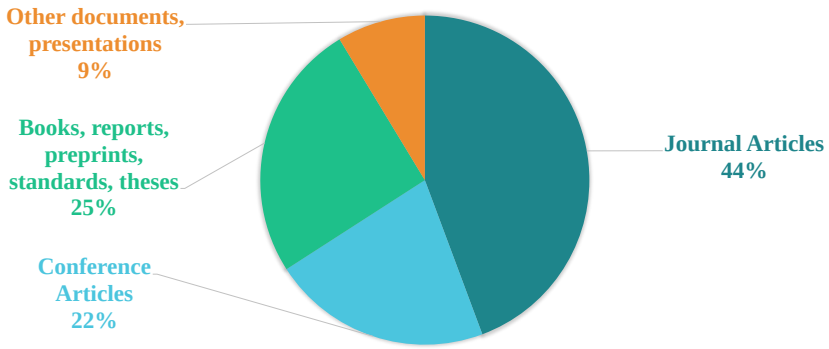
niques, as in C-UAS implementations that mainly used sensors fixed in place, instead of mobile, human-packable, and collaborative aerial ones [268].

In total, 519 documents have been retrieved through the European Commission's eLibrary and CORTEX tools, which allowed the access to works published in book, conferences, and journal articles, as well as EU-funded project deliverables and standards. As can be seen in Figure 6.3, most of the documents analysed were journal articles, accounting for a total of 228 documents. Although foundational techniques to DTI aircraft have been researched since 1980, most of the literature focused on C-UAS was produced in the last four years, with 80 journal articles published in 2023 alone.

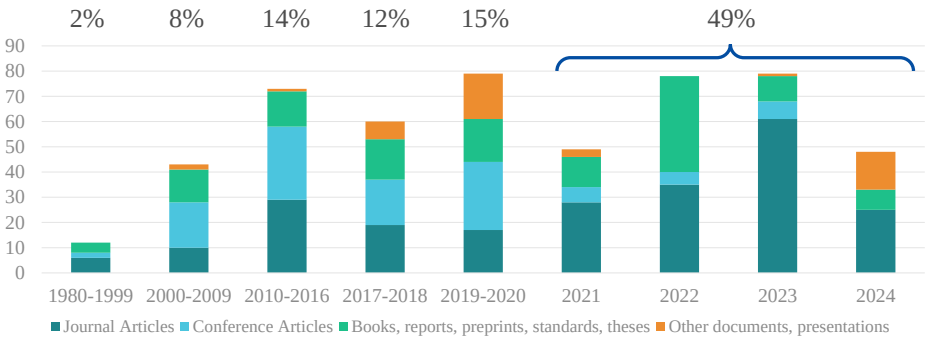
A fundamental takeaway from such a literature research is that the majority of C-UAS prefer the use of single type of sensor, even though multi-sensor systems are reportedly growing over time. An open challenge is finding ways for real-time multi-sensor data fusion, as it is currently one of the major limitations of the system scalability over the number of sensors and their types to be deployed. Furthermore, it is not always demonstrated convincingly that an increase in the number of sensors delivers an effective increase in accuracy and precision in DTI tasks.

Multi-sensor data fusion algorithms rely on key assumptions related on how architecturally close they are to their specific sensor features, i.e., early versus late fusion approaches [269]. This allows the multi-sensor data fusion to be treated as a composition of multiple models, promoting the use of DL-oriented models to analyse sensor raw data of a single type of sensor, e.g. YOLO for detection in electro-optical sensors [270], together with high-level models that focus on tracking and decision-making based on pre-processed and filtered features.

Current proposals can be divided into three major categories: a classical approach, which relies on traditional Bayesian probability, a modern one, mainly focused on the multi-layer perceptron neural network, and a hybrid version that mixes both approaches. Classical multi-sensor data fusion is based on tracking techniques that can be found in the aviation industry, especially from the efforts done by the Netherlands Aerospace Centre (NLR) to create the ATLAS system, currently employed to protect the European airspace. These are algorithms based on interacting multiple model (IMM) and probabilistic data association (PDA), as illustrated in Figure 6.4, plus multi-hypothesis tracking (MHT) and clustering.



(a) Types of documents.



(b) Distribution over time of the analysed documents.

Figure 6.3: Statistics of the 519 documents analysed for the meta-study analysis on multi-sensor data fusion algorithms for C-UAS.

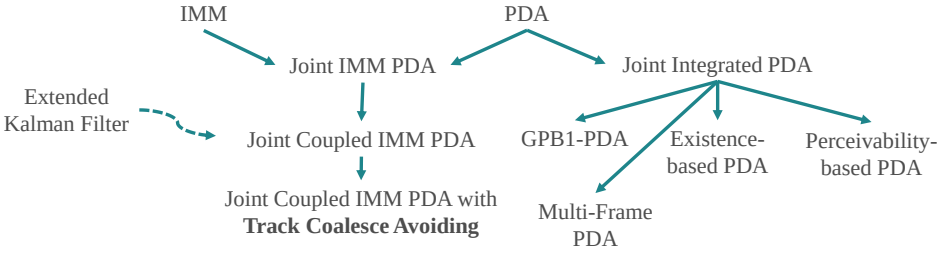


Figure 6.4: Relationship between the different algorithms of multi-sensor data fusion found for drones' DTI.

The main issue of these algorithms is their complexity, which hinder their possibility to be executed in real time, and their risk on merging close (but distinct) tracks. As illustrated in Figure 6.4, IMM and PDA are typically merged in order to get the advantages of the two, enhanced with safety mechanisms to avoid unwanted track merge, even under partial and missing data.

Conversely, MHT is one of the pioneering algorithms to use hypotheses (i.e., models) to update and maintain a track. It is a way to do model fitting. The advantage of MHT, which help its adoption in aircraft tracking, has been the model itself. Aircraft trajectories can be easily modelled as jump linear systems. While such an assumption may be applicable to fixed-wing UAs, this would not be the case for multi-copters, which are way more flexible and versatile than traditional aircraft and helicopters, and hence are more unpredictable, by introducing non-linearities in their movement. Thus, the MHT requires extensive expert knowledge in order to create efficient and accurate models, i.e., hypotheses, to be associated to the drones.

At the same time, neural networks architectures are quite flexible and can be adapted and tailored for the specific task of early or late fusion. The issue is that these NNs (with the MLP network being well-known) are opaque in nature (i.e., they do not give us clear causal information between inputs and outputs.) Furthermore, they need comprehensive (multi-modal) datasets in order to raise their trustworthiness and reliability. Alternatively, novel approaches of XAI may positively impact in this regard by providing simpler models and a way to explain the relationship between model outcomes and the input features.

It is critical to evaluate the success of DTI through a common framework of KPIs, in order to compare different multi-sensor data fusion techniques. Possible KPIs would be, for each sensor type, the detection probability at different distances, e.g., 20, 50, 80, 100, 1000 meters and beyond, along with the false alarm probability [271]. The lack of a uniform way to test multi-sensor, data-fusion-enabled C-UAS remains a barrier to measuring the effectiveness of such techniques.

6.4 Key challenges and the road ahead

With the emerging use cases of the society that mandate personalised, faster, safer, and more reliable and secure cyber-physical systems, UAS are becoming key enablers for the re-evaluation of transport, communication, and monitoring

solutions. Together with the new ecosystem of technologies proposed by 6G networks, UAs have the potential to be employed for large-scale BVLoS missions, traversing rural and urban areas, and thus simple and complex environments, which poses challenges for both the physical and the cyber worlds. As with all technologies, their improvement brings tremendous benefit for the society, but also new types of threats, which should be readily discussed and mitigated through proper defensive system, to protect society and institutions from the misuse of drones by malicious actors.

At the time of writing, the research on multi-sensor data fusion-enabled techniques for enhanced C-UAS implementation is limited by the following key points:

1. There is a lack of a unified framework for joint state estimation and data association identification [272].

2. Proposals should be validated on fulfilling the architectural challenges of C-UAS, i.e., a distributed system of sensors limited by bandwidth, timing, and spatio-temporal synchronisation requirements [266].

3. More research is needed on multi-modal range and direction fusion for accurate localisation [273], while minimizing association with estimation errors, which leads to performance improvements [272].

4. Specifically for the group of sensors related to electromagnetic spectrum analysis, sub-7 GHz analysis is more challenging in complex environments than mmWave, as Rician, Rayleigh, and Nakagami channel models demonstrate [274]. However, such spectrum portion remains key for non-line-of-sight detection and tracking.

5. For ML, unsupervised-based multi-sensor data fusion becomes computationally intractable as the number of targets increase [275].

6. More research is needed on finding an optimal trade-off between tracking accuracy and inference time [276]. This also means building a lightweight and high-performance data processing architecture.

7. Further research should be done on reliably filter out sensor noise and bias in a heterogeneous environment [266].

8. Currently, there is a lack of a comprehensive and high-quality dataset for multi-sensor data fusion-based C-UAS [266].
9. Further studies are needed to achieve consensus among multiple, and potentially conflicting, sensor reports [277].

7 Authentication and authorisation in modern digital services

As UAVs, HAPs, and satellites are integrated in a T/NTN, they become part of a more reliable and ubiquitous global network, while at the same time unlock services never seen before. As it has been seen, such cyber-physical architecture that is emerging must be dealt with careful security and safety considerations, in order to ensure that such technology is not misused and does not harm infrastructure and society.

As these emerging network entities become a platform for services, would it be in the form of a network slice or a digital resource for smart cities, it is imperative to understand how to protect modern digital services. They are increasingly designed and implemented according to micro-service patterns, by composing (also denoted as *chaining*) digital resources (i.e., data, networks, cloud services, applications, and things) from multiple vendors on a growing scale; this allows to create, process, share, and consume data and content in a digital continuum, blurring the frontiers between application domains and breaking the current closed silos of information. The loosely-coupled nature of micro-services allows to replace, duplicate, or remove part of them without affecting the operation of the overall application. As main result, digital services can be quickly provisioned in a matter of minutes or hours instead of days or weeks, can grow or shrink dynamically according to the evolving workload, can be easily deployed, replicated, and migrated in multiple locations and even over heterogeneous infrastructures. This perfectly fits the dynamic and agile nature of T/NTNs and matches modern business models, where digital services and business chains are expected to emerge, evolve, and dissolve much faster than traditional value-creating networks [7].

Unfortunately, cybersecurity mechanisms have largely been unable to keep the pace of this evolution towards data-oriented models. Indeed, they are still largely conceived to protect individual devices and specific infrastructures, either physical or virtual, but are largely ineffective to cope with complex dynamics, dispersion of data among the multitude of digital objects and infrastructures, non-deterministic, opaque, and partially inscrutable service topologies [278].

This is more difficult when resources from multiple providers are involved (e.g., cloud, storage, things, and heterogeneous networks) [279]. Similarly, detection of cyberattacks in these conditions is extremely challenging, due to the lack of proper visibility over third party's infrastructures and services. The interconnection of digital resources from the aforementioned heterogeneous domains creates complex systems of systems, which introduce more functions, management aspects, and security issues than the plain sum of the constituent components [280, 281].

This requires scalable AA schemes that give security operators limited access to digital resources (i.e., security capabilities) on behalf of the resource owner. Token-based mechanisms like OAuth2 [282] provide an effective solution to this problem, by avoiding the need to share secrets and to grant full control over the resources.

Although standard firewalling, antivirus, and intrusion prevention systems implemented by Resource Providers may be enough for some users, they are not probably enough for detecting and mitigating complex attacks [283].

To this end, the CSF proposed in [284] can be a viable solution. It was developed into the H2020 GUARD project. Its aim was to collect security-related events and measurements from dynamic and evolving systems and infrastructures of the Information and Communication Technologies (ICT). Data are collected and categorised by distributed third-party software agents. The power of such a CSF resides in the integration of different modules, external and internal to the platform, accomplishing heterogeneous and articulated tasks through several logical interconnections [278, 284, 285]. As a consequence, there is the need to protect information about service features, users data, and exchanged information from unauthorised access, so to guarantee a high level of security among multiple domains and protocols.

With the goal to meet these requirements, it is proposed an AA module that protects data flowing among module interfaces of the CSF. It guarantees resource availability only to authenticated subjects and allows only the specific actions they are authorised for. In this way all CSF components delegate their security mechanisms for inter-service communications to the proposed module, consequently facilitating the overall CSF implementation through a dedicated and uniform security system that purely focuses on authentication and fine-grained authorisations. This is actually a challenge in complex CSFs, characterised by

a vast variety of modules, provided by different parties, and operators that exchange heterogeneous and sensitive data.

7.1 Reference architecture

The GUARD platform can be seen as a collection of security services with the aim to analyse and protect data to provide situational awareness and counteract malicious patterns. Each service provides a specialised capability to manage the acquired data, based on particular criteria. The architecture can be generally subdivided into the following macro-blocks, illustrated in Figure 7.1:

- Security Services in the centralised platform, that focus on data analysis.
- Local Agents, external to the main core architecture, that extend GUARD capabilities by means of additional specialised security functions. Here the Context Broker, internal to the Core Platform, is vital in providing the status of the internal Security Services and managing Local Agent communications.
- A message broker, i.e. Kafka, for inter-service data exchange. Kafka is a distributed messaging system based on the publish/subscribe model that stores information in a distributed commit log [286].
- A Security Dashboard, to orchestrate and manage policies and security pipelines, i.e., the connections between a data source and a set of services.

The macro-blocks communicate via three APIs, each one with its own role:

- API #1 for collecting data, characterised by the communication of messages structured in a common data format using a message broker. Moreover, this API provides capability control by means of a representational state transfer (REST) interface exposed by Local Agents.
- API #2 for create, read, update, and delete operations on data models available on the platform by means of a common REST interface.
- API #3 for delivering notification messages. Furthermore, this API provides the configuration of security services through REST interfaces and remote procedure calls.

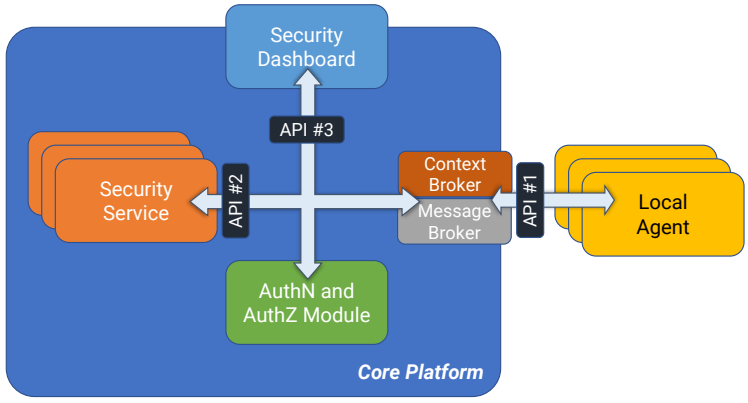


Figure 7.1: Macroscopic overview of the GUARD Architecture.

7.2 Proposed security mechanism

The goal of the proposed AA module is to provide protection to the interfaces presented in Section 7.1, while taking into account their differences in order to maximise the flexibility of access control policies to be encoded in an identity. A service identity can be used by the communicating services for sender validation and apply access policies based on attribute-based access control (ABAC) [287]. This access control method has been introduced by the National Institute of Standards and Technologies (NIST). It is based on granting or denying user requests based on both attributes of users and attributes of the object to be accessed to. Moreover, environmental conditions may be taken into account to further refine access policies. This access control model offers a high degree of flexibility in heterogeneous and complex architectures like GUARD [288]. In this scenario, it is necessary to issue identities from a centralised security platform apt to configure identities with all the security features in place.

To this end, the identity provider (IdP) authenticates services with secure identities and exposes public keys to verify them. Therefore, it is a middleware and a trusted authority between GUARD services and security operators, as illustrated in Figure 7.2. This subsystem has a logical core that manages identities for software entities, namely the "identity and attribute manager", and stores them in a persistent storage location, called the "identity database". An identity is a complex object with a dictionary of attributes and a security signature to protect

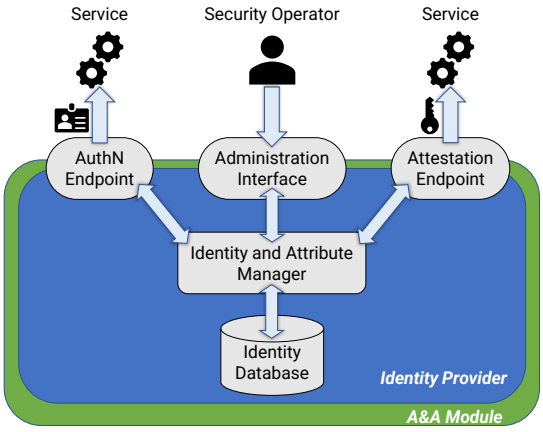


Figure 7.2: A close lookup of IdP software components with their interfaces and expected communicating actors.

it from forging, replay, and tampering attacks. It is defined and maintained by security operators through the Administration Interface. Identities can be retrieved by services through the Authentication Endpoint by means of a secret, which is generated upon service registration on the IdP. The identity is released in the form of a limited-lifespan security token that contains related attributes and a message authenticated code to preserve its integrity. Once services have been correctly identified, they can communicate with other services across the platform. On the other hand, services, that receive requests or data, validate the sender by its identity signature using the IdP public keys available from the Attestation Endpoint. These keys are used to decrypt the identity's message authenticated code and verify its digest.

Figure 7.3 proposes a three-actor communication model commonly found in cloud environments, where a service sends data to a message broker, or bus, which then delivers them to other services. Connections between services and message bus are protected by the same security mechanism to ensure that the data flow is protected, thus minimizing the risk of their leakage. The procedure is composed of the following steps:

1. Sender/Recipient retrieves its identity by authenticating with the IdP.

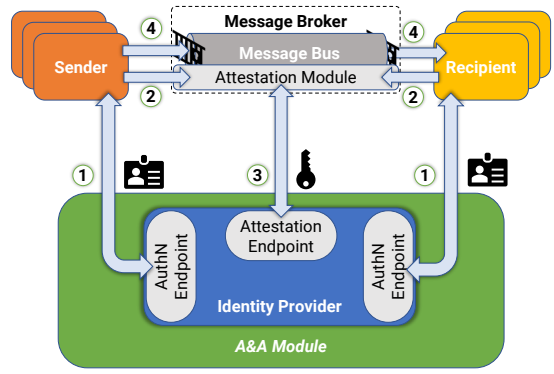


Figure 7.3: Communications guarded by the proposed security mechanism.

2. Sender and recipient connect to the message bus and send their identity during the initial connection phase (i.e., their handshake.)

3. The attestation module validates the identity and performs authorisation procedures to protect the message bus.

4. Sender/Recipient, if correctly authenticated and authorised, can send/receive messages.

For this particular communication model, the attributes, included in identities, declare the type of messages that are delivered in the message bus, like topic names. This means that senders and recipients must respect the declarations reported on the identity to ensure an authorised data exchange through the message bus. Therefore, once they are authorised, they cannot divert their behaviour and publish messages on other channels of the bus. Consequently, service interference is minimised, thus preserving the state of the message bus and their channels.

7.3 Implementation of the security mechanism

The experimental scenario has been developed as a stand-alone test bed. It has been organised in micro-services using Docker, which helps building small software units in a container-based environment [289].

The security mechanism hereby implemented focuses on the initial handshake procedure between services. Their authentication has been integrated with the proposed security mechanism with the support of the IdP. Moreover, authorisation operations have been tightly integrated to each service to provide flexibility over adopted policies for each particular operation on a specific service.

The experimental scenario is a composition of containers, enabling the evaluation of the proposal with a set of diverse service configurations:

- WSO2 identity server (IS), that acts as the IdP of reference, configured for the proposed security mechanism [290].
- An Apache Kafka broker [286] equipped with the Attestation Module to provide secure communications.
- A Java-based Kafka client that uses its identity to authenticate with Kafka and publish messages in a specific topic.
- A Java-based Kafka client that consumes messages from a determined topic.

The IdP, exposed by the AA module, is based on the open source IS project developed by WSO2 [290]. It is an extensible Identity Management platform that provides secure and standardised authentication mechanisms, identity federation, access control procedures, and APIs to automate its administration. This IdP has been identified as the ideal one to provide the basic software infrastructure for the AA module. In fact, it allows security operators to register, manage and dispose identities and attributes via a Hyper-Text Transport Protocol (HTTP)-based Administration Interface.

A service to protect must be registered on the IS as a service provider. In order to do so, a security operator has to configure the IS by providing essential information to recognise the service. service provider registration consists of a two-steps procedure:

1. Register the new service provider, also known as the application to protect. This operation consists in providing a service name and an optional description. At the end of the registration, the IS provides a unique identifier to refer to such service.
2. Register a new OAuth2 inbound authenticator associated with the service provider. It exposes the necessary software infrastructure of the IS for

authentication procedures that follow a specified protocol, OAuth2 in this case [290].

This procedure can be entirely automated by interfacing with the exposed IS APIs. Specifically, in order to register a new service provider programmatically, the procedure must send a request to the IS. It is composed of a payload, also known as Simple Object Access Protocol (SOAP) envelope, and a header that contains the HTTP authorisation field to authenticate with the IS. The SOAP envelope is an XML-formatted HTTP body containing name and description of the service provider and its OAuth2 inbound authenticator to activate.

In this test bed, Kafka has been protected in order to evaluate the authenticity of each connected service and their authorisations. This scenario has the goal of protecting the message bus in case of service misuse, minimizing the potential interference of infected services to other communications if they acquire rights to publish/subscribe to other topics.

To this end, the test bed comprises a Kafka broker and service examples that produce and consume messages. The goal of this test is to demonstrate the possibility of integrating the security mechanism with Kafka. This broker already provides high-level software interfaces for authentication via simple authentication and security layer (SASL) and OAuth2 [286], but it is limited to the validation of unsecured JSON Web Tokens (JWTs), i.e., it does not check the message authenticated code, and it does not support external IdPs like WSO2 IS. To overcome this limitation, this test bed includes a library that provides all the necessary implementations to enable security checks between Kafka and the IS by following SASL and OAuth2 specifications. It also decodes and validates JWTs through public keys available from the IdP JSON Web Key Sets (JWKS) endpoint. Furthermore, as the local authorisation mechanism, this extension mandates that each service declares a set of topics that it will use during execution through additional attributes encoded in the JWT. This is a confinement strategy to preserve application rights and limit what the service can do at runtime. Services must be compatible with this new procedure to be authorised by the broker, otherwise the broker will abruptly close the connection. The broker validates the identity and elaborates an authorisation response to send back to the sender. After that, the traditional Kafka session can begin by following its reference protocol.

7.4 Performance evaluation

The implementation of the AA module described in Section 7.3 has been realised in the GUARD platform, to test the communication functionalities and to validate the proposal. In this context, the performance of the module has been evaluated by checking the authentication procedures and access control mechanisms between GUARD services. Different performance indicators have been chosen in terms of response time (e.g., elapsed time for service AA and latency during message-based communications with Kafka) and resource usage (e.g., CPU Load and Memory Usage) during the AA processes.

The related measurements, made during the establishment of the handshake procedure between the service and Kafka to send/retrieve messages, have been carried out with and without the integration of the proposed security mechanism, to measure the time and resource overhead taken by the AA module.

Simulations have been set up in a Docker environment with 8 virtual processors, 4 GB of memory, and 4 GB of swap memory on solid state drive. It has been configured with (i) an IdP that constitutes the AA module, (ii) a Kafka broker with the AA module extension to provide AA functionalities, (iii) a number of Kafka clients, called *producers*, that publish data on a fixed reference topic, and (iv) one Kafka client, called *consumer*, that subscribes to the same reference topic of the producers.

Each simulation run lasts 10 minutes. During this time period, the producers publish messages in bursts of 100 messages each. After each burst, they are restarted and the process is repeated, until the end of the run. This generates a huge number of samples (about 1.1e5) for a robust analysis of the Key Performance Indicators evaluated in the following subsections.

7.4.1 Elapsed time for service authentication and authorisation

Figure 7.4 depicts the latency distribution for a number of concurrent producers (1, 3, and 5) that conclude successfully the authentication procedure to Kafka. Table 7.1 reports the mean (μ) and the variance (σ^2) of the latency obtained for service authentication and for a different number of producers.

As depicted in Table 7.1, both the latency mean value and the dispersion around it increase with the number of producers. This behaviour is confirmed by the curves depicted in Figure 7.4, where the latency distribution is shifted

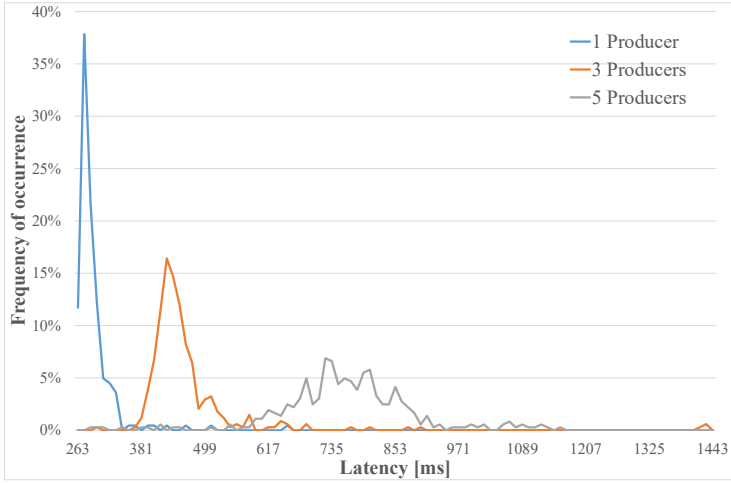


Figure 7.4: Latency distribution for service authentication to Kafka, for different numbers of producers.

towards higher values as the number of producers increases, being also sparser around the mean value.

The growth of the mean latency and the variance both derive from the complexity of the OAuth protocol used to authenticate services. The more concurrent authentication procedures are performed, the more resources are needed by the centralised IdP to accomplish those procedures simultaneously.

Regarding service authorisation, Kafka has been stressed with a number of concurrent producers publishing messages on the same reference topic, and with one consumer subscribing to that topic. The role of the AA module is to check the correct authorisation of each client during its publish/subscribe operations. Table 7.1 shows the mean and the variance results of the latency values obtained for service authorisation.

As it can be seen from Table 7.1, authorisation latency for 1 producer is very small, but it incurs in a performance penalty of 97.6% when the number of producers increases from 1 to 3, whereas the performance penalty is of 98.4% when the number of producers increases from 1 to 5. The dispersion of the

Number of concurrent producers	Authentication		Authorisation	
	μ [ms]	σ^2 [ms^2]	μ [ms]	σ^2 [ms^2]
1	296.75	1541.11	0.146	0.2227
3	471.23	12547.07	6.013	11211.5
5	768.88	15539	9.162	20472

Table 7.1: Mean and variance of AA latency for different number of producers.

measured latency around the mean highly increases for increasing numbers of producers. This is due to the producers concurrently requesting authorisation to publish new data.

7.4.2 Latency overhead in message reception

To evaluate the latency overhead due to the AA module, latency results have been compared with those obtained without the integration of the module. In the latter case, the only Transport Level Security (TLS) protocol has been implemented for a secure communication with Kafka. To derive the frequency of occurrence of latency overhead, the range of the experimented latency samples has been subdivided into 2000 bins, of size 1.2 ms each. In Figure 7.5, only the first 12 bins, up to 14.5 ms of latency, have been plotted. In fact, even if also higher latency values have been observed, their number is negligible and their sparsity is high. Therefore, they have not been represented in the figure for the sake of better clarity.

Given the same number of producers, the performance behaviour with and without the AA module are very similar to each other. Additionally, the latency distribution, in case of the adoption of the AA module, is sparser in the range depicted in Figure 7.5.

To provide a more complete analysis, Table 7.2 shows mean and variance of latency values for all the numbers of producers, considering all the results collected in the experiments.

If the AA module is adopted, the average value of the latency keeps around 5 ms, almost independently of the number of producers. Conversely, it progressively decreases for an increasing number of producers if the module is absent. This behaviour is mainly due to cache optimisation of IS and Kafka module, which penalises the use of a small number of authentication procedures (a more

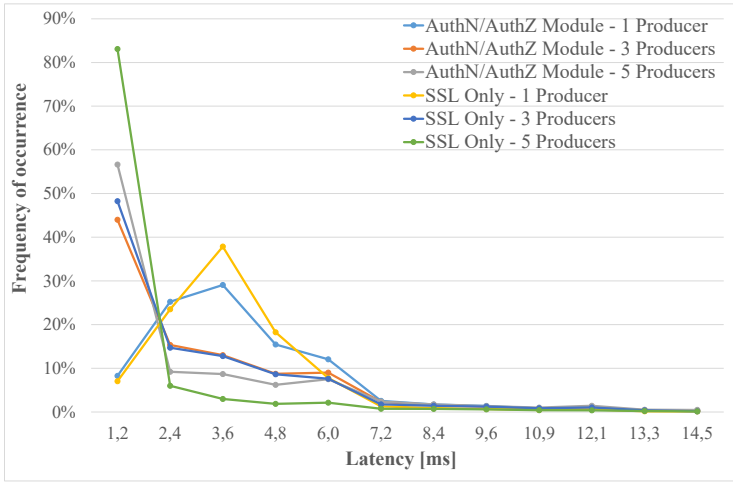


Figure 7.5: Latency distribution with and without the AA module, for different numbers of producers.

Number of concurrent producers	With AA module		Without AA module	
	μ [ms]	σ^2 [ms^2]	μ [ms]	σ^2 [ms^2]
1	5.286	544.744	11.727	6769.822
3	5.811	1875.025	5.738	2238.178
5	5.332	1550.494	2.476	2132.536

Table 7.2: Mean and variance of latency results with and without the AA module, for different number of producers.

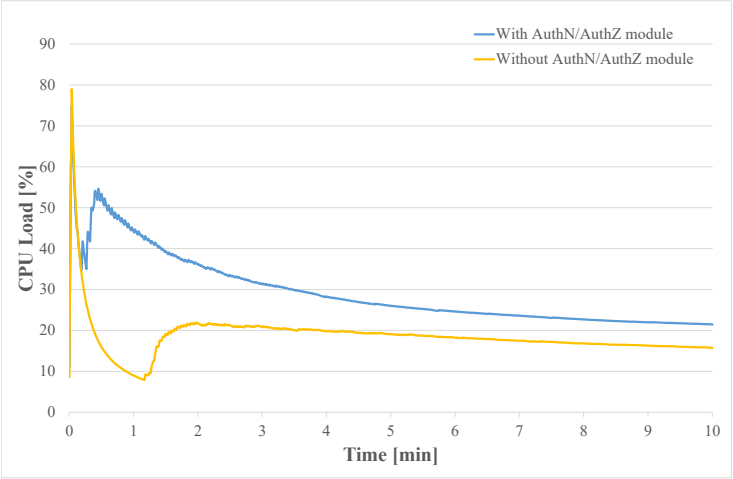


Figure 7.6: Percentage of CPU load used by Kafka with and without the AA module integration.

detailed and accurate steady-state analysis can be done in the future to evaluate the effective scalability of the mechanism.) By comparing the variance results, it can be argued that if the AA module is present, the variability around the mean latency value is smaller, if compared to the case of the only TLS adoption. This behaviour is mainly due to the AA procedures between the module and Kafka.

7.4.3 Resource consumption

To test the impact of the module implementation on the utilised resources, Kafka has been tested with and without the integration of the AA module, evaluating the temporal evolution of the CPU and memory load for 10 minutes of simulation. Figure 7.6 and 7.7 depict the related results.

From Figure 7.6, it can be noticed that after a start-up phase, which includes the procedure to authenticate Kafka as a trusted service provider in GUARD, the evolution of the CPU load is similar in the two scenarios with (blue line) and without (yellow line) the AA module. On average, Kafka with the AA module

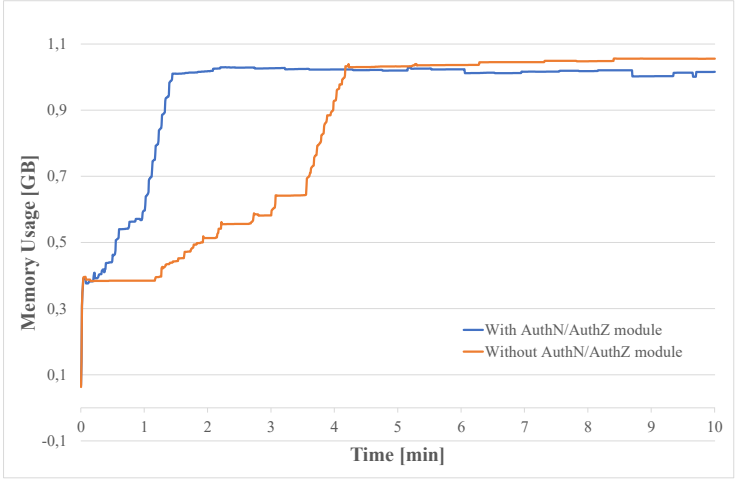


Figure 7.7: Memory usage of Kafka with and without the AA module integration.

integration requires a CPU Load of 29%, whereas the base version of it requires a CPU Load of 19%. So, the overhead in CPU Load introduced by the AA module is of about 36%.

Figure 7.7 shows that Kafka integrating the AA module has an average memory usage of 956 MB, whereas 834 MB are required, on average, without the AA module integration. To this end, the introduced overhead is of 13%.

7.5 Lesson learnt

An ABAC-based security mechanism has been presented to protect inter-service communications in the context of GUARD cyber-security framework. It has been implemented as a module that provides authentication procedures and local authorisation policies at each service by relying on a trusted identity provider managed by security operators. The proposed module has been successfully implemented in the GUARD environment by using a set of services and a Kafka broker for publish/subscribe messaging. Experimental results show that latency

performances of the proposed module are very close to those obtained without its integration, even if this comes at a cost of an increased resource consumption in terms of CPU and memory. Nevertheless, the proposed module brings a much higher degree of flexibility in AA procedures, thanks to the ABAC paradigm; this is a great value added in a complex platform with many heterogeneous modules and stakeholders.

Conclusion

NTNs have the transformative potential in achieving pervasive connectivity envisioned for 6G, addressing societal challenges such as climate change, sustainable development, and urbanisation. Through the integration of UAVs, HAPs, and satellites, these networks hold the promise of revolutionizing fields like emergency response, environmental monitoring, smart agriculture, and civil engineering. Such technology is increasingly central to creating large-scale missions that span urban and rural areas, highlighting the dual-use potential of NTNs for both civilian and defence applications.

Recognizing the challenges of large-scale UAV deployment, the role of deverticalizing middlewares and simulators is vital for rapid prototyping and testing, significantly reducing costs and accelerating development. While such middlewares are crucial to enable deployment of drones at scale, simulators are invaluable in preparing solutions for real-world deployment, offering a platform for professionals, researchers, and students to experiment with complex scenarios and gain critical skills, insights, and experience. To this end, a modular simulation tool has been developed to model various UAV parameters and integrate seamlessly with evolving technological requirements, supporting trajectory design, network optimisation, and data processing. As shown by the simulation campaign's results, it represents an invaluable companion in demonstrating the feasibility of envisioned use cases and scenarios.

Attention must be also put on ensuring resilience and security of these networks, given their rise as cyber-physical threats. Specifically, UAVs bring considerable efficiency in communications, but also expose society to risks from potential misuse by malicious actors. To this end, C-UAS solutions aim to safeguard these technologies, including a taxonomy of sensors for UAV DTI. Further research is encouraged in multi-sensor data fusion to enhance robustness and accuracy of C-UAS implementations, with detailed evaluations of current sensor algorithms to identify technical gaps.

Security within 6G framework also requires robust inter-service AA solutions. An ABAC model is proposed to secure communications across digital service

chains that incorporate UAVs, ensuring seamless, autonomous authorisation without compromising latency or performance. This approach supports modular, interoperable service architectures, allowing UAVs and associated systems to interact safely within multi-tenant networks.

Concluding, the scientific research is crucial in guiding further developments of NTNs and UAV networks to shape the future of connectivity. By continuing to address technical and security challenges through innovation, NTNs, especially drones, have the potential to transform society with the creation of an open, resilient, pervasive, and ubiquitous mobile communication infrastructure.

Acknowledgement

Quasi dieci anni fa decisi di intraprendere questo cammino. “Voglio diventare un ingegnere”, dissi tra me e me, nel silenzio di casa, fissando le mie pupille allo specchio. All’epoca, preferivo annunciare certe cose senza troppi testimoni. “Un ottimo ingegnere”, aggiunsi. Tanto valeva: perché dedicare tanti anni senza l’ambizione di scalare verso l’illusoria perfezione? Quanto più è doveroso battersi con la propria vita contro i problemi che affliggono l’umanità, tanto più è necessario impegnarsi per emergere dal mare della mediocrità.

Per me è quasi diventato un piacere alzare continuamente l’asticella, affrontando sfide sempre più complesse, nel tentativo di saziare la mia instancabile e bulimica curiosità. A tal proposito, devo ringraziare me stesso per non aver dato ascolto a Piero, mio Maestro e Amico, perché questi tre anni di dottorato mi hanno intrattenuto – e arricchito – come nient’altro avrebbe potuto.

A papà, mamma e mia sorella: vi ringrazio per la vostra infinita pazienza e per avermi tollerato, specialmente nei momenti in cui mi sono mostrato più fragile. A Sara: grazie per avermi accompagnato lungo tutto questo cammino. È stato uno splendido sentiero percorso insieme, ma ora siamo giunti al fatidico bivio. Grazie per questa avventura condivisa. Un pensiero speciale va a Roberto e Sara, Irene e Bruno, così come all’intero MqttTeam, al Telematics Lab e al PolibaX, per avermi ricordato, con costanza, che oltre allo studio e al duro lavoro esistono un mondo ricco di meraviglie e una sincera amicizia da coltivare. Ringrazio inoltre il mio supervisore e l’intero cuore pulsante del laboratorio per la stima e la fiducia che non mi hanno mai fatto mancare.

Se potessi incontrare il Giovanni del duemilaquindici, lo abbraccerei forte. Devo questa splendida avventura alla sua intuizione e alla sua ingenuità. La strada, però, non è ancora finita: un ingegnere, per essere davvero ottimo, non deve limitarsi al duro lavoro.

Infine, a chi sta leggendo queste righe: vi voglio un mondo di bene. Chissà, un giorno i miei nonni potranno vedere ciò che avrò plasmato e cambiato. Nel frattempo, dedico a loro questa dissertazione: una piccola goccia nel vasto mare di virtù e conoscenza.

Bibliography

[1] Giovanni Grieco, Giovanni Iacovelli, Pietro Boccadoro and Luigi Alfredo Grieco. **Internet of Drones Simulator: Design, Implementation, and Performance Evaluation**. *IEEE Internet of Things Journal* 10:2 (2023), 1476–1498. DOI: 10.1109/JIOT.2022.3207324 (see pages xxvii, 33).

[2] Giovanni Grieco, Giovanni Iacovelli, Daniele Pugliese, Domenico Striccoli and Luigi Alfredo Grieco. **A System-Level Simulation Module for Multi-UAV IRS-Assisted Communications**. *IEEE Transactions on Vehicular Technology* 73:5 (2024), 6740–6751. DOI: 10.1109/TVT.2023.3342298 (see pages xxvii, 33).

[3] Giovanni Iacovelli, Giovanni Grieco, Antonio Petrosino, Luigi Alfredo Grieco and Gennaro Boggia. **Fair Energy and Data Rate Maximization in UAV-Powered IoT-Satellite Integrated Networks**. *IEEE Transactions on Communications* 72:4 (2024), 2457–2469. DOI: 10.1109/TCOMM.2023.3343417 (see pages xxvii, 3).

[4] Giovanni Grieco, Domenico Striccoli, Giuseppe Piro, Raffaele Bolla, Gennaro Boggia and Luigi Alfredo Grieco. **Authentication and Authorization in Cyber-Security Frameworks: a Novel Approach for Securing Digital Service Chains**. In: *2022 IEEE 8th International Conference on Network Softwarization (NetSoft)*. 2022, 468–473. DOI: 10.1109/NetSoft54395.2022.9844030 (see page xxvii).

[5] Giovanni Grieco, Giovanni Iacovelli, Pietro Boccadoro and Luigi Alfredo Grieco. **On the Design of the Drone Control Layer**. In: *European Wireless 2021; 26th European Wireless Conference*. 2021, 1–7 (see page xxviii).

[6] Giovanni Grieco, Giovanni Iacovelli, Mattia Sandri, Marco Giordani, Michele Zorzi and Luigi Alfredo Grieco. **Preliminary Performance Evaluation of a Satellite-to-HAP Communication Link**. In: *European Wireless 2023; 28th European Wireless Conference*. 2023, 340–345 (see pages xxviii, 33).

[7] Alessandro Carrega, Giovanni Grieco, Domenico Striccoli, Manos Papoutsakis, Tomas Lima, José Ignacio Carretero and Matteo Repetto, 1–31. In: *Cybersecurity of Digital Service Chains: Challenges, Methodologies, and Tools*. Ed. by Joanna Kołodziej, Matteo Repetto and Armend Duzha. Cham: Springer International Publishing, 2022. ISBN: 978-3-031-04036-8. DOI: 10.1007/978-3-031-04036-8_1 (see pages xxviii, 175).

[8] Giovanni Grieco, Danilo Amendola and David Anderson. **Counter-drone systems and data fusion**. Tech. rep. KJ-01-24-196-EN-N. ISSN: 1831-9424. ISBN: 978-92-68-22705-3. Luxembourg (Luxembourg): Publications Office of the European Union, 2024. DOI: 10.2760/6037951 (see page xxviii).

[9] Giovanni Grieco, Arcangela Rago, Gennaro Boggia and Luigi Alfredo Grieco. **Cyber-Physical Spatial Awareness for Safe UAVs in XAI-enhanced 6G Networks: Vision, Challenges, and Opportunities** (2024). Submitted to *IEEE Network Magazine*, pending peer review (see page xxviii).

[10] Giovanni Grieco, Daniele Pugliese, Adnan Rashid and Ilaria Cianci. **IoD-Sim: an Open-Source Simulator for 6G Integrated Terrestrial/Non-Terrestrial Networks** (2024). Submitted to *Elsevier Computer Networks – Software Article*, pending peer review (see page xxviii).

[11] Sastri Kota, Giovanni Giambene, Mohammed Abdelsadek, Mohamed-Slim Alouini, Sarath Babu, Joan Bas, Sachin Chaudhari, Debabrata Dalai, Tasneem Darwish, Tomaso de Cola, Thomas Delamotte, Ashutosh Dutta, Ayush Dwivedi, Michael Enright, Marco Giordani, Alberto Gotta, Eman Hammad, Tamer Khattab, Andreas Knopp, Gunes Karabulut Kurt, B. S. Manoj, Jean-Daniel Medjo Me Biomo, Prashant Pillai, Pramud Rawat, Paresh Saxena, Pat Scanlan, Avinash Sharma, Ray Sperber, Zhili Sun, Daniele Tarchi, Neeraj Varshney, Seema Verma, Halim Yanikomeroglu, Kanglian Zhao and Liang Zhao. **INGR Roadmap Satellite Chapter**. In: *2022 IEEE Future Networks World Forum (FNWF)*. 2022, 1–182. DOI: 10.1109/FNWF55208.2022.00141 (see page 8).

[12] Zhengquan Zhang, Yue Xiao, Zheng Ma, Ming Xiao, Zhiguo Ding, Xianfu Lei, George K. Karagiannidis and Pingzhi Fan. **6G Wireless Networks: Vision, Requirements, Architecture, and Key Technologies**. *IEEE Vehicular Technology Magazine* 14:3 (2019), 28–41. DOI: 10.1109/MVT.2019.2921208 (see pages 7, 149).

[13] Marco Giordani, Michele Polese, Marco Mezzavilla, Sundeep Rangan and Michele Zorzi. **Toward 6G Networks: Use Cases and Technologies**. *IEEE Communications Magazine* 58:3 (2020), 55–61. DOI: 10.1109/MCOM.001.1900411 (see page 7).

[14] Muhammad Awais Javed, Tu N. Nguyen, Jawad Mirza, Junaid Ahmed and Bakhtiar Ali. **Reliable Communications for Cybertwin-Driven 6G IoVs Using Intelligent Reflecting Surfaces**. *IEEE Transactions on Industrial Informatics* 18:11 (2022), 7454–7462. DOI: 10.1109/TII.2022.3151773 (see page 7).

[15] Baofeng Ji, Yanan Wang, Ling Xing, Chunguo Li, Yi Wang, Hong Wen and Kaizhi Huang. **IRS-driven Cybersecurity of Healthcare Cyber Physical Systems**. *IEEE Transactions on Network Science and Engineering* (2022), 1–10. DOI: 10.1109/TNSE.2022.3213082 (see page 7).

[16] **IMT Vision–Framework and overall objectives of the future development of IMT for 2020 and beyond**. *Recommendation ITU* 2083:0 (2015) (see page 7).

[17] Harsh Tataria, Mansoor Shafi, Mischa Dohler and Shu Sun. **Six Critical Challenges for 6G Wireless Systems: A Summary and Some Solutions**. *IEEE Vehicular Technology Magazine* 17:1 (2022), 16–26. DOI: 10.1109/MVT.2021.3136506 (see page 7).

[18] Huanxi Cui, Jun Zhang, Yuhui Geng, Zhenyu Xiao, Tao Sun, Ning Zhang, Jiajia Liu, Qihui Wu and Xianbin Cao. **Space-air-ground integrated network (SAGIN) for 6G: Requirements, architecture and challenges**. *China Communications* 19:2 (2022), 90–108. DOI: 10.23919/JCC.2022.02.008 (see page 7).

[19] Mamta Agiwal, Abhishek Roy and Navrati Saxena. **Next Generation 5G Wireless Networks: A Comprehensive Survey**. *IEEE Communications Surveys & Tutorials* 18:3 (2016), 1617–1655. DOI: 10.1109/COMST.2016.2532458 (see page 7).

[20] Marco Giordani and Michele Zorzi. **Non-Terrestrial Networks in the 6G Era: Challenges and Opportunities**. *IEEE Network* 35:2 (2021), 244–251. DOI: 10.1109/MNET.011.2000493 (see page 7).

[21] Mohammad Mozaffari, Xingqin Lin and Stephen Hayes. **Toward 6G with Connected Sky: UAVs and Beyond**. *IEEE Communications Magazine* 59:12 (2021), 74–80 (see page 8).

[22] M. Gharibi, R. Boutaba and S. L. Waslander. **Internet of Drones**. *IEEE Access* 4 (2016), 1148–1162. DOI: 10.1109/ACCESS.2016.2537208 (see pages 8, 17, 41, 51, 156).

[23] Pietro Boccadoro, Domenico Striccoli and Luigi Alfredo Grieco. **An extensive survey on the Internet of Drones**. *Ad Hoc Networks* 122 (2021), 102600. ISSN: 1570-8705. DOI: https://doi.org/10.1016/j.adhoc.2021.102600. URL: https://www.sciencedirect.com/science/article/pii/S1570870521001335 (see pages 8, 9, 11, 17, 162).

[24] H. Shakhatreh, A. H. Sawalmeh, A. Al-Fuqaha, Z. Dou, E. Almaita, I. Khalil, N. S. Othman, A. Khreishah and M. Guizani. **Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges**. *IEEE Access* 7 (2019), 48572–48634. DOI: 10.1109/ACCESS.2019.2909530 (see page 8).

[25] Mohammad Aminul Hoque, Mahmud Hossain, Shahid Noor, S. M. Riazul Islam and Ragib Hasan. **IoTaaS: Drone-Based Internet of Things as a Service Framework for Smart Cities**. *IEEE Internet of Things Journal* 9:14 (2022), 12425–12439. DOI: 10.1109/JIOT.2021.3137362 (see page 9).

[26] Seppo Horsmanheimo, Lotta Tuomimäki, Vasilii Semkin, Stephan Mehnert, Tao Chen, Marko Ojennus and Lasse Nykänen. **5G Communication QoS Measurements for Smart City UAV Services**. In: *2022 16th European Conference on Antennas and Propagation (EuCAP)*. 2022, 1–5. DOI: 10.23919/EuCAP53622.2022.9769210 (see page 9).

[27] Saeed H Alsamhi, Ou Ma, M Samar Ansari and Sachin Kumar Gupta. **Collaboration of drone and internet of public safety things in smart cities: An overview of qos and network performance optimization**. *Drones* 3:1 (2019), 13 (see page 9).

[28] Nader S. Labib, Matthias R. Brust, Grégoire Danoy and Pascal Bouvry. **The Rise of Drones in Internet of Things: A Survey on the Evolution, Prospects and Challenges of Unmanned Aerial Vehicles**. *IEEE Access* 9 (2021), 115466–115487. DOI: 10.1109/ACCESS.2021.3104963 (see page 9).

[29] M. Mahdi Azari, Sourabh Solanki, Symeon Chatzinotas, Oltjon Kodheli, Hazem Sallouha, Achiel Colpaert, Jesus Fabian Mendoza Montoya, Sofie Pollin, Alireza Haqiqatnejad, Arsham Mostaani, Eva Lagunas and Bjorn Ottersten. **Evolution of Non-Terrestrial Networks From 5G to 6G: A Survey**. *IEEE Communications Surveys & Tutorials* 24:4 (2022), 2633–2672. DOI: 10.1109/COMST.2022.3199901 (see pages 9, 11).

[30] Saeed H. Alsamhi, Ou Ma, Mohammad Samar Ansari and Faris A. Almalki. **Survey on Collaborative Smart Drones and Internet of Things for Improving Smartness of Smart Cities**. *IEEE Access* 7 (2019), 128125–128152. DOI: 10.1109/ACCESS.2019.2934998 (see page 9).

[31] Saeed H Alsamhi, Faris A Almalki, Hatem Al-Dois, Soufiene Ben Othman, Jahan Hassan, Ammar Hawbani, Radyah Sahal, Brian Lee and Hager Saleh. **Machine learning for smart environments in B5G networks: connectivity and QoS**. *Computational Intelligence and Neuroscience* 2021 (2021) (see page 9).

[32] M. Mozaffari, W. Saad, M. Bennis, Y. Nam and M. Debbah. **A Tutorial on UAVs for Wireless Networks: Applications, Challenges, and Open Problems**. *IEEE Commun. Surveys Tuts* 21:3 (2019), 2334–2360. DOI: 10.1109/COMST.2019.2902862 (see page 9).

[33] Rojeena Bajracharya, Rakesh Shrestha, Shiho Kim and Haejoon Jung. **6G NR-U Based Wireless Infrastructure UAV: Standardization, Opportunities, Challenges and Future Scopes**. *IEEE Access* 10 (2022), 30536–30555. DOI: 10.1109/ACCESS.2022.3159698 (see page 9).

[34] Siyi Liao, Jun Wu, Jianhua Li, Ali Kashif Bashir and Wu Yang. **Securing collaborative environment monitoring in smart cities using blockchain enabled software-defined internet of drones**. *IEEE Internet of Things Magazine* 4:1 (2021), 12–18 (see page 9).

[35] Qiang Liu, Songlin Sun, Bo Rong and Michel Kadoch. **Intelligent Reflective Surface Based 6G Communications for Sustainable Energy Infrastructure**. *IEEE Wireless Communications* 28:6 (2021), 49–55. DOI: 10.1109/MWC.016.2100179 (see page 9).

[36] Giovanni Geraci, Adrian Garcia-Rodriguez, M. Mahdi Azari, Angel Lozano, Marco Mezzavilla, Symeon Chatzinotas, Yun Chen, Sundeep Rangan and Marco Di Renzo. **What Will the Future of UAV Cellular Communications Be? A Flight From 5G to 6G**. *IEEE Communications Surveys & Tutorials* 24:3 (2022), 1304–1335. DOI: 10.1109/COMST.2022.3171135 (see page 9).

[37] **IEEE Standard for Local and Metropolitan Area Networks–Part 15.7: Short-Range Wireless Optical Communication Using Visible Light**. *IEEE Std 802.15.7-2011* (2011), 1–309. DOI: 10.1109/IEEESTD.2011.6016195 (see page 9).

[38] **IEEE Draft Standard for Wireless Multi-Media Networks**. *IEEE P802.15.3-Rev.B/D4.0, January 2023* (2023), 1–685 (see page 9).

[39] Shimin Gong, Xiao Lu, Dinh Thai Hoang, Dusit Niyato, Lei Shu, Dong In Kim and Ying-Chang Liang. **Toward Smart Wireless Communications via Intelligent Reflecting Surfaces: A Contemporary Survey**. *IEEE Communications Surveys & Tutorials* 22:4 (2020), 2283–2314. DOI: 10.1109/COMST.2020.3004197 (see page 9).

[40] Giovanni Iacovelli, Angelo Coluccia and Luigi Alfredo Grieco. **Channel Gain Lower Bound for IRS-Assisted UAV-Aided Communications**. *IEEE Communications Letters* 25:12 (2021), 3805–3809 (see page 10).

[41] Ge, Linghui and Zhang, Hua and Wang, Jun-Bo and Li, Geoffrey Ye. **Reconfigurable Wireless Relaying With Multi-UAV-Carried Intelligent Reflecting Surfaces**. *IEEE Transactions on Vehicular Technology* 72:4 (2023), 4932–4947. DOI: 10.1109/TVT.2022.3227623 (see page 10).

[42] Lyu, Wanting and Xiu, Yue and Yang, Songjie and Yeoh, Phee Lep and Li, Yonghui and Zhang, Zhongpei. **Weighted Sum Age of Information Minimization in Wireless Networks With Aerial IRS**. *IEEE Transactions on Vehicular Technology* 72:4 (2023), 5390–5394. DOI: 10.1109/TVT.2022.3223691 (see page 10).

[43] Yuwen Qian, Chenguang Yang, Zhen Mei, Xiangwei Zhou, Long Shi and Jun Li. **On Joint Optimization of Trajectory and Phase Shift for IRS-UAV Assisted Covert Communication Systems**. *IEEE Transactions on Vehicular Technology* (2023), 1–11. DOI: 10.1109/TVT.2023.3271461 (see page 10).

[44] C. De Alwis, Q.V. Pham and M. Liyanage. **6G Frontiers: Towards Future Wireless Systems**. Wiley, 2022. ISBN: 9781119862338. URL: https://books.google.it/books?id=7LKeEAAAQBAJ (see page 10).

[45] Adrián García-Gutiérrez, Jesús Gonzalo, Diego Domínguez, Deibi López and Alberto Escapa. **Aerodynamic optimization of propellers for High Altitude Pseudo-Satellites**. *Aerospace Science and Technology* 96 (Jan. 2020), 105562 (see page 10).

[46] Yekta Turk and Engin Zeydan. **Satellite Backhauling for Next Generation Cellular Networks: Challenges and Opportunities**. *IEEE Communications Magazine* 57:12 (Dec. 2019), 52–57. DOI: 10.1109/MCOM.001.1900193 (see page 10).

[47] Michele Luglio, Mattia Quadrini, Cesare Roseti and Francesco Zampognaro. **Modes and Models for Satellite Integration in 5G Networks**. *IEEE Communications Magazine* 61:4 (Apr. 2023), 50–56. DOI: 10.1109/MCOM.002.2200209 (see page 10).

[48] Abdelaali Chaoub, Marco Giordani, Brejesh Lall, Vimal Bhatia, Adrian Kliks, Luciano Mendes, Khaled Rabie, Harri Saarnisaari, Amit Singhal, Nan Zhang et al. **6G for bridging the digital divide: Wireless connectivity to remote areas**. *IEEE Wireless Communications* 29:1 (July 2021), 160–168 (see page 10).

[49] Dengke Wang, Marco Giordani, Mohamed-Slim Alouini and Michele Zorzi. **The potential of multilayered hierarchical nonterrestrial networks for 6G: A comparative analysis among networking architectures**. *IEEE Vehicular Technology Magazine* 16:3 (Sept. 2021), 99–107 (see page 11).

[50] Jiajia Liu, Yongpeng Shi, Zubair Md. Fadlullah and Nei Kato. **Space-Air-Ground Integrated Network: A Survey**. *IEEE Communications Surveys & Tutorials* 20:4 (2018), 2714–2741. DOI: 10.1109/COMST.2018.2841996 (see page 11).

[51] 3GPP. **Study on New Radio (NR) to support non-terrestrial networks (NTNs)**. Technical Report (TR) 38.811. Version 15.2.0, Release 15. 3GPP, 2020 (see pages 11, 34, 41, 56, 58, 59, 117, 120, 127, 128).

[52] 3GPP. **Solutions for NR to support Non-Terrestrial Networks (NTN)**. Technical Report (TR) 38.821. Release 16. 3GPP, 2023 (see pages 11, 121).

[53] 3GPP. **Non-terrestrial networks (NTN) related RF and co-existence aspects**. Technical Report (TR) 38.863. Release 17. 3GPP, 2022 (see page 11).

[54] 3GPP. **Study on requirements and use cases for network verified UE location for Non-Terrestrial-Networks (NTN) in NR**. Technical Report (TR) 38.882. Release 18. 3GPP, 2022 (see page 11).

[55] Marco Giordani and Michele Zorzi. **Non-Terrestrial Networks in the 6G Era: Challenges and Opportunities**. *IEEE Network* 35:2 (2021), 244–251. DOI: 10.1109/MNET.011.2000493 (see page 11).

[56] Oltjon Kodheli, Eva Lagunas, Nicola Maturo, Shree Krishna Sharma, Bhavani Shankar, Jesus Fabian Mendoza Montoya, Juan Carlos Merlano Duncan, Danilo Spano, Symeon Chatzinotas, Steven Kisseleff, Jorge Querol, Lei Lei, Thang X. Vu and George Goussetis. **Satellite Communications in the New Space Era: A Survey and Future Challenges**. *IEEE Communications Surveys & Tutorials* 23:1 (2021), 70–109. DOI: 10.1109/COMST.2020.3028247 (see page 11).

[57] Zhenyu Xiao, Junyi Yang, Tianqi Mao, Chong Xu, Rui Zhang, Zhu Han and Xiang-Gen Xia. **LEO Satellite Access Network (LEO-SAN) towards 6G: Challenges and Approaches**. *IEEE Wireless Communications* (2022), 1–8. DOI: 10.1109/MWC.011.2200310 (see page 11).

[58] Dinh C. Nguyen, Ming Ding, Pubudu N. Pathirana, Aruna Seneviratne, Jun Li, Dusit Niyato, Octavia Dobre and H. Vincent Poor. **6G Internet of Things: A Comprehensive Survey**. *IEEE Internet of Things Journal* 9:1 (2022), 359–383. DOI: 10.1109/JIOT.2021.3103320 (see page 11).

[59] Xiao Lu, Ping Wang, Dusit Niyato, Dong In Kim and Zhu Han. **Wireless Charging Technologies: Fundamentals, Standards, and Network Applications**. *IEEE Communications Surveys & Tutorials* 18:2 (2016), 1413–1452. DOI: 10.1109/COMST.2015.2499783 (see page 12).

[60] Shuai Zhang, Weiqi Liu and Nirwan Ansari. **Joint Wireless Charging and Data Collection for UAV-Enabled Internet of Things Network**. *IEEE Internet of Things Journal* 9:23 (2022), 23852–23859 (see pages 12, 13).

[61] Jun Mu and Zhaojie Sun. **Trajectory Design for Multi-UAV-Aided Wireless Power Transfer toward Future Wireless Systems**. *Sensors* 22:18 (2022), 6859 (see pages 12, 13).

[62] Yuan Liu, Ke Xiong, Yang Lu, Qiang Ni, Pingyi Fan and Khaled Ben Letaief. **UAV-aided wireless power transfer and data collection in Rician fading**. *IEEE Journal on Selected Areas in Communications* 39:10 (2021), 3097–3113 (see pages 12–14).

[63] Yulin Hu, Xiaopeng Yuan, Guohua Zhang and Anke Schmeink. **Sustainable wireless sensor networks with UAV-enabled wireless power transfer**. *IEEE Transactions on Vehicular Technology* 70:8 (2021), 8050–8064 (see pages 12–14).

[64] Xiaopeng Mo, Yuwei Huang and Jie Xu. **Radio-map-based robust positioning optimization for UAV-enabled wireless power transfer**. *IEEE Wireless Communications Letters* 9:2 (2019), 179–183 (see pages 12, 13).

[65] Khoi Khac Nguyen, Antonino Masaracchia, Vishal Sharma, H Vincent Poor and Trung Q Duong. **RIS-assisted UAV communications for IoT with wireless power transfer using deep reinforcement learning**. *IEEE Journal of Selected Topics in Signal Processing* (2022) (see pages 12, 13).

[66] Xiaopeng Yuan, Hao Jiang, Yulin Hu and Anke Schmeink. **Joint Analog Beamforming and Trajectory Planning for Energy-Efficient UAV-Enabled Nonlinear Wireless Power Transfer**. *IEEE Journal on Selected Areas in Communications* 40:10 (2022), 2914–2929 (see pages 12–14).

[67] Xiaopeng Yuan, Yulin Hu and Anke Schmeink. **Joint design of UAV trajectory and directional antenna orientation in UAV-enabled wireless power transfer networks**. *IEEE Journal on Selected Areas in Communications* 39:10 (2021), 3081–3096 (see pages 12–14).

[68] Zhixin Liu, Songhan Zhao, Qingqing Wu, Yi Yang and Xinping Guan. **Joint Trajectory Design and Resource Allocation for IRS-Assisted UAV Communications With Wireless Energy Harvesting**. *IEEE Communications Letters* 26:2 (2022), 404–408. DOI: 10.1109/LCOMM.2021.3128545 (see page 12).

[69] Huimin Hu, Ke Xiong, Gang Qu, Qiang Ni, Pingyi Fan and Khaled Ben Letaief. **AoI-Minimal Trajectory Planning and Data Collection in UAV-Assisted Wireless Powered IoT Networks**. *IEEE Internet of Things Journal* 8:2 (2021), 1211–1223. DOI: 10.1109/JIOT.2020.3012835 (see pages 12, 13).

[70] Giancarlo Sciddurlo, Antonio Petrosino, Mattia Quadrini, Cesare Roseti, Domenico Striccoli, Francesco Zampognaro, Michele Luglio, Stefano Perticaroli, Antonio Mosca, Francesco Lombardi, Ivan Micheli, Antonio Ornatelli, Vincenzo Schena, Alessandro Di Mezza, Alessio Mattioni, Daniele Morbidelli, Gennaro Boggia and Giuseppe Piro. **Looking at NB-IoT Over LEO Satellite Systems: Design and Evaluation of a Service-Oriented Solution**. *IEEE Internet of Things Journal* 9:16 (2022), 14952–14964. DOI: 10.1109/JIOT.2021.3135060 (see pages 12, 144).

[71] Hugo Chelle, Michael Crosnier, Riadh Dhaou and Andre-Luc Beylot. **Adaptive Load Control for IoT Based on Satellite Communications**. In: *2018 IEEE International Conference on Communications (ICC)*. 2018, 1–7. DOI: 10.1109/ICC.2018.8422804 (see page 12).

[72] Zhixiang Gao, Aijun Liu, Chen Han and Xiaohu Liang. **Non-Orthogonal Multiple Access-Based Average Age of Information Minimization in LEO Satellite-Terrestrial Integrated Networks**. *IEEE Transactions on Green Communications and Networking* 6:3 (2022), 1793–1805. DOI: 10.1109/TGCN.2022.3159559 (see page 12).

[73] Cheng Zhan, Han Hu, Xiufeng Sui, Zhi Liu, Jianan Wang and Honggang Wang. **Joint Resource Allocation and 3D Aerial Trajectory Design for Video Streaming in UAV Communication Systems**. *IEEE Transactions on Circuits and Systems for Video Technology* 31:8 (2021), 3227–3241. DOI: 10.1109/TCSVT.2020.3035618 (see page 12).

[74] Sedighe Mirbolouk, Morteza Valizadeh, Mehdi Chehel Amirani and Samad Ali. **Relay Selection and Power Allocation for Energy Efficiency Maximization in Hybrid Satellite-UAV Networks With CoMP-NOMA Transmission**. *IEEE Transactions on Vehicular Technology* 71:5 (2022), 5087–5100. DOI: 10.1109/TVT.2022.3152048 (see page 13).

[75] Ying Wang, Zhendong Li, Yuanbin Chen, Man Liu, Xinpeng Lyu, Xiangwang Hou and Jingjing Wang. **Joint Resource Allocation and UAV Trajectory Optimization for Space–Air–Ground Internet of Remote Things Networks**. *IEEE Systems Journal* 15:4 (2021), 4745–4755. DOI: 10.1109/JSYST.2020.3019463 (see page 13).

[76] Ningyuan Wang, Feng Li, Dong Chen, Liang Liu and Zeyu Bao. **NOMA-Based Energy-Efficiency Optimization for UAV Enabled Space-Air-Ground Integrated Relay Networks**. *IEEE Transactions on Vehicular Technology* 71:4 (2022), 4129–4141. DOI: 10.1109/TVT.2022.3151369 (see page 13).

[77] Christopher R. Valenta and Gregory D. Durgin. **Harvesting Wireless Power: Survey of Energy-Harvester Conversion Efficiency in Far-Field, Wireless Power Transfer Systems**. *IEEE Microwave Magazine* 15:4 (2014), 108–120. DOI: 10.1109/MMM.2014.2309499 (see page 13).

[78] Takeru Terauchi, Katsuya Suto and Masashi Wakaiki. **Harvest-Then-Transmit-Based TDMA Protocol with Statistical Channel State Information for Wireless Powered Sensor Networks**. In: *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*. 2021, 1–5. DOI: 10.1109/VTC2021-Spring51267.2021.9449083 (see page 13).

[79] Cunhua Pan, Hong Ren, Kezhi Wang, Jonas Florentin Kolb, Maged Elkashlan, Ming Chen, Marco Di Renzo, Yang Hao, Jiangzhou Wang, A. Lee Swindlehurst, Xiaohu You and Lajos Hanzo. **Reconfigurable Intelligent Surfaces for 6G Systems: Principles, Applications, and Research Directions**. *IEEE Communications Magazine* 59:6 (2021), 14–20. DOI: 10.1109/MCOM.001.2001076 (see page 13).

[80] Zheng Chu, Pei Xiao, De Mi, Wanming Hao, Zihuai Lin, Qingchun Chen and Rahim Tafazolli. **Wireless Powered Intelligent Radio Environment with Non-Linear Energy Harvesting**. *IEEE Internet of Things Journal* (2022), 1–1. DOI: 10.1109/JIOT.2022.3162761 (see pages 13, 126, 127).

[81] Dingcai Zhang, Qingqing Wu, Miao Cui, Guangchi Zhang and Dusit Niyato. **Throughput Maximization for IRS-Assisted Wireless Powered Hybrid NOMA and TDMA**. *IEEE Wireless Communications Letters* 10:9 (2021), 1944–1948. DOI: 10.1109/LWC.2021.3087495 (see page 13).

[82] Maopeng Wu, Lijuan Su, Jianxun Chen, Xiaoli Duan, Donghua Wu, Yan Cheng and Yu Jiang. **Development and Prospect of Wireless Power Transfer Technology Used to Power Unmanned Aerial Vehicle**. *Electronics* 11:15 (2022), 2297 (see page 13).

[83] Zhongyu Wang, Tiejun Lv, Jie Zeng and Wei Ni. **Placement and Resource Allocation of Wireless-Powered Multiantenna UAV for Energy-Efficient Multiuser NOMA**. *IEEE Transactions on Wireless Communications* (2022), 1–1. DOI: 10.1109/TWC.2022.3169533 (see pages 13, 126).

[84] Alessandro Guidotti, Alessandro Vanelli-Coralli, Mohamed El Jaafari, Nicolas Chuberre, Jani Puttonen, Vincenzo Schena, Giuseppe Rinelli and Stefano Cioni. **Role and Evolution of Non-Terrestrial Networks Toward 6G Systems**. *IEEE Access* 12 (2024), 55945–55963. DOI: 10.1109/ACCESS.2024.3389459 (see pages 14, 15).

[85] 3GPP. **5G; Study on Using Satellite Access in 5G**. Technical Report (TR) 22.822. 3rd Generation Partnership Project (3GPP), July 2018. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3372 (see page 14).

[86] 3GPP. **Unmanned Aerial System (UAS) support in 3GPP**. Technical Specification (TS) 22.125. 3rd Generation Partnership Project (3GPP), June 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3545 (see page 15).

[87] 3GPP. **Support of Uncrewed Aerial Systems (UAS) connectivity, identification and tracking; Stage 2**. Technical Specification (TS) 23.256. 3rd Generation Partnership Project (3GPP), Dec. 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3853 (see page 15).

[88] 3GPP. **Study on Remote Identification of Unmanned Aerial Systems (UAS)**. Technical Report (TR) 22.822. 3rd Generation Partnership Project (3GPP), Sept. 2018. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3527 (see page 15).

[89] 3GPP. **Study on security aspects of Uncrewed Aerial Systems (UAS)**. Technical Report (TR) 33.854. 3rd Generation Partnership Project (3GPP), Dec. 2021. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3763 (see page 15).

[90] 3GPP. **Application layer support for Uncrewed Aerial System (UAS); Functional architecture and information flows**. Technical Specification (TS) 23.255. 3rd Generation Partnership Project (3GPP), Sept. 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3843 (see page 15).

[91] 3GPP. **NR; NR and NG-RAN Overall description; Stage-2**. Technical Specification (TS) 38.300. 3rd Generation Partnership Project (3GPP), Dec. 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3191 (see pages 15, 157).

[92] 3GPP. **NR; Medium Access Control (MAC) protocol specification**. Technical Specification (TS) 38.321. 3rd Generation Partnership Project (3GPP), Dec. 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3194 (see page 15).

[93] 3GPP. **Service requirements for the 5G system**. Technical Specification (TS) 22.261. 3rd Generation Partnership Project (3GPP), Sept. 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3107 (see pages 15, 157).

[94] 3GPP. **NR; User Equipment (UE) radio access capabilities**. Technical Specification (TS) 38.306. 3rd Generation Partnership Project (3GPP), Dec. 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3193 (see page 15).

[95] 3GPP. **NR; NR and NG-RAN Overall description; Stage-2**. Technical Specification (TS) 38.101-1. 3rd Generation Partnership Project (3GPP), Oct. 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3283 (see page 15).

[96] 3GPP. **NR; Radio Resource Control (RRC); Protocol specification**. Technical Specification (TS) 38.331. 3rd Generation Partnership Project (3GPP), Dec. 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3197 (see pages 15, 157).

[97] 3GPP. **NG-RAN; NG Application Protocol (NGAP)**. Technical Specification (TS) 38.413. 3rd Generation Partnership Project (3GPP), Dec. 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3223 (see page 15).

[98] 3GPP. **NG-RAN; Xn Application Protocol (XnAP)**. Technical Specification (TS) 38.423. 3rd Generation Partnership Project (3GPP), Dec. 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3228 (see pages 15, 157).

[99] 3GPP. **NG-RAN; F1 Application Protocol (F1AP)**. Technical Specification (TS) 38.473. 3rd Generation Partnership Project (3GPP), Dec. 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3260 (see pages 15, 157).

[100] N. Yumi, S. Yuki and O. Tetsuya. **An effective visual programming tool for learning and using robotics middleware**. In: *2016 IEEE/SICE International Symposium on System Integration (SII)*. 2016, 156–161. DOI: 10.1109/SII.2016.7843991 (see page 17).

[101] J. Boubin and C. Stewart. **SoftwarePilot: Fully Autonomous Aerial Systems Made Easier**. In: *2020 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C)*. 2020, 250–251. DOI: 10.1109/ACSOS-C51401.2020.00071 (see page 17).

[102] Anis Koubâa, Basit Qureshi, Mohamed-Foued Sriti, Azza Allouch, Yasir Javed, Maram Alajlan, Omar Cheikhrouhou, Mohamed Khalgui and Eduardo Tovar. **Dronemap planner: A service-oriented cloud-based management system for the internet-of-drones**. *Ad Hoc Netw.* 86 (2019), 46–62. DOI: 10.1016/j.adhoc.2018.09.013 (see page 18).

[103] G. Kakamoukas, P. Sarigiannidis and I. Moscholios. **High Level Drone Application Enabler: An Open Source Architecture**. In: *2020 12th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*. 2020, 1–4. DOI: 10.1109/CSNDSP49049.2020.9249442 (see page 18).

[104] E. Frew, K. Glasheen, C. Alexander Hirst, J. Bird and B. Argrow. **A Dispersed Autonomy Architecture for Information-Gathering Drone Swarms**. In: *2020 IEEE Aerospace Conference*. 2020, 1–11. DOI: 10.1109/AERO47225.2020.9172646 (see page 18).

[105] Aliaa Essameldin and Khaled A. Harras. **The Hive: An Edge-Based Middleware Solution for Resource Sharing in the Internet of Things**. In: *3rd Workshop on Experiences with the Design and Implementation of Smart Objects*. 2017, 13–18 (see page 18).

[106] D. Marcheras, M. Ayaida, N. Messai and F. Valentin. **A new middleware for managing heterogeneous robot in ubiquitous environments**. In: *2020 8th International Conference on Wireless Networks and Mobile Communications (WINCOM)*. 2020, 1–5. DOI: 10.1109/WINCOM50532.2020.9272477 (see page 18).

[107] Juan Besada, Luca Bergesio, Iván Campaña, Diego Vaquero-Melchor, Jaime López-Araquistain, Ana Bernardos and José Casar. **Drone Mission Definition and Implementation for Automated Infrastructure Inspection Using Airborne Sensors**. *Sensors* 18:4 (Apr. 2018), 1170. ISSN: 1424-8220. DOI: 10.3390/s18041170 (see page 18).

[108] V. Chamola, V. Hassija, V. Gupta and M. Guizani. **A Comprehensive Review of the COVID-19 Pandemic and the Role of IoT, Drones, AI, Blockchain, and 5G in Managing its Impact**. *IEEE Access* 8 (2020), 90225–90265. DOI: 10.1109/ACCESS.2020.2992341 (see page 29).

[109] J. Chakareski, S. Naqvi, N. Mastronarde, J. Xu, F. Afghah and A. Razi. **An Energy Efficient Framework for UAV-Assisted Millimeter Wave 5G Heterogeneous Cellular Networks**. *IEEE Trans. Green Commun. Netw.* 3:1 (2019), 37–44. DOI: 10.1109/TGCN.2019.2892141 (see page 30).

[110] N. Zhao, X. Pang, Z. Li, Y. Chen, F. Li, Z. Ding and M. Alouini. **Joint Trajectory and Precoding Optimization for UAV-Assisted NOMA Networks**. *IEEE Trans. Commun.* 67:5 (2019), 3723–3735. DOI: 10.1109/TCOMM.2019.2895831 (see page 30).

[111] Giovanni Grieco, Roberto Artuso, Pietro Boccadoro, Giuseppe Piro and Luigi Alfredo Grieco. **An Open Source and System-Level Simulator for the Internet of Drones**. In: *2019 IEEE 30th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC Workshops)*. 2019, 1–6. DOI: 10.1109/PIMRCW.2019.8880832 (see page 33).

[112] Mattia Sandri, Matteo Pagin, Marco Giordani and Michele Zorzi. **Implementation of a Channel Model for Non-Terrestrial Networks in ns-3**. In: *Workshop on ns-3 (WNS3)*. Arlington, VA, USA, 2023, 28–34. DOI: 10.1145/3592149.3592158 (see pages 34, 35, 56, 120).

[113] Sabur Baidya, Zoheb Shaikh and Marco Levorato. **FlyNetSim: An Open Source Synchronized UAV Network Simulator Based on Ns-3 and Ardupilot**. In: *Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. MSWIM '18. Montreal, QC, Canada: Association for Computing Machinery, 2018, 37–45. ISBN: 9781450359603. DOI: 10.1145/3242102.3242118. URL: https://doi.org/10.1145/3242102.3242118 (see pages 35, 36).

[114] N. R. Zema, A. Trotta, G. Sanahuja, E. Natalizio, M. Di Felice and L. Bononi. **CUSCUS: An integrated simulation architecture for distributed networked control systems**. In: *2017 14th IEEE Annual Consumer Communications Networking Conference (CCNC)*. 2017, 287–292. DOI: 10.1109/CCNC.2017.7983121 (see pages 35–37).

[115] Emerson Alberto Marconato, Mariana Rodrigues, Rayner de Melo Pires, Daniel Fernando Pigatto, Alex Roschildt Pinto, Kalinka RLJC Branco et al. **Avens-a novel flying ad hoc network simulator with automatic code generation for unmanned aircraft system**. In: *Proceedings of the 50th Hawaii international conference on system sciences*. 2017 (see pages 35–37).

[116] J. A. Millan-Romera, J. J. Acevedo, À. R. Castaño, H. Perez-Leon, C. Capitán and A. Ollero. **A UTM simulator based on ROS and Gazebo**. In: *2019 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED UAS)*. 2019, 132–141. DOI: 10.1109/REDUAS47371.2019.8999705 (see pages 35–37).

[117] Mauro Tropea, Peppino Fazio, Floriano De Rango and Nicola Cordeschi. **A New FANET Simulator for Managing Drone Networks and Providing Dynamic Connectivity**. *Electronics* 9:4 (2020). ISSN: 2079-9292. DOI: 10.3390/electronics9040543. URL: https://www.mdpi.com/2079-9292/9/4/543 (see pages 35–37).

[118] Srikrishna Acharya, Bharadwaj Amrutur, Mukunda Bharatheesha and Yogesh Simmhan. *CORNET 2.0: A Co-Simulation Middleware forRobot Networks*. 2021. arXiv: 2109.06979 [cs.RO] (see pages 35–37).

[119] Seongjoon Park, Woong Gyu La, Woonghee Lee and Hwangnam Kim. **Devising a Distributed Co-Simulator for a Multi-UAV Network**. *Sensors* 20:21 (2020), 6196 (see pages 35–38).

[120] Jani Puttonen, Sami Rantanen, Frans Laakso, Janne Kurjenniemi, Kari Aho and Guray Acar. **Satellite Model for Network Simulator 3**. In: ICST, Aug. 2014, 1–6. DOI: 10.4108/icst.simutools.2014.254631 (see page 35).

[121] Hyuckjin Choi and Junil Choi. **WiThRay: Versatile 3D Simulator for Intelligent Reflecting Surface-aided MmWave Systems**. In: *2021 International Symposium on Antennas and Propagation (ISAP)*. 2021, 1–2 (see pages 35, 38).

[122] Ertugrul Basar and Ibrahim Yildirim. **SimRIS Channel Simulator for Reconfigurable Intelligent Surface-Empowered Communication Systems**. In: *2020 IEEE Latin-American Conference on Communications (LATINCOM)*. 2020, 1–6 (see pages 35, 38).

[123] Le Hao, Agnes Fastenbauer, Stefan Schwarz and Markus Rupp. **Towards System Level Simulation of Reconfigurable Intelligent Surfaces**. In: *2022 International Symposium ELMAR*. 2022, 81–84 (see pages 35, 38, 39).

[124] Matteo Pagin, Marco Giordani, Amir Ashtari Gargari, Alberto Rech, Federico Moretto, Stefano Tomasin, Jonathan Gambini and Michele Zorzi. **End-to-End Simulation of 5G Networks Assisted by IRS and AF Relays**. In: *2022 20th Mediterranean Communication and Computer Networking Conference (MedComNet)*. 2022, 150–157 (see pages 35, 38).

[125] Bjorn Sihlbom, Marios I. Poulakis and Marco Di Renzo. **Reconfigurable Intelligent Surfaces: Performance Assessment Through a System-Level Simulator**. *IEEE Wireless Communications* (2022), 1–10 (see pages 35, 38).

[126] Qi GU, Dan WU, Xin SU, Hanning Wang, Jingyuan Cui and Yifei Yuan. **System-level Simulation of RIS assisted Wireless Communications System**. In: *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*. 2022, 1540–1545 (see pages 35, 38).

[127] Zahed Hossain, Qing Xia and Josep Miquel Jornet. **TeraSim: An ns-3 extension to simulate Terahertz-band communication networks**. *Nano Communication Networks* 17 (2018), 36–44. ISSN: 1878-7789. DOI: https://doi.org/10.1016/j.nancom.2018.08.001. URL: https://www.sciencedirect.com/science/article/pii/S1878778918300772 (see page 35).

[128] George F. Riley and Thomas R. Henderson, 15–34. In: *Modeling and Tools for Network Simulation*. Ed. by Klaus Wehrle, Mesut Güneş and James Gross. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. ISBN: 978-3-642-12331-3. DOI: 10.1007/978-3-642-12331-3_2. URL: https://doi.org/10.1007/978-3-642-12331-3_2 (see pages 35, 38, 41).

[129] Natale Patriciello, Sandra Lagen, Biljana Bojovic and Lorenza Giupponi. **An E2E simulator for 5G NR networks**. *Simulation Modelling Practice and Theory* 96 (2019), 101933. ISSN: 1569-190X. DOI: https://doi.org/10.1016/j.simpat.2019.101933. URL: https://www.sciencedirect.com/science/article/pii/S1569190X19300589 (see page 35).

[130] Andras Varga, 35–59. In: *Modeling and Tools for Network Simulation*. Ed. by Klaus Wehrle, Mesut Güneş and James Gross. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. ISBN: 978-3-642-12331-3. DOI: 10.1007/978-3-642-12331-3_3. URL: https://doi.org/10.1007/978-3-642-12331-3_3 (see page 35).

[131] Stefan Pratschner, Bashar Tahir, Ljiljana Marijanovic, Mariam Mussbah, Kiril Kirev, Ronald Nissel, Stefan Schwarz and Markus Rupp. **Versatile mobile communications simulation: the Vienna 5G Link Level Simulator**. *EURASIP Journal on Wireless Communications and Networking* 2018:1 (Sept. 2018), 226. DOI: 10.1186/s13638-018-1239-6 (see page 35).

[132] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng et al. **ROS: an open-source Robot Operating System**. In: *ICRA workshop on open source software*. Vol. 3. 3.2. Kobe, Japan. 2009, 5 (see page 35).

[133] N. Koenig and A. Howard. **Design and use paradigms for Gazebo, an open-source multi-robot simulator**. In: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*. Vol. 3. 2004, 2149–2154 vol.3. DOI: 10.1109/IROS.2004.1389727 (see page 35).

[134] 3GPP. **5G; Study on channel model for frequencies from 0.5 to 100 GHz**. Technical Report (TR) 38.901. Version 16.1.0. 3rd Generation Partnership Project (3GPP), Nov. 2020. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3173 (see pages 38, 56).

[135] EPSG. *WGS 84 / Pseudo-Mercator – Spherical Mercator, Google Maps, OpenStreetMap, Bing, ArcGIS, ESRI*. URL: https://epsg.io/3857 (visited on 04/07/2023) (see page 45).

[136] CM Shepherd. **Theoretical design of primary and secondary cells. part 3. battery discharge equation**. Tech. rep. NAVAL RESEARCH LAB WASHINGTON DC, 1963 (see page 50).

[137] Olivier Tremblay, Louis-A Dessaint and Abdel-Illah Dekkiche. **A generic battery model for the dynamic simulation of hybrid electric vehicles**. In: *2007 IEEE Vehicle Power and Propulsion Conference*. Ieee. 2007, 284–289 (see page 50).

[138] Sun, Y. and Xu, D. and Ng, D. W. K. and Dai, L. and Schober, R. **Optimal 3D-trajectory design and resource allocation for solar-powered UAV communication systems**. *IEEE Transactions on Communications* 67(6) (2019), 4281–4298 (see page 50).

[139] Justin Morris, Kazim Ergun, Behnam Khaleghi, Mohen Imani, Baris Aksanli and Tajana Simunic. **HyDREA: Utilizing Hyperdimensional Computing for a More Robust and Efficient Machine Learning System**. *ACM Trans. Embed. Comput. Syst.* 21:6 (Oct. 2022). ISSN: 1539-9087. DOI: 10.1145/3524067. URL: https://doi.org/10.1145/3524067 (see page 56).

[140] Joseph Benin, Michael Nowatkowski and Henry Owen. **Vehicular Network simulation propagation loss model parameter standardization in ns-3 and beyond**. In: *2012 Proceedings of IEEE Southeastcon*. 2012, 1–5. DOI: 10.1109/SECon.2012.6196929 (see page 56).

[141] *ITU-R P.1238*. https://www.itu.int/rec/R-REC-P.1238/en. Accessed: 31/12/2024 (see page 56).

[142] *ITU-R P.1411*. https://www.itu.int/rec/R-REC-P.1411/en. Accessed: 31/12/2024 (see page 56).

[143] ITU. **Attenuation by atmospheric gases and related effects**. *Reccomendation P.676* (2013) (see page 58).

[144] Giovanni Iacovelli, Angelo Coluccia and Luigi Alfredo Grieco. **Multi-UAV IRS-Assisted Communications: Multi-Node Channel Modeling and Fair Sum-Rate Optimization via Deep Reinforcement Learning**. *IEEE Internet of Things Journal* (2023), 1–1. DOI: 10.1109/JIOT.2023.3299018 (see pages 59–61).

[145] Mohammad A Al-Jarrah, Ki-Hong Park, Arafat Al-Dweik and Mohamed-Slim Alouini. **Error Rate Analysis of Amplitude-Coherent Detection over Rician Fading Channels with Receiver Diversity**. *IEEE Transactions on Wireless Communications* 19:1 (2019), 134–147 (see page 59).

[146] ETSI. **Reconfigurable Intelligent Surfaces (RIS); Communication Models, Channel Models, Channel Estimation and Evaluation Methodology**. Group Report (GR) 003. Version 1.1.1. European Telecommunications Standards Institute (ETSI), June 2023. URL: https://www.etsi.org/deliver/etsi%5C_gr/RIS/001%5C_099/003/01.01.01%5C_60/gr%5C_RIS003v010101p.pdf (see page 90).

[147] S Senthil Kumaran. **Practical LXC and LXD: linux containers for virtualization and orchestration**. Springer, 2017 (see page 105).

[148] 3GPP. **LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures**. Technical Specification (TS) 36.213. Version 8.3.0 Release 8. 3rd Generation Partnership Project (3GPP), Nov. 2008. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2427 (see page 111).

[149] Marco Mezzavilla, Marco Miozzo, Michele Rossi, Nicola Baldo and Michele Zorzi. **A lightweight and accurate link abstraction model for the simulation of LTE networks in ns-3**. In: *Proceedings of the 15th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*. 2012, 55–60 (see page 114).

[150] Min-Gyu Kim and Han-Shin Jo. **Performance Analysis of NB-IoT Uplink in Low Earth Orbit Non-Terrestrial Networks**. *Sensors* 22:18 (2022). ISSN: 1424-8220. DOI: 10.3390/s22187097. URL: https://www.mdpi.com/1424-8220/22/18/7097 (see page 125).

[151] Mohammad Mahdi Azari, Fernando Rosas, Kwang-Cheng Chen and Sofie Pollin. **Ultra Reliable UAV Communication Using Altitude and Cooperation Diversity**. *IEEE Trans. Commun.* 66:1 (2017), 330–344 (see page 126).

[152] Yunfei Chen, Nan Zhao and Mohamed-Slim Alouini. **Wireless Energy Harvesting Using Signals From Multiple Fading Channels**. *IEEE Transactions on Communications* 65:11 (2017), 5027–5039. DOI: 10.1109/TCOMM.2017.2734665 (see page 127).

[153] Victor Monzon Baeza, Eva Lagunas, Hayder Al-Hraishawi and Symeon Chatzinotas. **An Overview of Channel Models for NGSO Satellites**. In: *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)*. 2022, 1–6. DOI: 10.1109/VTC2022-Fall57202.2022.10012693 (see page 127).

[154] Niloofar Okati, Taneli Riihonen, Dani Korpi, Ilari Angervuori and Risto Wichman. **Downlink Coverage and Rate Analysis of Low Earth Orbit Satellite Constellations Using Stochastic Geometry**. *IEEE Transactions on Communications* 68:8 (2020), 5120–5134. DOI: 10.1109/TCOMM.2020.2990993 (see page 127).

[155] Qingyue Chen, Zhugang Wang, Gert Frølund Pedersen and Ming Shen. **Joint Satellite-Transmitter and Ground-Receiver Digital Pre-Distortion for Active Phased Arrays in LEO Satellite Communications**. *Remote Sensing* 14:17 (2022). ISSN: 2072-4292 (see page 127).

[156] Constantine A. Balanis. **Antenna Theory: Analysis and Design, 2nd Edition**. Wiley, 1996 (see pages 127, 128, 144).

[157] L. J.Ippolito. **Satellite Communications Systems Engineering: Atmospheric Effects, Satellite Link Design and System Performance, 2nd Edition**. Wiley, 2017. ISBN: 1119259371 (see page 128).

[158] ITU. **Attenuation by atmospheric gases and related effects**. Recommendation. ITU-R P.676-12. International Telecommunication Union (ITU), 2019 (see pages 128, 129).

[159] ITU. **Attenuation due to clouds and fog**. Recommendation. ITU-R P.840-8. International Telecommunication Union (ITU), 2019 (see pages 128, 129).

[160] ITU. **Propagation data and prediction methods required for the design of Earth-space telecommunication systems**. Tech. rep. Recommendations Radiowave Propag. ITU-R 618-12. International Telecommunication Union, 2015 (see pages 128, 129).

[161] ITU. **Propagation data and prediction methods required for the design of Earth-space telecommunication systems**. Recommendation. ITU-R P.618-13. International Telecommunication Union (ITU), 2017 (see pages 128, 129).

[162] ITU. **Characteristics of precipitation for propagation modelling**. Recommendation. ITU-R P.837-7. International Telecommunication Union (ITU), 2021 (see pages 128, 129).

[163] 3GPP. **Study on provision of low-cost Machine-Type Communications (MTC) User Equipments (UEs) based on LTE**. Tech. rep. TR 36.888. Release 12. 3rd Generation Partnership Project (3GPP), June 2013 (see page 129).

[164] 3GPP. **Medium Access Control (MAC) protocol specification**. Tech. rep. TS 36.321. version 15.8.0 Release 15. Evolved Universal Terrestrial Radio Access (E-UTRA), Jan. 2020 (see page 130).

[165] John R Barry, Edward A Lee and David G Messerschmitt. **Digital communication**. Springer Science & Business Media, 2003 (see pages 132, 133).

[166] **Guidelines for evaluation of radio interface technologies for IMT-Advanced – Report ITU-R M.2135-1**. Tech. rep. International Telecommunication Union (ITU), 2009 (see page 144).

[167] Qingqing Zhang, Gang Wang, Jie Chen, Georgios B. Giannakis and Qingwen Liu. **Mobile Energy Transfer in Internet of Things**. *IEEE Internet of Things Journal* 6:5 (2019), 9012–9019. DOI: 10.1109/JIOT.2019.2926333 (see page 144).

[168] Y. Zeng, J. Xu and R. Zhang. **Energy Minimization for Wireless Communication With Rotary-Wing UAV**. *IEEE Trans. Wireless Commun.* 18:4 (Apr. 2019), 2329–2345 (see page 145).

[169] Shen Wang, M. Atif Qureshi, Luis Miralles-Pechuán, Thien Huynh-The, Thippa Reddy Gadekallu and Madhusanka Liyanage. **Explainable AI for 6G Use Cases: Technical Aspects and Research Challenges**. *IEEE Open Journal of the Communications Society* 5 (2024), 2490–2540. DOI: 10.1109/OJCOMS.2024.3386872 (see pages 149, 151).

[170] Nasir Khan, Sinem Coleri, Asmaa Abdallah, Abdulkadir Celik and Ahmed M. Eltawil. **Explainable and Robust Artificial Intelligence for Trustworthy Resource Management in 6G Networks**. *IEEE Communications Magazine* 62:4 (2024), 50–56. DOI: 10.1109/MCOM.001.2300172 (see pages 149, 150).

[171] Tobias Lundby, Martin Peter Christiansen and Kjeld Jensen. **Towards a Weather Analysis Software Framework to Improve UAS Operational Safety**. In: *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*. 2019, 1372–1380. DOI: 10.1109/ICUAS.2019.8798271 (see pages 150, 151).

[172] Mozhou Gao, Chris H Hugenholtz, Thomas A Fox, Maja Kucharczyk, Thomas E Barchyn and Paul R Nesbit. **Weather constraints on global drone flyability**. *Scientific Reports* 11:1 (2021), 12092 (see page 151).

[173] Chengyi Qu, Francesco Betti Sorbelli, Rounak Singh, Prasad Calyam and Sajal K. Das. **Environmentally-Aware and Energy-Efficient Multi-Drone Coordination and Networking for Disaster Response**. *IEEE Transactions on Network and Service Management* 20:2 (2023), 1093–1109. DOI: 10.1109/TNSM.2023.3243543 (see pages 151, 153).

[174] Chenrui Sun, Gianluca Fontanesi, Berk Canberk, Amirhossein Mohajerzadeh, Symeon Chatzinotas, David Grace and Hamed Ahmadi. **Advancing UAV Communications: A Comprehensive Survey of Cutting-Edge Machine Learning Techniques**. *IEEE Open Journal of Vehicular Technology* (2024), 1–31. DOI: 10.1109/OJVT.2024.3401024 (see page 151).

[175] Hajiar Yuliana, Iskandar and Hendrawan. **A Review of Coverage Predictions in Mobile Communication Systems using Machine Learning Algorithm**. In: *2023 17th International Conference on Telecommunication Systems, Services, and Applications (TSSA)*. 2023, 1–7. DOI: 10.1109/TSSA59948.2023.10367027 (see pages 151, 152).

[176] Laurențiu Gabriel Militaru, Dan Popescu and Loretta Ichim. **4G/LTE Issues of Low Altitude UAV Flying Systems**. In: *2020 24th International Conference on System Theory, Control and Computing (ICSTCC)*. 2020, 874–879. DOI: 10.1109/ICSTCC50638.2020.9259755 (see page 151).

[177] Dave Townend and Stuart D. Walker. **A 3D Statistical Framework for the UK's Mobile Network**. In: *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*. 2020, 1–5. DOI: 10.1109/PIMRC48278.2020.9217304 (see page 151).

[178] Usama Masood, Hasan Farooq, Ali Imran and Adnan Abu-Dayya. **Interpretable AI-Based Large-Scale 3D Pathloss Prediction Model for Enabling Emerging Self-Driving Networks**. *IEEE Transactions on Mobile Computing* 22:7 (2023), 3967–3984. DOI: 10.1109/TMC.2022.3147191 (see page 152).

[179] Melina Geis, Benjamin Sliwa, Caner Bektas and Christian Wietfeld. **TinyDRaGon: Lightweight Radio Channel Estimation for 6G Pervasive Intelligence**. In: *2022 IEEE Future Networks World Forum (FNWF)*. 2022, 658–663. DOI: 10.1109/FNWF55208.2022.00121 (see page 152).

[180] Shota Yamada, Takeo Fujii, Katsuya Suto and Koya Sato. **Observation Data and 3D Map-based Radio Environment Estimation for Drone Wireless Communications**. In: *2023 Fourteenth International Conference on Ubiquitous and Future Networks (ICUFN)*. 2023, 70–75. DOI: 10.1109/ICUFN57995.2023.10199770 (see page 152).

[181] Eduardo Nuno Almeida, Helder Fontes, Rui Campos and Manuel Ricardo. **Position-Based Machine Learning Propagation Loss Model Enabling Fast Digital Twins of Wireless Networks in ns-3**. In: *Proceedings of the 2023 Workshop on Ns-3*. WNS3 '23. , Arlington, VA, USA, Association for Computing Machinery, 2023, 69–77. ISBN: 979-8-40-070747-6. DOI: 10.1145/3592149.3592150. URL: https://doi.org/10.1145/3592149.3592150 (see page 153).

[182] 3GPP. **Architecture enhancements for 5G System (5GS) to support network data analytics services**. Technical Specification (TS) 23.288. 3rd Generation Partnership Project (3GPP), Dec. 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3579 (see page 157).

[183] 3GPP. **Telecommunication management; Performance Management (PM); Performance measurements Evolved Universal Terrestrial Radio Access Network (E-UTRAN)**. Technical Specification (TS) 32.425. 3rd Generation Partnership Project (3GPP), Apr. 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2011 (see page 157).

[184] 3GPP. **Evolved Universal Terrestrial Radio Access (E-UTRA) and NR; Multi-connectivity; Overall Description; Stage-2**. Technical Specification (TS) 37.340. 3rd Generation Partnership Project (3GPP), Dec. 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3198 (see page 157).

[185] 3GPP. **LTE Positioning Protocol (LPP)**. Technical Specification (TS) 37.355. 3rd Generation Partnership Project (3GPP), Dec. 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3710 (see page 157).

[186] 3GPP. **E1 general aspects and principles**. Technical Specification (TS) 37.480. 3rd Generation Partnership Project (3GPP), Mar. 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3954 (see page 157).

[187] 3GPP. **E1 Application Protocol (E1AP)**. Technical Specification (TS) 37.483. 3rd Generation Partnership Project (3GPP), Dec. 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3957 (see page 157).

[188] 3GPP. **NR; Requirements for support of radio resource management**. Technical Specification (TS) 38.133. 3rd Generation Partnership Project (3GPP), Oct. 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3204 (see page 157).

[189] 3GPP. **NR; Physical layer procedures for data**. Technical Specification (TS) 38.214. 3rd Generation Partnership Project (3GPP), Sept. 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3216 (see page 157).

[190] 3GPP. **NR; Physical layer procedures for data**. Technical Specification (TS) 38.215. 3rd Generation Partnership Project (3GPP), Sept. 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3216 (see page 157).

[191] 3GPP. **NG Radio Access Network (NG-RAN); Stage 2 functional specification of User Equipment (UE) positioning in NG-RAN**. Technical Specification (TS) 38.305. 3rd Generation Partnership Project (3GPP), Dec. 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3310 (see page 157).

[192] 3GPP. **System architecture for the 5G System (5GS)**. Technical Specification (TS) 23.501. 3rd Generation Partnership Project (3GPP), Dec. 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3144 (see page 157).

[193] 3GPP. **NG-RAN; Architecture description**. Technical Specification (TS) 38.401. 3rd Generation Partnership Project (3GPP), Dec. 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3219 (see page 157).

[194] 3GPP. **NG-RAN; Xn general aspects and principles**. Technical Specification (TS) 38.420. 3rd Generation Partnership Project (3GPP), Mar. 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3225 (see page 157).

[195] 3GPP. **NG-RAN; NR Positioning Protocol A (NRPPa)**. Technical Specification (TS) 38.455. 3rd Generation Partnership Project (3GPP), Dec. 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3256 (see page 157).

[196] 3GPP. **NG-RAN; F1 general aspects and principles**. Technical Specification (TS) 38.470. 3rd Generation Partnership Project (3GPP), Dec. 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3257 (see page 157).

[197] 3GPP. **Procedures for the 5G System (5GS)**. Technical Specification (TS) 23.502. 3rd Generation Partnership Project (3GPP), Jan. 2025. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3145 (see page 157).

[198] 3GPP. **Policy and charging control framework for the 5G System (5GS); Stage 2**. Technical Specification (TS) 23.503. 3rd Generation Partnership Project (3GPP), Dec. 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3334 (see page 157).

[199] 3GPP. **Management and orchestration; 5G Network Resource Model (NRM); Stage 2 and stage 3**. Technical Specification (TS) 28.541. 3rd Generation Partnership Project (3GPP), Dec. 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3400 (see page 157).

[200] 3GPP. **Management and orchestration; 5G performance measurements**. Technical Specification (TS) 28.552. 3rd Generation Partnership Project (3GPP), Sept. 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3413 (see page 157).

[201] 3GPP. **Management and orchestration; 5G end to end Key Performance Indicators (KPI)**. Technical Specification (TS) 28.554. 3rd Generation Partnership Project (3GPP), Sept. 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3415 (see page 157).

[202] 3GPP. **Telecommunication management; Charging management; Exposure function Northbound Application Program Interfaces (APIs) charging**. Technical Specification (TS) 32.254. 3rd Generation Partnership Project (3GPP), July 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3275 (see page 157).

[203] 3GPP. **Telecommunication management; Charging management; 5G system, charging service; Stage 3**. Technical Specification (TS) 32.291. 3rd Generation Partnership Project (3GPP), Sept. 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3398 (see page 157).

[204] 3GPP. **Management and orchestration; Artificial Intelligence/ Machine Learning (AI/ML) management**. Technical Specification (TS) 28.105. 3rd Generation Partnership Project (3GPP), Sept. 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3970 (see page 157).

[205] 3GPP. **Study on Artificial Intelligence (AI)/Machine Learning (ML) for NR air interface**. Technical Specification (TS) 38.843. 3rd Generation Partnership Project (3GPP), Jan. 2024. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3983 (see page 157).

[206] *Delegated regulation - 2019/945 - EN - EUR-Lex*. en. Doc ID: 32019R0945 Doc Sector: 3 Doc Title: Commission Delegated Regulation (EU) 2019/945 of 12 March 2019 on unmanned aircraft systems and on third-country operators of unmanned aircraft systems Doc Type: R Usr_lan: en. URL: https://eur-lex.europa.eu/eli/reg_del/2019/945/oj (visited on 14/08/2024) (see pages 159, 160, 163).

[207] G. Markarian, R. Karlović, H. Nitsch and K. Chandramouli. **Security Technologies and Social Implications**. Wiley, 2022. DOI: 10.1002/9781119834175 (see page 159).

[208] European Commission, Joint Research Centre, M Stępniak, I Cheimariotis, C Lodi, M Rataj, J Zawieska, M Grosso and A Marotta. **Research and innovation on drones in Europe – An assessment based on the Transport Research and Innovation Monitoring and Information System (TRIMIS)**. Publications Office of the European Union, 2024. DOI: 10.2760/02357 (see pages 159, 161).

[209] *Implementing regulation - 2019/947 - EN - EUR-Lex*. en. Doc ID: 32019R0947 Doc Sector: 3 Doc Title: Commission Implementing Regulation (EU) 2019/947 of 24 May 2019 on the rules and procedures for the operation of unmanned aircraft (Text with EEA relevance.) Doc Type: R Usr_lan: en. URL: https://eur-lex.europa.eu/eli/reg_impl/2019/947/oj (visited on 20/07/2024) (see pages 159, 160, 163).

[210] Counter Terrorism Preparedness Network, C Church, V. Liberatori, G. Markerian, K. Otmane, M. Paterson, Project Stadia, A. Staniforth, A. Townsend-Drake, K. Wright and P. Monk. **Preparing for Hostile Drones in Urban Environments: Report 2024**. Counter Terrorism Preparedness Network, 2024 (see page 160).

[211] Markus Krueckemeier, Fabian Schwartau, Sebastian Paul and Joerg Schoebel. **Passive Radar Transmitter Localization Using a Planar Approximation**. *IEEE Transactions on Aerospace and Electronic Systems* 57:5 (2021), 3405–3415. DOI: 10.1109/TAES.2021.3074133 (see page 160).

[212] Liping Di and Eugene Yu. **Remote Sensing Big Data**. Springer Nature, 2023. DOI: 10.1007/978-3-031-33932-5 (see pages 160, 162–165, 167).

[213] Jian Wang, Yongxin Liu and Houbing Song. **Counter-Unmanned Aircraft System(s) (C-UAS): State of the Art, Challenges, and Future Trends**. *IEEE Aerospace and Electronic Systems Magazine* 36:3 (2021), 4–29. DOI: 10.1109/MAES.2020.3015537 (see pages 160, 161, 166, 168).

[214] European Commission. *Regulation (EU) 2018/1139 of the European Parliament and of the Council of 4 July 2018 on common rules in the field of civil aviation and establishing a European Union Aviation Safety Agency*. 2018. URL: http://data.europa.eu/eli/reg/2018/1139/oj (see page 160).

[215] European Commission. *Action Plan to support the protection of public spaces, COM/2017/0612 final*. 2017. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52017DC0612 (see page 160).

[216] European Commission. *Communication from the Commission to the European Parliament, the European Council, the Council, the European Economic and Social Committee and the Committee of the Regions on the EU Security Union Strategy, COM/2020/605 final*. 2020. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52020DC0605 (see page 160).

[217] European Commission. *Communication from the Commission to the European Parliament, the European Council, the Council, the European Economic and Social Committee and the Committee of the Regions – A Counter-Terrorism Agenda for the EU: Anticipate, Prevent, Protect, Respond, COM/2020/795 final*. 2020. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52020DC0795 (see page 160).

[218] European Commission. *Communication from the Commission to the Council and the European Parliament on countering potential threats posed by drones, COM/2023/659 final*. 2023. URL: https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=COM:2023:659:FIN (see page 161).

[219] Daniel Rauschen, Dennis Gläsel, Hans Peter Such, Philipp Zimmermann and Markus Antweiler. **Commercial Of The Shelf Counter UAV**. In: *2019 International Conference on Military Communications and Information Systems (ICMCIS)*. 2019, 1–5. DOI: 10.1109/ICMCIS.2019.8842707 (see page 161).

[220] Seongjoon Park, Hyeong Tae Kim, Sangmin Lee, Hyeontae Joo and Hwangnam Kim. **Survey on Anti-Drone Systems: Components, Designs, and Challenges**. *IEEE Access* 9 (2021), 42635–42659. DOI: 10.1109/ACCESS.2021.3065926 (see page 161).

[221] Georgia Lykou, Dimitrios Moustakas and Dimitris Gritzalis. **Defending Airports from UAS: A Survey on Cyber-Attacks and Counter-Drone Sensing Technologies**. *Sensors* 20:12 (2020). ISSN: 1424-8220. DOI: 10.3390/s20123537. URL: https://www.mdpi.com/1424-8220/20/12/3537 (see page 161).

[222] A. Melnichuk, E. A. Kuzina and N. K. Yurkov. **Methods and Means for Countering Unmanned Aerial Vehicles**. In: *2020 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*. 2020, 1–6. DOI: 10.1109/ICIEAM48468.2020.9112082 (see page 161).

[223] COURAGEOUS. *Review of current C-UAS Frameworks (Methods & Technologies) - D2.2 - V2.0*. 2022 (see pages 161, 165, 167–169).

[224] Rajanikanth Nagaraj Kashi, Anushka Prashanth, Sumukh Rajanikanth Kashi and Gayathri Prabhakara. **A survey and analysis of drone detection systems using a systems approach superposed on scenarios**. *Systems Engineering* 27:3 (2024), 598–636. DOI: https://doi.org/10.1002/sys.21735. eprint: https://incose.onlinelibrary.wiley.com/doi/pdf/10.1002/sys.21735. URL: https://incose.onlinelibrary.wiley.com/doi/abs/10.1002/sys.21735 (see page 161).

[225] Samuel S. Blackman and Robert Popoli. **Design and analysis of modern tracking systems**. The Artech House radar library. Artech House, 1999 (see page 161).

[226] Sebastien Hengy, Martin Laurenzis, Stéphane Schertzer, Alexander Hommes, Franck Kloeppel, Alex Shoykhetbrod, Thomas Geibig, Winfried Johannes, Oussama Rassy and Frank Christnacher. **Multimodal UAV detection: study of various intrusion scenarios**. In: *Electro-Optical Remote Sensing XI*. Ed. by Gary Kamerman and Ove Steinvall. Vol. 10434. International Society for Optics and Photonics. SPIE, 2017, 104340P. DOI: 10.1117/12.2278212. URL: https://doi.org/10.1117/12.2278212 (see page 161).

[227] C. G. Leela Krishna and Robin R. Murphy. **A review on cybersecurity vulnerabilities for unmanned aerial vehicles**. In: *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. 2017, 194–199. DOI: 10.1109/SSRR.2017.8088163 (see pages 161, 162).

[228] European Commission. *CRITICAL INFRASTRUCTURE PROTECTION & RESILIENCE - JRC DRONE Project to Counter UAS for CI*. 2022. URL: https://ec.europa.eu/newsroom/cipr/items/752427 (see page 161).

[229] European Commission, Joint Research Centre, P Hansen and R Pinto Faria. **Protection against Unmanned Aircraft Systems – Handbook on UAS protection of critical infrastructure and public space – A five phase approach for C-UAS stakeholders**. Publications Office of the European Union, 2023. DOI: doi/10.2760/18569 (see page 161).

[230] European Commission, Joint Research Centre, V Karlos and M Larcher. **Protection against Unmanned Aircraft Systems – Handbook on UAS risk assessment and principles for physical hardening of buildings and sites**. Publications Office of the European Union, 2023. DOI: doi/10.2760/969680 (see page 161).

[231] Vittorio Ugo Castrillo, Angelo Manco, Domenico Pascarella and Gabriella Gigante. **A Review of Counter-UAS Technologies for Cooperative Defensive Teams of Drones**. *Drones* 6:3 (2022). ISSN: 2504-446X. DOI: 10.3390/drones6030065. URL: https://www.mdpi.com/2504-446X/6/3/65 (see pages 162, 168).

[232] R. Yuvaraj and S. Velliangiri. **A Comprehensive Study on Unmanned Aerial Vehicle Security Issues**. In: *2023 IEEE 5th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA)*. 2023, 544–550. DOI: 10.1109/ICCCMLA58983.2023.10346935 (see page 162).

[233] Christos Laoudias, Adriano Moreira, Sunwoo Kim, Sangwoo Lee, Lauri Wirola and Carlo Fischione. **A Survey of Enabling Technologies for Network Localization, Tracking, and Navigation**. *IEEE Communications Surveys & Tutorials* 20:4 (2018), 3607–3644. DOI: 10.1109/COMST.2018.2855063 (see page 162).

[234] Nader Al-lQubaydhi, Abdulrahman Alenezi, Turki Alanazi, Abdulrahman Senyor, Naif Alanezi, Bandar Alotaibi, Munif Alotaibi, Abdul Razaque and Salim Hariri. **Deep learning for unmanned aerial vehicles detection: A review**. *Computer Science Review* 51 (2024), 100614. ISSN: 1574-0137. DOI: https://doi.org/10.1016/j.cosrev.2023.100614. URL: https://www.sciencedirect.com/science/article/pii/S1574013723000813 (see page 162).

[235] Muhammad Saqib, Sultan Daud Khan, Nabin Sharma and Michael Blumenstein. **A study on detecting drones using deep convolutional neural networks**. In: *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. 2017, 1–5. DOI: 10.1109/AVSS.2017.8078541 (see page 162).

[236] Stamatios Samaras, Eleni Diamantidou, Dimitrios Ataloglou, Nikos Sakellariou, Anastasios Vafeiadis, Vasilis Magoulianitis, Antonios Lalas, Anastasios Dimou, Dimitrios Zarpalas, Konstantinos Votis, Petros Daras and Dimitrios Tzovaras. **Deep Learning on Multi Sensor Data for Counter UAV Applications—A Systematic Review**. *Sensors* 19:22 (2019). ISSN: 1424-8220. DOI: 10.3390/s19224837. URL: https://www.mdpi.com/1424-8220/19/22/4837 (see pages 162, 169).

[237] M.W Gardner and S.R Dorling. **Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences**. *Atmospheric Environment* 32:14 (1998), 2627–2636. ISSN: 1352-2310. DOI: https://doi.org/10.1016/S1352-2310(97)00447-0. URL: https://www.sciencedirect.com/science/article/pii/S1352231097004470 (see page 162).

[238] Li Wang, Jun Tang and Qingmin Liao. **A Study on Radar Target Detection Based on Deep Neural Networks**. *IEEE Sensors Letters* 3:3 (2019), 1–4. DOI: 10.1109/LSENS.2019.2896072 (see page 162).

[239] R.A. Hoogendoorn and W.H.L. Neven. **ARTAS: Multisensor tracking in an ATC environment**. Tech. rep. National Aerospace Laboratory NLR, 1997 (see page 162).

[240] Bhargav Patel and Dmitri Rizer. **Counter-unmanned aircraft systems technology guide**. *New York: US Department of Homeland Security: Science and Technology Directorate* (2019) (see page 163).

[241] Esrat Farhana Dulia and Syed A. M. Shihab. **Designing a Surveillance Sensor Network with Information Clearinghouse for Advanced Air Mobility**. *Sensors* 24:3 (2024). ISSN: 1424-8220. DOI: 10.3390/s24030803. URL: https://www.mdpi.com/1424-8220/24/3/803 (see page 163).

[242] IEEE. **IEEE Standard Letter Designations for Radar-Frequency Bands**. *IEEE Std 521-2019 (Revision of IEEE Std 521-2002)* (2020), 1–15. DOI: 10.1109/IEEESTD.2020.8999849 (see page 164).

[243] Peter Wellig, Peter Speirs, Christof Schuepbach, Roland Oechslin, Matthias Renker, Urs Boeniger and Hans Pratisto. **Radar Systems and Challenges for C-UAV**. In: *2018 19th International Radar Symposium (IRS)*. 2018, 1–8. DOI: 10.23919/IRS.2018.8448071 (see page 165).

[244] Arik D. Brown. **Radar Challenges, Current Solutions, and Future Advancements for the Counter Unmanned Aerial Systems Mission**. *IEEE Aerospace and Electronic Systems Magazine* 38:9 (2023), 34–50. DOI: 10.1109/MAES.2023.3289928 (see page 165).

[245] Hao Kong, Cheng Huang, Jiadi Yu and Xuemin Shen. **A Survey of mmWave Radar-Based Sensing in Autonomous Vehicles, Smart Homes and Industry**. *IEEE Communications Surveys & Tutorials* (2024), 1–1. DOI: 10.1109/COMST.2024.3409556 (see page 165).

[246] Dave Tahmoush. **Review of micro-Doppler signatures**. *IET Radar, Sonar & Navigation* 9:9 (2015), 1140–1146 (see page 165).

[247] M. Hassanalian and A. Abdelkefi. **Classifications, applications, and design challenges of drones: A review**. *Progress in Aerospace Sciences* 91 (2017), 99–131. ISSN: 0376-0421. DOI: 10.1016/j.paerosci.2017.04.003 (see page 165).

[248] Wen Jiang, Yanping Wang, Yang Li, Yun Lin and Wenjie Shen. **Radar Target Characterization and Deep Learning in Radar Automatic Target Recognition: A Review**. *Remote Sensing* 15:15 (2023). ISSN: 2072-4292. DOI: 10.3390/rs15153742. URL: https://www.mdpi.com/2072-4292/15/15/3742 (see page 165).

[249] Jianwei Li, Zhentao Yu, Lu Yu, Pu Cheng, Jie Chen and Cheng Chi. **A Comprehensive Survey on SAR ATR in Deep-Learning Era**. *Remote Sensing* 15:5 (2023). ISSN: 2072-4292. DOI: 10.3390/rs15051454. URL: https://www.mdpi.com/2072-4292/15/5/1454 (see page 165).

[250] Junyu Li and Cheng Peng. **Weighted residual network for SAR automatic target recognition with data augmentation**. *Frontiers in Neurorobotics* 17 (2023). ISSN: 1662-5218. DOI: 10.3389/fnbot.2023.1298653. URL: https://www.frontiersin.org/journals/neurorobotics/articles/10.3389/fnbot.2023.1298653 (see page 165).

[251] Wenbo Weng. **Multitask Sparse Representation of Two-Dimensional Variational Mode Decomposition Components for SAR Target Recognition**. *Scientific Programming* 2023:1 (2023), 8846287. DOI: https://doi.org/10.1155/2023/8846287. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1155/2023/8846287. URL: https://onlinelibrary.wiley.com/doi/abs/10.1155/2023/8846287 (see page 165).

[252] Denys Kolesnikov, Semen Zhyla, Volodymyr Pavlikov and Oleksandr Mazurenko. **Imaging Simulation for Radar with Static Aperture Synthesis Method**. In: *2023 IEEE International Conference on Information and Telecommunication Technologies and Radio Electronics (UkrMiCo)*. 2023, 192–197. DOI: 10.1109/UkrMiCo61577.2023.10380361 (see page 166).

[253] Simeon Zhyla, Anatoliy Popov, Eduard Tserne, Gleb Cherepnin, Danyil Kovalchuk and Olha Inkarbaieva. **UAV-born Narrowband Radar Complex for Direction Finding of Radio Sources**. In: *2023 13th International Conference on Dependable Systems, Services and Technologies (DESSERT)*. 2023, 1–6. DOI: 10.1109/DESSERT61349.2023.10416434 (see page 166).

[254] Igor Bisio, Chiara Garibotto, Halar Haleem, Fabio Lavagetto and Andrea Sciarrone. **RF/WiFi-based UAV surveillance systems: A systematic literature review**. *Internet of Things* 26 (2024), 101201. ISSN: 2542-6605. DOI: https://doi.org/10.1016/j.iot.2024.101201. URL: https://www.sciencedirect.com/science/article/pii/S2542660524001422 (see page 166).

[255] Driss Aouladhadj, Ettien Kpre, Virginie Deniau, Aymane Kharchouf, Christophe Gransart and Christophe Gaquière. **Drone Detection and Tracking Using RF Identification Signals**. *Sensors* 23:17 (2023). ISSN: 1424-8220. DOI: 10.3390/s23177650. URL: https://www.mdpi.com/1424-8220/23/17/7650 (see page 166).

[256] Aerospace and Defence Industries Association of Europe Standardization (ASD-STAN). **Drone Remote ID – Introduction to the European UAS Digital Remote ID Technical Standard** (2021) (see page 166).

[257] Suna Choi and Kyu-min Kang. **Study on Remote Identification and Integrated Information Management Technology for Unmanned Aerial Vehicles**. In: *2023 32nd International Conference on Computer Communications and Networks (ICCCN)*. 2023, 01–02. DOI: 10.1109/ICCCN58024.2023.10230213 (see page 166).

[258] Jiangyi Deng, Xiaoyu Ji, Beibei Wang, Bin Wang and Wenyuan Xu. **Dr. Defender: Proactive Detection of Autopilot Drones Based on CSI**. *IEEE Transactions on Information Forensics and Security* 19 (2024), 194–206. DOI: 10.1109/TIFS.2023.3311964 (see page 167).

[259] Aleksi Vihelä, Matias Turunen, Mikko Heino and Taneli Riihonen. **Adaptive RF Self-interference Canceller Control and Spectrum Sensing in Full-Duplex EW Radio**. In: *2024 International Conference on Military Communication and Information Systems (ICMCIS)*. 2024, 1–7. DOI: 10.1109/ICMCIS61231.2024.10540876 (see page 167).

[260] Udara De Silva, Hiruni Silva and Arjuna Madanayake. **Augmented Envelope Neural Networks on RF-SoC for Digital Self-Interference Cancellation**. *IEEE Access* 12 (2024), 44091–44103. DOI: 10.1109/ACCESS.2024.3380819 (see page 167).

[261] Kalit Naresh Inani, K.S. Sangwan and Dhiraj. **Machine Learning based framework for Drone Detection and Identification using RF signals**. In: *2023 4th International Conference on Innovative Trends in Information Technology (ICITIIT)*. 2023, 1–8. DOI: 10.1109/ICITIIT57246.2023.10068637 (see page 167).

[262] Ayoub Benali Amjoud and Mustapha Amrouch. **Object Detection Using Deep Learning, CNNs and Vision Transformers: A Review**. *IEEE Access* 11 (2023), 35479–35516. DOI: 10.1109/ACCESS.2023.3266093 (see pages 167, 168).

[263] Yingrui Jin, Zhaoyuan Shi, Xinlong Xu, Guang Wu, Hengyi Li and Shengjun Wen. **Target Localization and Grasping of NAO Robot Based on YOLOv8 Network and Monocular Ranging**. *Electronics* 12:18 (2023). ISSN: 2079-9292. DOI: 10.3390/electronics12183981. URL: https://www.mdpi.com/2079-9292/12/18/3981 (see page 168).

[264] Ulzhalgas Seidaliyeva, Lyazzat Ilipbayeva, Kyrmyzy Taissariyeva, Nurzhigit Smailov and Eric T. Matson. **Advances and Challenges in Drone Detection and Classification Techniques: A State-of-the-Art Review**. *Sensors* 24:1 (2024). ISSN: 1424-8220. DOI: 10.3390/s24010125. URL: https://www.mdpi.com/1424-8220/24/1/125 (see pages 168, 169).

[265] Bert van den Broek, Jos van der Velde, Michiel van den Baar, Loek Nijsten and Rob van Heijster. **Automatic threat evaluation for border security and surveillance**. In: *Counterterrorism, Crime Fighting, Forensics, and Surveillance Technologies III*. Ed. by Henri Bouma, Radhakrishna Prabhu, Robert James Stokes and Yitzhak Yitzhaky. Vol. 11166. International Society for Optics and Photonics. SPIE, 2019, 111660G. DOI: 10.1117/12.2532308. URL: https://doi.org/10.1117/12.2532308 (see page 169).

[266] Xuan Wang, Zhaojie Sun, Abdellah Chehri, Gwanggil Jeon and Yongchao Song. **Deep learning and multi-modal fusion for real-time multi-object tracking: Algorithms, challenges, datasets, and comparative study**. *Information Fusion* 105 (2024), 102247. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2024.102247. URL: https://www.sciencedirect.com/science/article/pii/S1566253524000253 (see pages 169, 173, 174).

[267] ALADDIN Consortium. **D4.19 - Report on standardisation, regulation, and SOTA progress V7**. Tech. rep. European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 740859, 2020 (see page 169).

[268] Honggu Kang, Jingon Joung, Jinyoung Kim, Joonhyuk Kang and Yong Soo Cho. **Protect Your Sky: A Survey of Counter Unmanned Aerial Vehicle Systems**. *IEEE Access* 8 (2020), 168671–168710. DOI: 10.1109/ACCESS.2020.3023473 (see page 170).

[269] Stamatios Samaras, Vasileios Magoulianitis, Anastasios Dimou, Dimitrios Zarpalas and Petros Daras. **UAV Classification with Deep Learning Using Surveillance Radar Data**. In: *Computer Vision Systems*. Ed. by Dimitrios Tzovaras, Dimitrios Giakoumis, Markus Vincze and Antonis Argyros. Cham: Springer International Publishing, 2019, 744–753. ISBN: 978-3-030-34995-0 (see page 170).

[270] Tianchen Deng, Yi Zhou, Wenhua Wu, Mingrui Li, Jingwei Huang, Shuhong Liu, Yanzeng Song, Hao Zuo, Yanbo Wang, Yutao Yue, Hesheng Wang and Weidong Chen. *Multi-Modal UAV Detection, Classification and Tracking Algorithm – Technical Report for CVPR 2024 UG2 Challenge*. 2024. arXiv: 2405.16464 [cs.RO]. URL: https://arxiv.org/abs/2405.16464 (see page 170).

[271] X. Shi, C. Yang, W. Xie, C. Liang, Z. Shi and J. Chen. **Anti-Drone System with Multiple Surveillance Technologies: Architecture, Implementation, and Challenges**. *IEEE Communications Magazine* 56:4 (2018), 68–74. DOI: 10.1109/MCOM.2018.1700430 (see page 172).

[272] Wanying Zhang, Yan Liang, Yun Zhu and Yumei Zhang. **A variational Bayesian approach for formation target tracking**. *Aerospace Science and Technology* 146 (2024), 108965. ISSN: 1270-9638. DOI: 10.1016/j.ast.2024.108965 (see page 173).

[273] Emilio Calvanese Strinati, George C. Alexandropoulos, Madhusudan Giyyarpuram, Philippe Sehier, Sami Mekki, Vincenzo Sciancalepore, Maximilian Stark, Mohamed Sana, Benoit Denis, Maurizio Crozzoli, Navid Amani, Placido Mursia, Raffaele D Errico, Mauro Boldi, Francesca Costanzo, Francois Rivet and Henk Wymeerschx. *Distributed Intelligent Integrated Sensing and Communications: The 6G-DISAC Approach*. 2024. arXiv: 2402.18271 [eess.SP]. URL: https://arxiv.org/abs/2402.18271 (see page 173).

[274] Hazem Sallouha, Sharief Saleh, Sibren De Bast, Zhuangzhuang Cui, Sofie Pollin and Henk Wymeersch. **On the Ground and in the Sky: A Tutorial on Radio Localization in Ground-Air-Space Networks**. *IEEE Communications Surveys & Tutorials* (2024), 1–1. DOI: 10.1109/COMST.2024.3417336 (see page 173).

[275] Jingxin Wei, Feng Luo, Jiawei Qi and Xianxian Luo. **Distributed Fusion of Labeled Multi-Bernoulli Filters Based on Arithmetic Average**. *IEEE Signal Processing Letters* 31 (2024), 656–660. DOI: 10.1109/LSP.2024.3364506 (see page 173).

[276] Zhixing Wang, Gaofan Zhou, Jinzhen Yao, Jianlin Zhang, Qiliang Bao and Qintao Hu. **Self-Prompting Tracking: A Fast and Efficient Tracking Pipeline for UAV Videos**. *Remote Sensing* 16:5 (2024). ISSN: 2072-4292. DOI: 10.3390/rs16050748. URL: https://www.mdpi.com/2072-4292/16/5/748 (see page 173).

[277] Tiancheng Li, Haozhe Liang, Bing Xiao, Quan Pan and You He. **Finite mixture modeling in time series: A survey of Bayesian filters and fusion approaches**. *Information Fusion* 98 (2023), 101827. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2023.101827. URL: https://www.sciencedirect.com/science/article/pii/S1566253523001434 (see page 174).

[278] M. Repetto, A. Carrega and R. Rapuzzi. **An architecture to manage security operations for digital service chains**. *Future Generation Computer Systems* 115 (2021), 251–266. DOI: 10.1016/j.future.2020.08.044 (see pages 175, 176).

[279] F. Alt and E. Zezschwitz. **Emerging trends in usable security and privacy**. *Journal of Interactive Media* 18:3 (Dec. 2019), 189–195 (see page 176).

[280] A.P. Sage and C.D. Cuppan. **On the Systems Engineering and Management of Systems of Systems and Federations of Systems**. *Information, Knowledge, Systems Management* 2:4 (2001), 325–345 (see page 176).

[281] Matteo Repetto, Domenico Striccoli, Giuseppe Piro, Alessandro Carrega, Gennaro Boggia and Raffaele Bolla. **An autonomous cybersecurity framework for next-generation digital service chains**. *Journal of Network and Systems Management* 29:37 (May 2021). DOI: 10.1007/s10922-021-09607-7 (see page 176).

[282] D. Hardt. *The OAuth 2.0 Authorization Framework*. RFC 6749. Oct. 2012. URL: https://www.rfc-editor.org/rfc/rfc6749.txt (see page 176).

[283] L. Caviglione, W. Mazurczyk, M. Repetto, A. Schaffhauser and M. Zuppelli. **Kernel-level tracing for detecting stegomalware and covert channels in Linux environments**. *Computer Networks* 191 (May 2021). DOI: 10.1016/j.comnet.2021.108010 (see page 176).

[284] M Repetto, A. Carrega and A. Duzha. **A novel cyber-security framework leveraging programmable capabilities in digital services**. In: *Proc. Fourth Italian Conference on Cyber Security*. CEUR Workshop Proceedings. 2020, 201–211 (see page 176).

[285] M. Repetto, A. Carrega and G. Lamanna. **An architecture to manage security services for cloud applications**. In: *2019 4th International Conference on Computing, Communications and Security (ICCCS)*. 2019, 1–8. DOI: 10.1109/CCCS.2019.8888061 (see page 176).

[286] *Apache Kafka*. https://kafka.apache.org/ (see pages 177, 181, 182).

[287] V. Hu, D. Ferraiolo, R. Kuhn, A. Schnitzer, K. Sandlin, R. Miller and K. Scarfone. **Guide to Attribute Based Access Control (ABAC) Definition and Considerations**. Special Publication (NIST SP) 800-162. National Institute of Standards and Technology (NIST), 2019 (see page 178).

[288] N. Dan, S. Hua-Ji, C. Yuan and G. Jia-Hu. **Attribute Based Access Control (ABAC)-Based Cross-Domain Access Control in Service-Oriented Architecture (SOA)**. In: *2012 International Conference on Computer Science and Service System*. 2012, 1405–1408. DOI: 10.1109/CSSS.2012.354 (see page 178).

[289] D. Jaramillo, D. V. Nguyen and R. Smart. **Leveraging microservices architecture by using Docker technology**. In: *SoutheastCon 2016*. 2016, 1–5. DOI: 10.1109/SECON.2016.7506647 (see page 180).

[290] *WSO2 Identity Server Documentation*. https://is.docs.wso2.com/en/latest/ (see pages 181, 182).