UNIVERSIDADE FEDERAL DE MINAS GERAIS
Instituto de Ciências Exatas
Programa de Pós-Graduação em Ciência da Computação

Lailla Milainny Siqueira Bine

Design and Analysis of Routing Protocols for Internet of Drones

Belo Horizonte
2023

Lailla Milainny Siqueira Bine

Design and Analysis of Routing Protocols for Internet of Drones

Final Version

Dissertation presented to the Graduate Program in Computer Science of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

Advisor: Antonio Alfredo Ferreira Loureiro
Co-Advisor: Linnyer Beatrys Ruiz Aylon

Belo Horizonte
2023

2023, Lailla Milainny Siqueira Bine.
Todos os direitos reservados

Bine, Lailla Milainny Siqueira.

B612d Design and analysis of routing protocols for internet of drones [recurso eletrônico] / Lailla Milainny Siqueira Bine – 2023.
1 recurso online (246 f. il, color.) : pdf.

Orientador: Antônio Alfredo Ferreira Loureiro.
Coorientador: Linnyer Beatrys Ruiz Aylon.
Tese (Doutorado) - Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciências da Computação.
Referências: f. 220-246

1. Computação – Teses. 2. Aeronave não tripulada – Teses. 3. Roteamento (Administração de redes de computadores) – Protocolos - Teses. I. Loureiro, Antônio Alfredo Ferreira. II. Aylon, Linnyer Beatrys Ruiz. III. Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Computação. IV. Título.

CDU 519.6*82.9(043)

Ficha catalográfica elaborada pela bibliotecária Irenquer Vismeg Lucas Cruz
CRB 6/819 - Universidade Federal de Minas Gerais - ICEx

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Design and Analysis of Routing Protocols for Internet of Drones

LAILLA MILAINNY SIQUEIRA BINE

Tese defendida e aprovada pela banca examinadora constituída pelos Senhores(a):

Prof. Antonio Alfredo Ferreira Loureiro - Orientador
Departamento de Ciência da Computação - UFMG

Profa. Linnyer Beatrys Ruiz Aylon - Coorientadora
Departamento de Informática - UEM

Luiz Filipe Menezes Vieira
Prof. Luiz Filipe Menezes Vieira
Departamento de Ciência da Computação - UFMG

Eduardo Cerqueira
Prof. Eduardo Coelho Cerqueira
Instituto de Tecnologia - UFPA

Raquel Mini
Profa. Raquel Aparecida de Freitas Mini
Ericsson Research

Prof. Heitor Soares Ramos Filho
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 24 de outubro de 2023.

To Marcianita, Antonio, Wuigor, Alisson and Bianca.

Acknowledgments

I thank God for blessing me on this journey and the people who contributed to this work. In particular:

To my advisors, Prof. Azzedine Boukerche (uOttawa), Canada, and Prof. Antonio A. F. Loureiro (UFMG), Brazil, for their support and advice through my graduate studies. I would like to express my sincere gratitude to you both for giving me such a golden opportunity to study abroad and for the support of each step of this process during the COVID-19 pandemic. Thank you so much for the encouragement and guidance not only related to research but to life.

To Prof. Linnyer Beatrys Ruiz Aylon for encouraging me to continue in academic life and for all the suggestions and advice, not only during the period of this work but throughout my graduate studies, thank you very much.

To my parents, Antonio and Marcianita, for always teaching me to follow the right direction. Dad, thank you for being an example of a student and inspiring me to be too. Mom, thank you so much for being an example of a strong and independent woman.

To Alisson, for the countless suggestions and advice. The result of this work would not be the same without your participation. Thank you for the encouragement, for always believing in me, and for being by my side at all times. I love you so much!

To Wuigor and Bianca, for the patience and concern to make my days better. I would also like to extend my thanks to my friends and colleagues in the MANNA Lab (Brazil), WISEMAP Lab (Brazil), and PARADISE Research Lab (Canada) for all kinds of support that I received from them. I would like to give my special thanks to Katharinne and Roniel for welcoming Alisson and me to Canada.

This research was partially supported by the NSERC TRANSIT Project and NSERC DIVA Strategic Network Research Program and the Coordination for the Improvement of Higher Education Personnel (CAPES).

“*We’ve always defined ourselves by the ability to overcome the impossible. And we count these moments. These moments when we dare to aim higher, to break barriers, to reach for the stars, to make the unknown known. We count these moments as our proudest achievements. But we lost all that. Or perhaps we’ve just forgotten that we are still pioneers. And we’ve barely begun. And that our greatest accomplishments cannot be behind us, that our destiny lies above us.*”

(Interstellar)

Resumo

A Internet dos Drones (IoD) é uma tecnologia emergente que permitirá uma nova era de serviços e aplicações de drones. No entanto, muitas barreiras e desafios permanecem até que possamos controlar uma rede IoD complexa. A comunidade científica ainda está discutindo, estudando e investigando a melhor forma de implementar essa rede para tornar o IoD viável, confiável e eficiente. Além disso, os princípios que norteiam as redes sem fio terrestres e mesmo as redes tradicionais de VANTs (Veículos Aéreos Não Tripulados) não se aplicam ao IoD principalmente porque permite que drones distintos realizando diferentes aplicações compartilhem o espaço aéreo.

Primeiramente, esta tese discute, explora e estuda os desafios da IoD através da Computação Urbana. Essa análise pode trazer insights sobre a organização do espaço aéreo e os protocolos de comunicação IoD, porque revelam a dimensão, as necessidades e os desafios desse ambiente. Em seguida, propomos e estudamos uma arquitetura topológica do espaço aéreo e das vias aéreas para construir um ambiente de acesso coordenado para drones. Discutimos a aplicação de entrega por drone no contexto do IoD e apresentamos diretrizes de decisão sobre quando usar a entrega por drone e quando optar por entregas tradicionais. Apresentamos um método para gerar planejamento de trajetória para cobrir o espaço aéreo organizado em vias aéreas.

Em seguida, estudamos os protocolos de roteamento para IoD considerando à topologia estabelecida. Propomos um protocolo de roteamento geocast para IoD para disseminar mensagens de alerta em situações imprevisíveis que possam ocorrer na rede. Além disso, propomos um protocolo IoD usando a abordagem IoD definida por software para controlar e permitir maior programabilidade e adaptabilidade da rede. Adicionalmente, exploramos a colaboração de IoD com outras redes por meio de dois protocolos híbridos. Primeiramente, propomos um protocolo para unir redes IoD e redes de ônibus. As redes de ônibus apresentam alta previsibilidade, permitindo que os drones aproveitem os ônibus para preencher lacunas em sua comunicação quando a rede IoD é escassa. Por fim, propomos um protocolo para um cenário futurístico em que redes veiculares, IoD e Internet of Flying Cars (IoFC) colaboram em situações especificas como, por exemplo a disseminação de mensagens de emergência.

As soluções desenvolvidas foram comparadas com trabalhos relacionados através de simulações em um simulador que permite a criação e gerenciamento de vias aéreas. Os resultados são promissores e mostram o potencial de métodos de planejamento de caminhos e protocolos de comunicação desenvolvidos em diferentes cenários. Finalmente, apresentamos as próximas etapas a serem desenvolvidas. Em resumo, esta tese fornece procedimentos e discussões que podem orientar o desenvolvimento futuro para superar barreiras relacionadas a problemas fundamentais em IoD, como comunicação e mobilidade.

Palavras-chave: internet dos drones; protocolos de roteamento; drones.

Abstract

The Internet of Drones (IoD) is an emerging technology that will enable a new era of drone services and applications. However, many barriers and challenges remain until we can control a complex IoD network. The scientific community is still discussing, studying, and investigating the best way to implement this network to become the IoD viable, reliable, and efficient. Furthermore, the principles that guide terrestrial wireless networks and even traditional Unmanned Aerial Vehicle (UAV) networks do not apply to IoD mainly because it allows distinct drones performing different applications to share the airspace.

First, this thesis discusses, explores, and studies the challenges of IoD through Urban Computing. This analysis can bring insights into the design of the IoD airspace organization and communication protocols because they reveal the IoD environment dimension, needs, and challenges. Then, we propose and study a topological architecture of airspace and airways to build a coordinated drone access environment. We discuss the drone delivery application in the context of IoD and present decision guidelines on when to use drone delivery and when to opt for traditional deliveries. Also, we present a method for generating path planning to cover airspace organized in airways.

Next, we study the routing protocols for IoD regarding the established topology. We propose a geocast routing protocol for IoD to disseminate alert messages in unpredictable situations that may occur in the network. Furthermore, we propose an IoD protocol using the Software-defined IoD approach to control and allow greater programmability and adaptability of the network. In addition, we explore IoD collaboration with other networks through two hybrid protocols. First, we propose a protocol to join IoD and bus networks. Bus networks feature predictability allowing drones to take advantage of buses to fill gaps in their communication when the network is sparse. Finally, we propose a protocol for a futuristic scenario in which vehicular networks, IoD and Internet of Flying Cars (IoFC) collaborate in specific situations such as the dissemination of emergency messages.

The developed solutions have been compared with related works through simulations in a simulator that allows the creation and management of airways. The results are promising and show the potential of path planning methods and communication protocols developed in different scenarios. Finally, we present the next steps to be developed. In summary, this thesis provides procedures and discussions that can guide future development to overcoming barriers related to fundamental problems in IoD, such as communication and mobility.

Keywords: internet of drones; routing protocols; drones.

List of Figures

List of Tables

List of Publications

Journal Papers

1. Bine, L. M. S., Boukerche, A., Ruiz, L. B., and Loureiro, A. A. F. “Connecting Internet of Drones and Urban Computing: Definitions, Applications, and Challenges”. *Computer Networks*, p. 110136, 2023. doi: `https://doi.org/10.1016/j.comnet.2023.110136`.

2. Bine, L. M. S., Boukerche, A., Ruiz, L. B., and Loureiro, A. A. F. “IoDGR: A Geocast Routing Protocol for Sparse Internet of Drones”. *Computer Networks*, Under review.

3. Bine, L. M. S., Svaigen, A. R., Boukerche, A., Ruiz, L. B., and Loureiro, A. A. F. “Flavors of the Next Generation of Unmanned Aerial Vehicles Networks”. *IoT Journal*, Second Round.

4. Bine, L. M. S., Boukerche, A., Ruiz, L. B., and Loureiro, A. A. F. “Internet of Drones and Terrestrial Networks: A Successful Partnership”. *IEEE Internet of Things Magazine*, v. 6, n. 4, p. 104-110, 2023. doi: `https://doi.org/10.1109/IOTM.001.2200265`.

5. Bine, L. M. S., Boukerche, A., Ruiz, L. B., and Loureiro, A. A. F. “A Novel Ant Colony-inspired Coverage Path Planning for Internet of Drones”. *Computer Networks*, v. 235, p. 109963, 2023. doi: `https://doi.org/10.1016/j.comnet.2023.109963`.

6. Bine, L. M. S., Boukerche, A., Ruiz, L. B., and Loureiro, A. A. F. “IoDMix: A novel routing protocol for Delay-Tolerant Internet of Drones integration in Intelligent Transportation System”. *Ad Hoc Networks*, v. 148, p. 103204, 2023. doi: `https://doi.org/10.1016j.adhoc.2023.103204`.

7. Bine, L. M., Boukerche, A., Ruiz, L. B., and Loureiro, A. A. “Leveraging Urban Computing with the Internet of Drones”. *IEEE Internet of Things Magazine*, v. 5, n. 1, p. 160-165, 2022. doi: `https://doi.org/10.1109/IOTM.003.2100091`.

Conference Papers

1. Bine, L. M. S., Boukerche, A., Ruiz, L. B., and Loureiro, A. A. F. “Drone Delivery: Why, Where, and When”. *Proceedings of the ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks*, 2023. doi: 10.1145/3616394.3618265.

2. Bine, L. M. S., Boukerche, A., Ruiz, L. B., and Loureiro, A. A. F. “ARAT: An altitude-based routing protocol for Hybrid Aerial-terrestrial Networks”. *Proceedings of the IEEE International Conference on Communications (ICC)*, 2023. doi: 10.1109/ICC45041.2023.10279654.

3. Bine, L. M. S., Boukerche, A., Ruiz, L. B., and Loureiro, A. A. F. “Coverage Path Planning for Internet of Drones”. *Proceedings of the ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks*, pp. 49-57, 2022. doi: 10.1145/3551663.3558675.

4. Bine, L. M. S., Boukerche, A., Ruiz, L. B., and Loureiro, A. A. F. “Um Protocolo de Roteamento Store-Carry-Forward para Unir Redes de Ônibus e Internet dos Drones”. (In Portuguese) *Proceedings of the 40th Brazilian Symposium on Computer Networks and Distributed Systems (SBRC)*, 2022. doi: 10.5753/sbrc.2022.222340.

5. Bine, L. M. S., Boukerche, A., Ruiz, L. B., and Loureiro, A. A. F. “IoDSCF: A Store-Carry-Forward Routing Protocol for joint Bus Networks and Internet of Drones”. *Proceedings of the 42nd IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2022. doi: 10.1109/ICDCS54860.2022.00096.

6. Bine, L. M. S., Boukerche, A., Ruiz, L. B., and Loureiro, A. A. F. “PSDIoD: A Position-based Routing Protocol for Software-Defined Internet of Drones”. *Proceedings of the IEEE International Conference on Communications (ICC)*, 2022. doi: 10.1109/ICC45855.2022.9839173.

7. Bine, L. M. S., Boukerche, A., Ruiz, L. B., and Loureiro, A. A. F. “IoDAGR: An Airway-based Geocast Routing Protocol for Internet of Drones”. *Proceedings of the IEEE International Conference on Communications (ICC)*, pp. 1 - 6, 2021. doi: 10.1109/ICC42927.2021.9500444.

List of Abbreviations

ACO Ant Colony Optimization.

AntIoD Ant Colony Optimization Algorithm for CPP in IoD.

AODV Ad-hoc On-demand Distance Vector Routing.

AP Access Points.

ARAT Altitude-based Routing Protocol for Hybrid Aerial-terrestrial Networks.

BN Bus Network.

BSN Bicycle Sharing Networks.

CLMN Column Mobility Model.

CN Connection Range.

CO Communication Range.

CPP Coverage Path Planning.

D2FC Drone-to-Flight Car.

D2V Drone-to-Vehicle.

DEM Digital Elevation Model.

DG-Castor Direction-based GeoCast Routing Protocol.

DRG Distributed Robust Geocast.

DSM Digital Surface Model.

DTIoD Delay-Tolerant Internet of Drones.

DTM Digital Terrain Model.

DTSG Dynamic Time-Stable Geocast Routing.

FAA United States Federation Aviation Administration.

FANET Flying Ad hoc Network.

FC2RD	Flight Car-to-Robust Drone.
Geo-SID	Geocast-based Safety Information Dissemination.
GeoAKOM	Smart Geocasting Protocol for Vehicular Networks.
Geo-SDVN	Geocast Protocol for Software Defined Vehicular Networks.
GeoUAV	A Geocast Routing Protocol for Fleet of UAVs.
GLS	Grid Location Service.
GM	Gauss-Markov.
GPS	Global Positioning System.
HAP	High-altitude Platforms.
HAPS	High Altitude Platform Stations.
HATN	Hybrid Aerial-terrestrial Network.
HLS	Hierarchical Location Service.
IoD	Internet of Drones.
IoDAGR	an Airway-based Geocast Routing Protocol for the Internet of Drones.
IoDGR	Geocast Routing Protocol for the Internet of Drones.
IoDMix	Delay-tolerant Internet of Drones Protocol in a Multi-vehicle Scenario.
IoDSCF	Store-carry-forward Protocol to Joint Bus Networks and IoD.
IoDSIM	IoD Simulator.
IoDSP	Internet of Drone Service Provider.
IoFC	Internet of Flying Cars.
IoFT	Internet of Flying Things.
IoT	Internet of Things.
IoTF	Internet of Flying Things.
IoV	Internet of Vehicles.
ITS	Intelligent Transport System.

IVG Inter-Vehicular Geocast protocol.

LBSN Location-Based Social Networking.

LoS Line of Sight.

MANET Mobile Ad hoc Network.

MCTP Mobile Control Transport Protocol.

MEC Mobile Edge Computing.

MG Manhattan Grid.

PDR Packet Delivery Ratio.

PPRZM Paparazzi Mobility Model.

PRS Pursue Model.

PSDIoD Position-based Routing Protocol for Software-Defined IoD.

PTN Public Transportation Network.

RDP Robust Drones Platforms Networks.

RLS Reactive Location Service.

ROVER Reliable Geographical Multicast Routing.

RPC Radio Power Consumption.

RRPR Reliable Predictive Routing.

SBD Schedule by Distance.

SDDN Software-Defined Drone Network.

SD-UAVNet (Software-Defined UAV Network.

SDIoD Software-Defined Internet of Drones.

SDN Software Defined Network.

SDVN (Software Defined Vehicular Network.

SRCM Semi-Random Circular Movement.

ST Smooth Turn.

TD Task Drones.

TZ Transmission Zone.

UAS Unmanned Aircraft System.

UAV Unmanned Aerial Vehicle.

UC Urban Computing.

URLLC Ultra-Reliable and Low-Latency Communication.

V2D Vehicle-to-Drone.

V2V Vehicle-to-Vehicle Communication.

VANET Vehicular Ad hoc Network.

VNT Vehicular Transport Protocol.

ZOA Zone of Approaching.

ZOF Zone of Forwarding.

ZOR Zone of Relevance.

ZSP Zone Service Provider.

Contents

Chapter 1

Introduction

The rapid advance of ubiquitous wireless connectivity and sensor miniaturization has provided new scenarios for applying Unmanned Aerial Vehicle (UAV), also known as drones. In the past, drones were mainly used in military applications, like surveillance, reconnaissance, and strike [1]. However, with the popularization of UAVs, public interest has been growing for other purposes, such as rescue operations, agriculture, communication resilience in disasters, public safety, transportation management, monitoring, emergency assistance, and possibly the most promising among them: drone delivery service [1, 2, 3, 4].

Considering that all applications need to use the airspace simultaneously, Gharibi et al. [5] proposed the Internet of Drones (IoD), an architecture to organize access to the airspace. Moreover, the IoD is part of Internet of Things (IoT), which means that the IoD can collect and transmit data. Recently, the United States Federation Aviation Administration (FAA) [6] showed interest in remotely tracking the drone's location. This may indicate that the UAVs' capacity to connect to the Internet will have an increasingly vital role in enhancing the safety and security of people and can be the future to implement UAV applications in urban areas.

An essential step to be solved to make the IoD possible is the communication between the network components. In IoD, it is essential to communicate with drones whenever they perform tasks that need some sort of coordination, management, and synchronization, for instance. The specific characteristics of UAV networks, such as 3D mobility, the high speed of nodes, and the irregular UAV distribution, must be considered in the design of a routing protocol in the area [7]. IoD differs from other UAV networks by enabling the drone organization in airways. In addition, it needs to manage different applications that may vary at different levels, ranging from drone hardware to network topologies. And all these drones must be able to share the same airspace in a coordinated and fair way. Considering this new world IoD has a gap in fundamental network problems like communication and mobility that need to be explored and overcome.

Furthermore, the IoD utilization generated data gathered across various applications, including urban sensing, rescue operations, agriculture, communication resilience during disasters, public safety, transportation management, monitoring, emergency assistance, and drone delivery services, holds the potential to enhance Urban Computing

(UC) applications. Urban Computing benefits from a large amount of data generated in urban centers to improve people's lives [8]. IoD is a new way of creating new and complementary data to existing ones and providing services through drones. However, prior to commencing data collection with IoD, it is imperative to design routing protocols that are well-suited to this extensive-scale scenario. IoD is still at the beginning of its development and has many steps and challenges to be overcome. Thus, there is a great research opportunity in several segments related to IoD and the connection between IoD-UC.

1.1 Thesis Statement

Thesis Problem: The deployed of IoD faces numerous challenges. There are substantial gaps in the existing literature concerning fundamental aspects of IoD, including both intra and inter-communication, as well as mobility within an environment where multiple applications operate concurrently. Intra-communication pertains to interactions between drones, while inter-communication involves communication between IoD and other networks. The unique characteristics of this scenario, encompassing dynamic network topology, airspace utilization, uniform UAV distribution, rapid node speeds, and limited energy resources, necessitate the development of new protocols and path planning methods to address these gaps effectively. Moreover, drones possess significant potential as data sources for enhancing Urban Computing (UC) applications. Therefore, our primary objective is to resolve communication and mobility challenges to actualize the use of drone data within the realm of UC.

Proposal: In this thesis, we aim to design and propose routing protocols and path planning methods for IoD within diverse scenarios. This endeavor is aimed at addressing communication and mobility challenges associated with IoD to ultimately enrich UC through the integration of IoD data.

1.2 Objectives

The main objectives of this thesis are to propose topologies and frameworks for organizing the airspace, develop protocols for efficient data dissemination Intra-IoD and Inter-IoD, and collect data to enhance UC applications. Specifically, we aim for the design of protocols to focus on sparse and emergency scenarios. In addition, we seek to develop the protocols so that they can collaborate with the UC.

The design of protocols for IoD must consider the aspects that make this environment unique such as the airways, demanding collaboration between drones, energy constraints, fair use of airspace, and collaboration with UC. Furthermore, the IoD protocol development should consider factors like collision avoidance and traffic and congestion management in a 3D scenario. The well-known and well-investigated wireless ad-hoc network issues have different dimensional and complex aspects in the IoD scenario. Therefore, these elements compose a singular environment in IoD that needs to be explored.

In general, we seek to explore the distinct aspects of IoD and use them as advantages for communication protocol design. Additionally, we intend to understand how IoD can collaborate with other networks and vice versa. Many aspects mentioned also influence the drones' mobility and path planning. Therefore, we plan to identify and propose solutions to two fundamental problems: communication and mobility. We intend to explore and advance the state of the art in Intra-network and Inter-network communication protocols. We intend to overcome barriers related to path planning in airways in a multi-drone and multi-application environment. Also, we intended to explore the relation of IoD-UC considering the communication and mobility aspects.

1.3 Contributions

Currently, UAV protocols have not been designed considering the IoD. To overcome this gap, we developed protocols for different scenarios as shown in Table 1.1. The protocols are mainly distinguished by the scenario and context in which they are inserted. The Geocast Routing Protocol for the Internet of Drones (IoDGR) and Position-based Routing Protocol for Software-Defined IoD (PSDIoD) protocols are Intra-network protocols while Delay-tolerant Internet of Drones Protocol in a Multi-vehicle Scenario (IoDMix) and Altitude-based Routing Protocol for Hybrid Aerial-terrestrial Networks (ARAT) are Inter-network protocols.

In the following, we list the contributions made in this thesis.

- **Two routing protocols for IoD:** This contribution proposes two Intra-IoD protocols. The first, IoDGR, seeks to solve the problem of geocast message dissemination in emergence scenarios (see Figure 1.1 - IoDGR). Specifically, we use the airways topology and the path planning of drones to improve the rate of delivery of packages. The second PSDIoD considers a scenario where Zone Service Provider (ZSP) can

Table 1.1: Protocols developed comparison

Protocol	**Context**	**Description**	**Application Scenario**
IoDGR	Intra-IoD	An airway-aware geocast protocol	Sparse network and emergence situation
PSDIoD	Intra-IoD	Position-based protocol	ZSPs can direct communicate with Drones forming an IoD Defined Network
IoDMix	Inter-IoD	A store-carry-forward protocol	Collaboration between IoD, Public Transportation Network (PTN), Bicycle Sharing Networks (BSN) and Vehicular Ad hoc Network (VANET)
ARAT	Inter-IoD	An altitude-based routing protocol	Collaboration between IoD and other aerial networks in an emergence situation

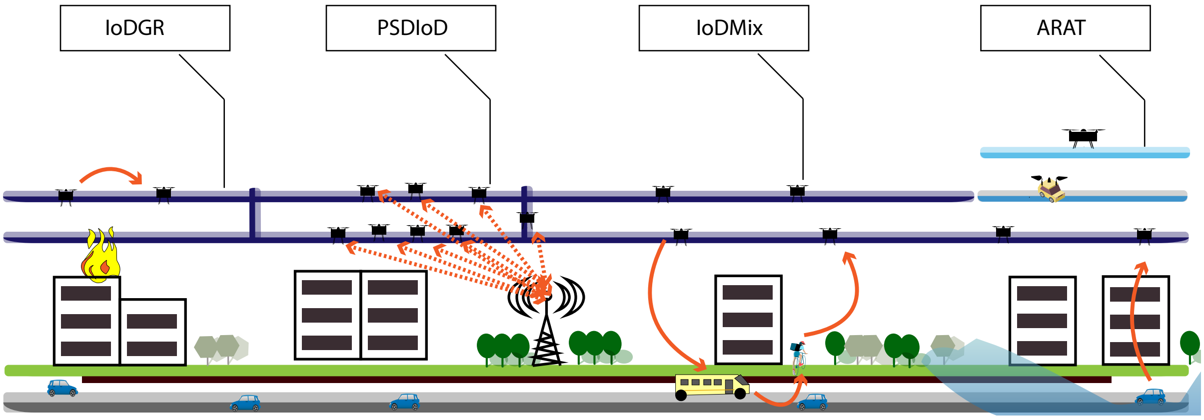
communicate directly with drones (see Figure 1.1 - PSDIoD). Thus, the protocol takes advantage of this factor to disseminate messages between drones;

- **Two routing protocols for joint IoD and Other Networks:** This contribution proposes two Inter-IoD protocols. The first, IoDMix, performs cooperation between drones, PTN, BSN and VANET to fill the communication gaps in scenarios where the IoD network is not connected (see Figure 1.1 - IoDMix). The second ARAT considers a cooperation scenario between Vehicular Ad hoc Network (VANET) and Aerial networks. Specifically, an emergency occurs and the only way to deliver a message to the ZSP is to send it to a robust node in the aerial network (see Figure 1.1 - ARAT);

- **Airways Path Planning for IoD :** This contribution proposes an airspace and airway topological architecture in IoD. Airspace and drone organization influences the development of network protocols. Therefore, we propose the airways organization in urban spaces parallel to the landways to delimit where the drones can pass. Also, we propose the airspace organization in layers so that other aerial networks can share the airspace with IoD;

- **IoD guidelines for deciding when to use drone delivery and when to use traditional deliveries:** This contribution conducts an analysis of drone delivery within the IoD context, comparing it with conventional deliveries. To be specific, we simulate real traditional deliveries within the IoD scenario to gain insights into the circumstances where each delivery method is more advantageous, depending on the specific environmental conditions.

Figure 1.1: Protocols Developed Overview



Source: Elaborated by the author

1.4 Thesis Organization

This thesis is organized as follows.

- **Chapter 2**: shows the IoD concepts. Specifically, we contextualize the IoD proprieties, the relation between IoD and other networks, and the challenges of design protocols for IoD;
- **Chapter 3**: brings an IoD analysis through UC concepts and applications. First, we discuss how IoD and UC are connected. After, we propose a framework for UC in the IoD context. We also present IoD-UC applications and highlight the challenges and important issues in the field;
- **Chapter 4**: introduces airspace and airway topological architecture to organize the airspace for allowing multiple drones and applications at the same time. We devise an analysis of drone delivery within the IoD context, comparing it with conventional deliveries, and propose guidelines based on the insights. Also, we design a Coverage Path Planning for the IoD scenario (CPP-IoD). We present IoD guidelines for deciding when to use drone delivery and when to use traditional deliveries;
- **Chapter 5**: focus on the IoD routing protocols development. We present two protocols the first one considers a geocast scenario. The second one focuses on a Software Defined Network approach. Specifically, we introduce the Software-defined Internet of Drones concept and present a protocol for this environment;
- **Chapter 6**: present the design of two hybrid protocols that involve IoD and other networks. The first one is a store-carry-forward protocol that creates a collaboration

between IoD, VANET, PTN and BSN. The second one considers a futurist scenario where IoD collaborates with UAVs and other aerial networks;

- **Chapter 7**: highlights the results of this thesis and its future work.

Chapter 2

Internet of Drones

In this Chapter, we present the basic concepts related to IoD. In Section 2.4, we detail IoD properties regarding topology and mobility. Section 2.6 discuss the partnership between IoD and Other Networks, considering VANETs, PTNs, and IoT. In Section 2.5 we present the import issues related to protocols design for IoD. Specifically, we discuss the physical, MAC, Network, Transport, and Application layers. Finally, a final discussion is given in Section 2.7.

2.1 Introduction

Gharibi et al. [5] conceptualized the term IoD as an architecture allowing coordinated air space for UAVs. However, IoD is more comprehensive than Garibit et al.'s definition. Two factors are fundamental in characterizing if a network is integrated into the Internet of Drones. First of all, at least one node needs to be a drone. Secondly, the network needs to have access to the Internet. This connection can exist in many forms: the UAV can have a connection with the Internet; the base station can have a connection with the Internet; or one node of the network (drone or base station) can have a connection with other devices that have access to the Internet.

Other devices may be part of the IoD, but the typical nodes that integrate this network are:

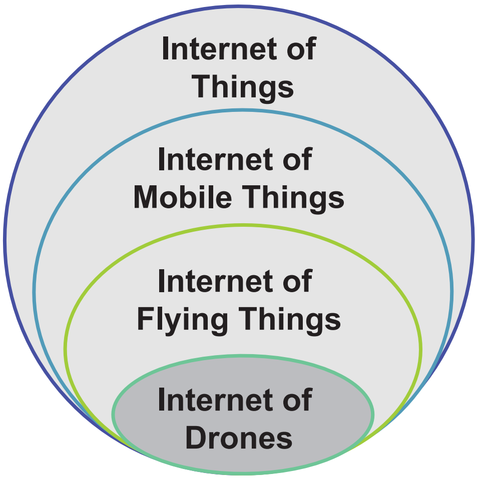
- **Drone or UAV**: The main object in IoD is a flight device with no onboard pilot [9, 10]. Drones come in various sizes and configurations. Usually, in single missions are used the large ones while in swarms are used the small ones [1].
- **Base Station**: The node responsible for controlling or/and aiding the drone in a mission. Also, this node can receive data collected for further analysis. In some cases, such as Flying Ad hoc Network (FANET), a drone can play the role of a base station [1].

Since drones are an emerging technology, different terms have been used in UAV studies. Even the term drone has different interpretations. In its original signified, a drone refers to an Unmanned Aerial Vehicle (UAV). However, the term drone has recently evolved into a more generic term, meaning any unmanned robot [11]. Despite this, we refer to drones only as air vehicles in this work. Accordingly, studies have used the terms drone and UAV to refer to IoD mobile air nodes [1, 12, 13].

Also, it is possible to find the term Unmanned Aircraft System (UAS) to refer to drones [14]. Fahlstrom and Gleason [14] considered that a UAS usually comprises more elements than an air vehicle, such as one or more base stations and a data link. In other words, the system consists of all the elements necessary to carry out a mission [15]. Thus, UAS is the same as the IoD if some node has Internet access.

With the popularization of UAVs, there was a surge of applications in different areas, including surveillance, public safety, transportation management, search and rescue, monitoring, and emergency assistance [1, 2, 3, 4]. With so many application possibilities came the need to connect drones to a network [16]. Thus, terms such as "Internet of Flying Things" (IoTF) [15, 17] and "Internet of Drones" (IoD) [5] emerged. Pigatto et al. [15] define IoTF as integrating flying things in the Internet of Things (IoT) paradigm. There is a tendency to connect all objects on the Internet, forming an IoT ecosystem. As shown in Figure 2.1, IoTF is a subarea of this ecosystem in which UAVs belong.

Figure 2.1: IoD and IoT related terms



Source: Elaborated by the author

IoTF encompasses any element able to fly, including manned flying objects. UAV-IoT [4, 18] is another terminology for integrating UAV with IoT. In this case, only unmanned aerial vehicles are considered. IoD [5] can be seen as an architecture responsible for coordinating the access to the air space for UAVs. However, as mentioned, this term is broader and represents any UAV network that can access the Internet, thus being a subarea of IoT, as depicted in Figure 2.1.

2.2 The Name of the Game

This section considers two main aspects. Firstly, we discuss the terminologies associated with various UAV networks found in the literature. Subsequently, we explore the fundamental elements constituting UAV networks using a notation. This analysis involves an investigation of the similarities and distinctions between these networks.

2.2.1 A Systematic Terminology of UAV Networks

The potential of drone networks to perform various applications has received interest from academic and corporate circles over the years. Such networks can feature diverse architectures, organizations, and components, leading to the use of multiple terminologies to refer to them [1, 5, 14, 19, 20, 21, 22]. Figure 2.2 depicts the terminologies considered in this work. Notice that different UAV networks can be present in various areas such as rural, industrial, urban, and port. We detail each term in the sequence.

However, the studies in the literature consider some similar aspects when discussing the characteristics of a given UAV network, converging in five main terminologies. Table 2.1 summarizes the main characteristics of each terminology. In a nutshell, UAV networks commonly vary regarding their domain, the number of UAVs as aerial nodes, the need for path planning that a UAV has to perform its task, in what altitude UAVs are allowed to fly, the scenarios that the network can be applied, and the third-party networks that the related terminology can be integrated with. We discuss each term thoroughly, including its aspects, as follows.

- **Internet of Flying Things (IoFT)** [23, 24, 25, 26] extends the concept of IoT by establishing dedicated Internet access and collaborating with different types of flying vehicles, such as manned and unmanned ones. Therefore, UAVs play a vital role in this concept. Zaidi et al. [23] discuss that IoFT follows a flying-cloud computing layered architecture where UAVs have a front-edge layer to communicate with the cloud layer through a communication layer. This communication layer can be deployed as a fog-computing layer that can communicate to the Cloud via the Internet or with the UAVs using a wireless connection. In IoFT, UAVs commonly act as sensor nodes, gathering and sending data through the Internet [24]. For instance, IoFT allows drones and other aerial devices to collect data from surface buoys in Underwater Wireless Sensor Networks (UWSNs) guided by satellites and

Table 2.1: Summarization of the main characteristics of each discussed terminology

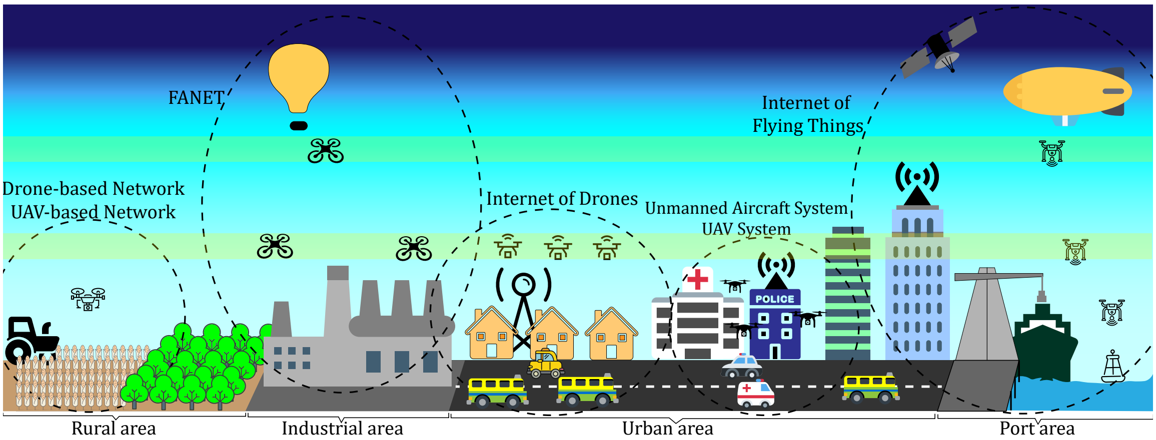
	Terminologies				
Design criteria	**Internet of Flying Things (IoFT)**	**Flying Ad hoc Network (FANET)**	**Drone/UAV-based network**	**Internet of Drones (IoD)**	**Unmanned Aircraft/Aerial Vehicle System**
Domain	Broadest (all flying things)	Temporary networks	Single-user operations	large-scale networks with UAVs as aerial nodes	Specific missions
Number of UAVs	Large compared to other nodes	Large/medium compared to other nodes	Varies according to the application	Large	Small
Path planning	Depends on the application	Dynamic, based on the application needs	Depends on the remote controller, commonly	Coordinated by the infrastructure for multiple drones	Pre-planned path for specific mission
Flying Altitude	Varies greatly depending on the flying nodes	Low to medium	Low	Low to medium	Low to medium
Typical Scenario	All environments	Temporary networks in environments with urgent needs	Single-purpose applications	All environments but focusing on urban centers	Single-purpose applications assisted by multiple UAVs
Network Integration	Satellites, HAPs, Terrestrial and Oceanic-based networks	Constrained by the applications needs	Based on a ground control network	Any third-party network connected to the Internet	Based on a ground system

High Altitude Platform Stations (HAPs) [25].

- **UAV/Unmanned Aircraft System** refers to a self-organized architecture model divided into two main layers: the aerial and the ground layers. UAVs compose a set of aerial nodes that exclusively collaborate with different terrestrial network devices [27, 28]. Furthermore, the connection with the Internet is not mandatory in networks covered by this terminology, differing from IoFT. UAV/Unmanned Aircraft Systems have a plethora of applications. For instance, an urban surveillance service primarily relies on drones while being supported and managed by grounded base stations without sharing the gathered data with the Internet through the Cloud [29].

- **Drone-based/UAV-based networks** are organized similarly to UAV/Unmanned Aircraft Systems. However, the nodes consist exclusively of UAVs in such a way a remote part controls them, or their trajectory is predefined [27, 30]. These networks are deployed for services operated by UAVs solely, such as aerial cinematography [31]. The deployment of a Drone-based/UAV-based network demands precise orchestration among the UAVs' mobility and well-defined trajectory [30], which are essential characteristics of these networks. Furthermore, the drone's resource allocation must be provisioned following the current weather conditions and environmental communications interference. Considering the previous terminologies, a drone-based/UAV-based network is part of the UAV/Unmanned Aircraft System [14].

Figure 2.2: Examples of application scenarios for each terminology



Source: Elaborated by the author

- **Flying Ad hoc Network (FANET)** exhibits an Ad Hoc structure, obligatorily, having different aerial nodes, such as weather balloons, drones, or flying vehicles [32, 33]. However, the aerial nodes are not connected to the Internet, which differs this type of network from an IoFT architecture. Due to the large number of applications that can operate following the FANET concept (e.g., Searching & Rescue in remote areas) [34], different routing protocols [32] and mobility models [33] have been designed for this network. These mobility models directly affect the modeling of the communication protocols since they impact how the UAVs (and other aerial nodes) will move to perform a given task [33].

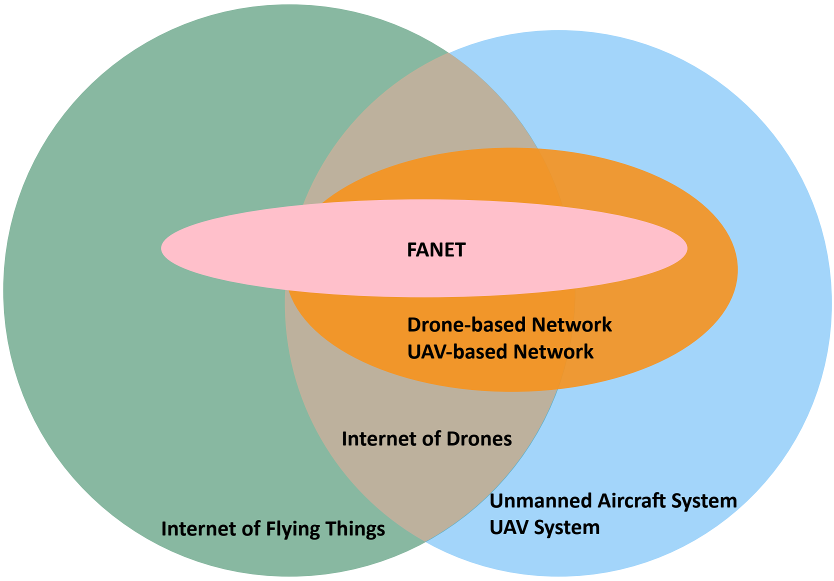
- **Internet of Drones (IoD)** as the nomenclature suggests, is an architecture composed exclusively of drones as aerial nodes so that they must be connected to the Internet [5]. As drones are a type of unmanned flying vehicle, IoD is a subset of IoFT [35]. An essential characteristic of IoD is the concept of well-defined “aerial routes” dedicated to the drone’s flight, named airways. This concept is based on the aerial navigation system of large-scale aviation but in a constrained environment, such as an urban region. Therefore, IoD is a proper architecture for urban sensing, representing a sustainable way to collect data from a given environment.

Figure 2.2 depicts examples of application scenarios where each terminology can be deployed. However, they are not limited to the provided examples. As previously discussed, the terminologies cover various environments and domains. A drone-based/UAV-based network refers, in a nutshell, to constrained UAV network infrastructures for specific domains in which the UAV trajectory is predefined or controlled remotely by an operator. FANETs are robust architectures composed of different aerial nodes (including UAVs) so that the network topology and communication occur in an ad hoc manner. IoD represents a new mobile network paradigm where UAVs compose a set of nodes sharing both the airspace and communication channels to perform their tasks connected to the Internet.

UAV/Unmanned Aircraft Systems are composed of UAVs that exclusively compose an aerial layer of nodes, connecting with the nodes in the ground layer to assist with some tasks, such as aerial surveillance. IoFT, in turn, is an extension of the IoT paradigm that considers different aerial nodes connected to the Internet to gather and spread data, such as UAVs, HAPs, and satellites.

To provide a broader understanding of the terminologies, Figure 2.3 provides a Venn diagram showing their relationships. The first point that differentiates the terms is the type of mobile node. In some cases, such as FANET and IoFT, other nodes beyond drones can be part of the network. We classify IoD as a subset of IoFT because the former focuses explicitly on network nodes comprising drones. The second point that distinguishes the terms is the connection with the Internet. IoD resides at the intersection of IoFT and UAV Systems since a UAV System becomes part of IoD when connected to the Internet.

Figure 2.3: Venn diagram representing the relations among the terminologies of UAV Networks



Source: Elaborated by the author

Drone-based Networks, as a part of IoD, consist of multiple interconnected drones linked to the Internet. Without this connection, Drone-based Networks are considered only part of UAV Systems. FANETs represent a specific type of a Drone-based Network and can be regarded as constituents of both IoD and UAV Systems. IoD is only considered when the FANET is connected to the Internet. Moreover, FANETs may include nodes that are not drones, thereby falling exclusively under the category of IoFT.

2.2.2 Discussion about the elements of UAV networks

In this section, we considered the literature and presented the essential elements of UAV networks and their terminologies. Table 2.2 shows how these networks can be defined following the considered notation. We discuss in detail these aspects as follows.

Table 2.2: General structure of UAV-based networks

Network	**Mobile Nodes**	**Nodes**			**Infrastructure**				**Environment**	
		N	GN	BS	ic	CP	ml	cc	fz	as
IoFT	$N = \mathcal{D} \cup \mathcal{AN}$	≥ 1	≥ 0	≥ 0	Y	≥ 1	Y | N	Y | N	Y	Y | N
FANET	$N = \mathcal{D} \cup \mathcal{AN}$	≥ 1	–	≥ 0	Y | N	1	Y | N	Y | N	Y	Y | N
Drone/UAV-based Network	$N = \mathcal{D}$	≥ 1	–	–	Y | N	1	–	–	Y	Y | N
UAV/Unmanned Aircraft System	$N = \mathcal{D}$	≥ 1	≥ 0	≥ 0	Y | N	≥ 1	Y | N	Y | N	Y	Y | N
IoD	$N = \mathcal{D}$	≥ 1	≥ 0	≥ 1	Y	≥ 1	Y | N	Y	Y	Y

2.2.2.1 Fundamental Elements of UAV Networks

Drones in civil environments evolved from a unique UAV controlled by a human to a group of cooperative UAVs. Due to this evolution, several architecture arrangements for UAV-based networks emerged, such as the B5G architecture [19] and the Internet of Drones [5]. Therefore, these architectures present different components and requirements in different scenarios.

Despite their differences, drone-based architectures have some fundamental components that serve as the groundwork for their deployment. These components can be described using the following notation:

- Network nodes (sets):
  - $\mathcal{D}$: network nodes composed of drones, uniquely;
  - $\mathcal{AN}$: aerial network nodes that are not drones, such as high-altitude balloons and vertical take-off and landing vehicles;
  - $\mathcal{GN}$: ground network nodes, such as related sensors, vehicles, and other devices;
  - $\mathcal{BS}$: base stations related to the deployed network.
- Network Infrastructure:
  - ic: it represents the connection to the Internet, which can be Y for yes and N otherwise;

  - $\mathcal{CP}$: set of communication protocols involved in the network. For instance, a network n_1 can use the MAVLink protocol, while a network n_2 can use UAVCan [36];
  - *ml*: it indicates the presence of middleware layers when the UAV network collaborates with third-party networks, which can be Y for yes and N otherwise;
  - *cc*: it indicates the use of a cloud, which can be Y for yes and N otherwise.

- Environmental Policies:
  - *fz*: it indicates whether regulations/restrictions related to the drone flying zones (e.g., military zones and airports) are in place (Y) or not (N);
  - *as*: it indicates whether the airspace has airways to fly (Y), including their traffic policies, or not (N).

2.2.2.2 Discussion Regarding the Types of Nodes

Table 2.2 shows that any network requires at least one mobile node, i.e., ≥ 1 in column N. The type of mobile node varies according to the terminology, although most networks are solely composed of drones. Other aerial nodes (e.g., high-altitude balloons) can belong to IoFT and FANET rather than drones. Although drones are the primary devices, ground nodes, such as terrestrial sensors, can also be present (≥ 0 in column GN), except for FANETs and Drone-based Networks that do not have. Similarly, base stations can manage/coordinate the network (column *BS*, value ≥ 0), being always present in IoD (value ≥ 1).

2.2.2.3 Discussion Regarding the Infrastructure

The connection to the Internet is a central component to differentiating the nomenclatures (column ic), being mandatory for IoFT and IoD since most applications receive and send information from the Internet, such as weather conditions. This connection is not required for other approaches, but it can exist. For instance, weather balloons of a FANET can feed the data of a climate website. Also, any network has at least one arrangement of communication protocols to interconnect the nodes. For instance, FANET has a single set of communication protocols since it is an ad hoc network [20]. Hence, all nodes use the same protocols (i.e., 1 in column *CP*). In comparison, drones of an IoD network can offer services to ground nodes using 6LoWPAN while using different protocols to communicate with the base stations and receive navigation directives (≥ 1 in column *CP*).

Another essential component is the presence of middleware to allow collaboration with third-party networks. Although this element is not mandatory for any infrastructure (Y | N in column *ml*), this is not a part of drone/UAV-based networks because the provided services are restricted to drones. These premises are valid in a cloud service (column *cc*). However, IoD must consider the fundamentals of this paradigm [5].

2.2.2.4 Discussion Regarding the Environment

Regardless of the network, there is a minimum regulation involving flying zones (column *fz*). These regulations range from basic flight authorizations (e.g., flight allowance over national parks) to navigation regulations, such as maximum allowed speed and the sharing of drone identification with government authorities [37, 38]. They impact the airspace organization based on the environment where the network will operate. Commonly, the networks consider the airspace free-to-fly. Hence, there are no policies regarding a shared space, except for collision avoidance mechanisms managed by the hardware of each drone [35]. Considering the current growth of DaaS applications, a collision represents a severe risk to deploying UAV networks in dense environments, such as urban centers [37]. IoD is the only paradigm to define controlled airspace to fly, expressed through the concept of airways (column *as*, value Y). However, airspace can be controlled using airways or another element in all other types of networks, where the flyable airspace is a graph whose edges represent the airways, and the nodes represent both aerial waypoints and PoIs, including free-to-fly areas [35].

2.3 Requirements of the Next Generation of UAV Networks

The growth of drones has been witnessed across various areas, including urban centers, rural areas, and industry. In urban environments, drones already provide some services, such as aerial photography and surveillance [39]. Also, some applications are currently being tested in terms of viability, for instance, delivery of goods [40]. In rural areas, drones have transformed agriculture practices by providing farmers with aerial data on smart crops, irrigation needs, and pest management [35]. Additionally, industries have embraced drones for tasks like aerial surveillance, monitoring of pipelines and power lines, and assessing hard-to-reach areas in sectors such as oil and gas, construction, and mining [35].

Given the prospected scenario, these services will compete for the use of airspace and also will collaborate intelligently, representing the next generation of UAV networks. They will have increased autonomy, enhanced communication capabilities, improved drone coordination, and seamless integration with other emerging applications [41]. As technology continues to evolve, these networks hold immense promise for shaping the future of aerial operations across various sectors.

In this direction, the next generation of UAV networks will demand various requirements to meet the provided service. In this section, we present these fundamental requirements. Table 2.3 summarizes how the mentioned terminologies address each requirement, identified by a label referred to throughout this section, facilitating a comprehensive discussion.

Table 2.3: Requirements of the next generation of UAV-based networks and their relations with the terminologies

Network Terminology	**Requirements**														
	Airspace Org.		**Fair Use of Airspace**			**Network Mgmt**			**Drones Collaboration**			**Energy Mgmt**		**Security/ Privacy**	
	RA1	RA2	RB1	RB2	RB3	RC1	RC2	RC3	RD1 – RD3	RD4	RD5	RE1	RE2	RF1	RF2
IoFT	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
FANET	✓				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Drone/UAV-based Network							✓	✓	✓						✓
UAV/Unmanned Aircraft System	✓				✓	✓	✓	✓	✓	✓		✓		✓	✓
IoD	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

2.3.1 Airspace Organization

Just as the growth in the number of ground vehicles demanded new regulations and changes in the road network, the expansion of the use of drones also points in the same direction. The design of strategies for the airspace organization can ensure the safe and efficient operation of UAVs [42], where establishing dedicated airways helps mitigate potential conflicts and collisions between UAVs. By designating specific flight corridors and altitudes, the network can handle effective traffic management, facilitating the integration of drones [43]. Indeed, the level of this organization is linked with the robustness and density of drones sustained by the environment.

The airspace organization can be treated differently for urban and non-urban environments. In scenarios with sparse traffic, such as non-urban environments, we might have a free flight of aerial nodes [5]. In this case, the network considers the whole airspace available for drones performing their services. Therefore, drones must manage themselves and deal with potential collision situations. Some networks (e.g., Drone/UAV-based, UAV/Unmanned Aircraft Systems, and FANETs) commonly assume this organization since the provided services are more restricted. On the other hand, free flight is not reasonable in urban IoD/IoFT due to limited airspace, obstacles such as buildings, and the high level of vehicle congestion that do not contribute to easy flight management [5, 43].

Therefore, the next generation of UAV networks will require a suitable airspace organization according to the robustness and the demands of the provided services. The related requirements are presented as follows.

- **RA1 – Aerial Zones**: Although airspace continues "free-to-fly", there are geographic boundaries involved with the provided service, where drones must be granted access. For instance, given a drone-assisted network coverage application, drones can be grouped to cover different regions [44], imposing a geographic delimitation for each one. Many applications, such as search and rescue [45], precision farming [44], spraying [46], and aerial imagery [45], to name a few, need a more flexible flight to attend to the mission's goal;
- **RA2 – Airways**: A direct way of airspace organization is the use of airways, as proposed for IoD [5, 38, 43, 47]. Airways can be established together with aerial zones, restricting the airspace where drones can fly and even the direction of a flight. This approach is suitable for urban areas, where it is possible to have many drones sharing the airspace [43]. Different proposals for organizing airways are found in the literature. For instance, Labib et al. [43] pointed out that each airway has its characteristics (velocity limits, flight headings, and maximum traffic capacity). The advantages of this approach are detailed in the sequence.
  - *Clarity of the UAV flight zone:* An UAV network can face challenges with public opinion due to the fear of having multiple drones flying over people. Thus, airways can make clear where drones can fly, helping with IoD acceptance.
  - *Collision avoidance:* Airways promote a predictable scenario about the drone's flight. Thus, it is reasonable to assume that airways can mitigate potential drone collisions, considering their rules of use [47].
  - *Laws and regulations:* The use of airways can make the creation of aerial regulations easier. Airways should address altitude limits, flight paths, and specific areas designated for drone activities [35].

2.3.2 Fair Use of the Airspace

Currently, the limited presence of drones operating within a specific geographic area mitigates the need for airspace competition management. However, this issue poses a significant challenge for environments with a great concentration of flying nodes. Therefore, formulating strategies to tackle a fair use of airspace is crucial to ensure equitable access to airspace resources and promote the efficient and safe coexistence of diverse drone operations [5]. Promoting a fair use of the airspace involves the following requirements:

- **RB1 – Airspace allocation**: This strategy represents the primary approach to promoting fair use of airspace by different drone companies, mitigating congestion, collisions, and conflicts. For instance, at some significant events, like football or concerts, we can have a lot of people concentrated in the same place. This situation can overload the cellular network, and it is possible to use drones as relay nodes to allow everyone to communicate. A company can allocate the airspace in the region during the events to allow this service. The airspace allocation can grant exclusive flight access to a group of drones in a given area at a specific time interval.

- **RB2 – Congestion control**: Similar to strategies taken by road-based navigation applications (e.g., Waze and Google Maps), congestion control aims to mitigate bottlenecks over the available airspace to fly. Considering the widespread of UAVs, aerial congestion control can be designed in scenarios where the network can collect data from the aerial nodes and create policies to manage aerial zones dynamically [35]. The management includes (but is not limited to) the restriction/allowance of flying zones, the definition of a maximum number of UAVs inside a region, and the creation/removal of airways. The guidelines to elaborate policies can consider different environmental aspects, such as the risk of aerial attacks [47] and the sensing of preferable routes to the UAVs [48];

- **RB3 – DaaS priority**: Even defining airspace allocation and congestion control strategies, some UAV applications surpass any defined policy due to the priority of the provided service. Analogous to road networks, emergency vehicles (e.g., ambulances) have priority over cars, trucks, and other common vehicles. Likewise, emergency/authority drones (e.g., providing health or police-assisted services) must prefer to fly freely over the airspace or also close to a specific region. This priority can be established using only the path planning (e.g., using priority constraints in the path planning [49]) or considering airways and aerial zone specifics for priority drones.

2.3.3 Network Management

Efficient network control is indispensable in any UAV network and can be achieved through effective management practices. In the context of UAV systems, the responsibility for management primarily rests with base stations that oversee the entire mission of a UAV group [25]. On the other hand, FANETs typically adopt a more centralized management approach, where one of the drones assumes the role of the base station and manages the other nodes in the network [20]. IoD needs a more intricate management strategy involving UAV communication and navigation policies. IoFT and IoD are specifically designed for large networks with many nodes [5, 35]. Consequently, their management approach tends to be distributed rather than centralized, unlike other terminologies that cater to smaller networks with centralized management.

Furthermore, UAVs have been deployed as Access Points (AP) and radio towers within the lower-altitude tier of B5G aerial access networks, enabling wireless and computing services to ground users. Likewise, they can act as a bridge between High-altitude Platforms (HAP) (e.g., satellites or stratospheric balloons) with the ground [50, 51]. Therefore, besides managing the UAVs, the next generation of UAV networks also requires intelligent and collaborative management with the abovementioned paradigms. These requirements are listed as follows.

- **RC1 – Network interoperability**: This requirement ensures that UAVs can collaborate and share information with ground base stations, control centers, HAPs, and water surface devices [52]. Therefore, UAVs can effectively provide the resources and capabilities of an integrated network, leading to improved operational efficiency, extended communication range, and the facilitation of novel applications embracing aerial, ground, and underwater networks [50]. These issues encompass the integration with B5G and 6G technologies, benefiting from Ultra-Reliable and Low-Latency Communication (URLLC) [51]. In this context, UAVs represent a promising set of devices for deploying Mobile Edge Computing (MEC) that enables computation, storage, and networking resources to be located closer to mobile devices at the network's edge. By bringing computing capabilities closer to the end-users and devices, UAVs can reduce latency, improve efficiency, and enable new applications and services in collaborative mobile networks [50];

- **RC2 – Topology changeability**: The next generation of UAV networks should exhibit the flexibility to adapt to changes in the network topology, environmental dynamics, and evolving user requirements [33]. Indeed, UAVs move fast through the airspace, which can affect deeply the nodes' connection and, therefore, the network topology [5]. Hence, the network enables it to efficiently respond to dynamic factors such as varying network conditions and geographic changes from collaborative networks [50]. The modeling of Software Defined Networks (SDNs) represents a proper strategy to handle the fast topology changes caused by the UAV's mobility [53, 54]. Likewise, the literature presents some efforts in the design of topology-based and routing protocols considering drone dynamics [32, 33, 34];

- **RC3 – Communication compatibility**: For successful integration with ground and HAPs networks, it is crucial to ensure compatibility in communication protocols, frequencies, and standards [35, 55]. The network must establish smooth and uninterrupted communication links with both networks, allowing for efficient collaboration and data sharing [41, 56]. The literature presents some efforts toward the compatibility and interoperability between different networks. For instance, the FlyNet system architecture [41] aims to support the composition of edge-to-core (i.e., end-to-end) platforms to support UAV-assisted applications, mainly in scientific domains. This architecture covers different categories of components (e.g., edge devices, edge servers, and core computing), providing guidelines to deploy the infrastructure, pursuing minimum latency and network compliance. Due to the plurality of communication protocols as well as strategies of data transmission, standardized network management solutions are significant challenges to be addressed.

2.3.4 Collaboration with Drones

Employing UAVs in mobile networks offers a significant advantage in facilitating extensive collaboration among diverse nodes, enabling effective communication and service provisioning [35]. This advantage stems primarily from the inherent attributes of drones, including their dynamic mobility and Line of Sight (LoS) communication capabilities. This collaboration can occur in two directions: among the UAV nodes in the network and nodes from different network paradigms (e.g., aerial and grounded networks). We discuss each direction as follows.

2.3.4.1 Collaboration Among Drones

Independent of the case, the network nodes collaborate. In all scenarios, drones are expected to collaborate through the exchange of messages or by altering their trajectories to assist other nodes in the network [35]. A coordinated exhibition of drones, aka “ballet of drones”[57], is a noteworthy example of their collaboration. The requirements related to collaboration are beyond drone synchronism, including:

- **RD1 – Routing**: Drones can collaborate to facilitate the network message’s exchange, which allows for the transmission of pertinent information across the network. There are many protocols available for routing between drones [7]. For instance, geographic routing is typically used to disseminate emergency messages in the network [58]. Also, it is common to use delay-tolerant protocols, considering that in some cases, the network can be sparse, and it is necessary to find another node to forward the message due to the high dynamic node connectivity [59]. Therefore, suitable routing protocols must be applied to cover this requirement [60];
- **RD2 – Collision avoidance**: Regardless of the node type, airspace organization, or network management, nodes are expected to collaborate to prevent collisions among themselves or other objects [61]. To this end, drones may transmit messages containing obstacle alerts, alter their course and speed, and maintain a safe distance from other nodes to uphold the physical integrity of the network. The literature offers numerous strategies for implementing collision avoidance, including a method based on a thin-plate splines algorithm [62], which aims to minimize deformation of the swarm’s formation while navigating around obstacles, approaches leveraging deep reinforcement learning [61], and methods such as rapidly-exploring random trees (RRT), among others [63];
- **RD3 – Goal achievement**: In a drone network, objectives are frequently accomplished through the collaborative efforts of multiple nodes. For instance, a search and rescue operation may require the participation of several drones to ensure swifter and more effective coverage of the search area. Drones can either collaborate on a particular task to attain the mission objective [64], such as in scan activities, or each drone can operate distinct tasks contributing towards the common goal [65]. For instance, in an urban sensing application, a drone can collect data about the traffic, and others can collect data about air quality. Although the tasks are distinct, the common goal is to extract data from the city.

Collaboration can manifest at varying degrees in scenarios with multiple node types. For instance, some messages may be routed across all network nodes, while others may only be restricted to drone-to-drone communication. Such management decisions may hinge on the mission objective. Regarding airspace management, where a constant flow of drone traffic is expected, collaboration can also play a key role in maintaining network efficiency. In IoD, for instance, an alert message may be disseminated to all drones in a particular area in case of a security threat (e.g., the detection of smoke). The affected drones can then alter their routes accordingly. Other forms of collaboration in networks with a high and continuous volume of vehicle traffic, as is typical of IoD, include:

- **RD4 – Weather information**: One of the significant challenges of drone networks is the influence of the weather. In certain countries, regulations stipulate that drone operations are contingent upon weather conditions aligning with the manufacturer's specifications [66]. Drones can access this information and decide to interrupt a mission. In IoD, this information can be received in real-time, and drones can automatically determine what to do in case of bad weather [5]. In some cases, a drone that has access to the Internet can forward this message to other drones that are not connected to the Internet. Adhering to drone legislation necessitates flexibility in deployment schedules to accommodate potential weather-induced delays or cancellations, extending beyond the manufacturer's defined operating parameters [66]. Thus, drones can collaborate by passing on news related to the weather so that it is possible to maintain the security of the network [5];
- **RD5 – Traffic congestion avoidance**: One way of organizing many drones in a network with continuous traffic is using airways. When planning the path planning of drones, it will be necessary to consider the airways with the highest flow so it will be possible to avoid vehicle congestion on the airways. This type of collaboration must happen between all services and applications. Strategies used in vehicular networks to avoid traffic congestion [67, 68, 69] can be considered. For instance, real-time path planning can alleviate traffic congestion in an urban environment [67]. In the case of drones, real-time path planning can be more advantageous, considering that the "human factor" in vehicular networks can alter the route more. For instance, a person can make additional stops or change the final destination.

2.3.4.2 Collaboration Between Drones and Third-party Networks

Besides the collaboration among the drones themselves, UAVs can contribute to other mobile network paradigms in the air, on the ground, and even underwater [51, 52]. Drones and HAPs can establish resilient communication networks over vast regions. Drones can perform as relays, exploiting their agility and adaptability to bridge communication gaps and establish connectivity in remote or challenging-to-access areas (also representing a collaboration with ground networks) supported by HAPs [70]. Moreover, drones started collaborating with terrestrial vehicles as an active modal in parcel delivery systems, operating in the last mile [71]. The air-to-ground collaboration mediated by drones fosters data exchange, real-time monitoring, and distributed computing.

UAV networks might also collaborate with underwater networks, mainly UAV-assisted data acquisition systems [52]. In this environment, sensor nodes are spread underwater, sending data through acoustic signals to surface buoys. Thus, drones fly over the buoys to gather data and transmit it to the responsible network authority.

The requirements related to this type of collaboration deepen the ones presented in the previous section, in which routing protocols (RD1) represent a crucial aspect of promoting the collaboration, as well as the deployment of aerial policies to organize the airspace better (RD2, RD3, and RD5).

2.3.5 Energy-related Management

Maintaining a continuous flow of drones is crucial to providing a high QoS. However, this aspect represents a significant challenge for many applications due to their limited energy. To address this issue, investing in developing hardware solutions to increase the drone's energy capacity could be a viable solution [72]. Additionally, it is essential to consider energy efficiency in developing applications, methods, and protocols. Nevertheless, given the energy restrictions of drones, completing missions may require alternative approaches, leading to two requirements:

- **RE1 – Mobile recharge/piggybacking stations**: To extend the drone's range and carry out longer missions, it is essential to plan for recharge stops. These stops can be made at recharge stations, enabling the drone to cover greater distances. Studies in the literature seek to develop methods for distributing these points into suitably located charging stations [73, 74]. Two objectives are explored: the number of stations to be deployed and minimizing distances traveled [73]. Different station locations are explored, such as building tops [75] and buses [76]. In addition, taking a ride on other vehicles such as buses, trucks, and trains can further increase the drone's coverage and range [40], even with stops for recharging its battery [77];

- **RE2 – Collaborative multi-modal mission**: To ensure the successful completion of tasks, it is important to plan them considering the impossibility of recharges. In this case, the UAV network can plan a collaborative mission with a subsequent UAV or even with another modal, such as public transport or trucks [78]. For instance, drones can provide a last-mile delivery service, being responsible for carrying the package on the last part of the trip [79]. Coordination between different modes has been the biggest challenge in this requirement [78]. This problem is modeled as mixed-integer linear programming (MILP), an NP-hard problem [80]. Different algorithms are used to solve it, such as the greedy heuristic algorithm [80], an adaptive extensive neighborhood search (ALNS) [81], a combination of column generation (CG), and the logic-based Benders decomposition for designing a hybrid algorithm [82], among others.

Recharging stations can play an essential role in IoD. They can be strategically placed in buildings for easy drone access. Another suggestion is to add recharge points to the vehicles used for transporting drones, allowing the drone to partially or entirely recharge its battery during the ride.

2.3.6 Security and Privacy

As the growth of UAVs takes place, there is an expansion in the requirements for security and privacy, extending beyond the single UAV's perspective and encompassing a broader scope [83]. Indeed, security and privacy requirements are mandatory for all networks, but UAV networks have particular aspects that demand attention. Hence, ensuring standard security requirements, such as the encryption of the communication channel, preventing unauthorized access, maintaining fault tolerance, and managing battery efficiency, is insufficient to provide a proper level of security/privacy for the next generation of UAV networks [56, 83, 84].

We discuss the critical requirements as follows.

- **RF1 – Third-party data protection**: Given that drones will compose an integrated and collaborative environment, drones can carry sensitive information of network users. For example, delivery drones need to know personal information, such as the delivery address and who the delivery is destined for. A drone with a hospital delivery address could carry medicine, blood, vaccines, or other medical material [85]. In this scenario, an attacker could obtain the delivery address using Software/Hardware exploitation [86, 87]. With the delivery address and through inference attack [37, 88], it could determine what type of cargo that drone is carrying and, in this case, decide to divert the drone from the route and steal the cargo. Hence, security/privacy requirements must protect related third-party users/clients. These requirements can be ensured by designing lightweight authentication protocols, well-defined flight policies, and applying data anonymization or obfuscation techniques based on a required level of data utility [37, 38].

- **RF2 – Emergency control**: This requirement involves the actions taken by drones as a device and by the network to mitigate the side effects of emergencies. Thus, the network must ensure protocols for emergency landing, fail-safe mechanisms, autonomous return-to-home, and navigation guidelines in case of loss of connection [89]. Most of these situations involve the drone finding a location for an emergency landing. In this concept, it is necessary to automatically recognize a landing site considering nearby points, maneuver towards it, and perform a safe landing given the emergency scenario [90, 91]. In this context, Convolutional Neural Networks (CNNs) are attracting significant attention from the academic community to generate models that analyze images in real-time to identify safe landing locations [90, 91, 92, 93].

2.3.7 Discussion about the Requirements

In this section, we discuss the interrelation between the requirements presented previously. Considering that the next generation of UAV networks is expected to have a large number of drones, the complexity of managing those requirements is challenging. In many cases, the demands are related and affect each other in a complicated way to consider individually. For instance, achieving proper airspace organization, network management, and effective collaboration between drones contribute to ensuring fair use of the airspace. Furthermore, energy-related management is directly linked to network management.

In some cases, network management is responsible for addressing energy-related issues. For example, in a given scenario, the base station might be tasked with network management and path planning, needing to consider the energy capacity during the path planning process. Another energy-related aspect is collaboration, where the interaction between drones and third-party networks can influence energy consumption. Drones can take a ride in other vehicles, such as trucks and buses, so collaboration also affects energy [94].

Airspace organization also influences other requirements. Well-planned airspace organization facilitates fair airspace usage and streamlines network management processes. Moreover, security and privacy aspects are impacted by all the other demands. For instance, inadequately established airspace organizations can lead to security vulnerabilities leading to collisions of drones. Similarly, energy is another crucial requirement tied to the operation of a drone. Also, collaboration among different drone companies can potentially lead to data breaches, whose privacy can be exploited by adversaries. Hence, it is evident that every requirement in UAV networks is closely linked to security and privacy concerns.

2.4 IoD Properties

The drone is a type of vehicle, as a FANET is a subgroup of a Vehicular Ad hoc Network (VANET), which in turn is part of a Mobile Ad hoc Network (MANET) [1, 95]. Table 2.4 presents the main differences between IoD and the other networks. The significant difference between UAV networks and IoD is that in IoD, almost all characteristics are variable and depend on the mission. The variability extends to attributes such as topology, which can shift markedly based on the drone's mission objectives. For example, when monitoring traffic at an intersection, the topology may display less rapid dynamics than instances involving a police drone engaged in a pursuit. In addition, IoD diverges notably from conventional UAV networks in its capacity to access the Internet. Furthermore, IoD can exhibit an airway-oriented structure, necessitating congestion control, and coordination mechanisms to regulate drone movements effectively.

Table 2.4: Comparison of MANET, VANET, UAVs Networks and the Internet of Drones

	MANET	**VANET**	**UAVs Networks**	**Internet of Drones**
Nodes	Mobile devices	Vehicles and road side units	UAVs and base stations	UAVs and base stations
Topology	Random, ad-hoc	Star and ad-hoc	Star, mesh, ad-hoc	Star, mesh, ad-hoc; delimited with airway or not
Topology changes	Dynamic	Dynamic	Dynamic or stationary; depends on the mission	Dynamic or stationary; depends on the mission and the number of airways (if airway-based)
Mobility Speed	Slow	High	High and slow; depends on the mission and the type of UAV	High and slow; depends on the mission, the type of UAV and the airway speed limit (if airway-based)
Mobility Model	Usually random waypoint	Highways or roads	Predetermined path or not	Predetermined path or not; airways or free flight
Management complexity	High and self-configuring	High and self-configuring	High and self-configuring or low and non self-configuring; depends on the mission	High and self-configuring or low and non self-configuring; depends on the mission; it is also expected to control multiple drone networks at the same time (high difficulty of control)
Energy	Usually battery powered	Car or own battery	Own battery and lifetime usually low	Own battery and lifetime usually low, but it expected the possibility of recharging the nodes during the mission

Changes in topology are much more variable in UAV Networks than in MANETs and VANETs. Usually, the topology in ad-hoc networks is very dynamic. In UAV networks, the changes can be more frequent due to factors such as the need to recharge, the occurrence of a malfunction, and the constant variation of UAVs' relative positions [1].

The topology issues also apply to IoD. Besides, IoD may influence the topology due to the airways (when airway-based) or no-fly zones (e.g., airports, governmental buildings, and environmental protection areas). UAVs must follow predefined rules related to fly regions. Therefore, network topology changes are associated with existing airways or no-fly zones. In single-UAV and star networks, the topology tends to be more stable and even stationary, but the network is still subject to the factors mentioned above.

The node's mobility speed is a characteristic that influences the topology dynamicity. While in MANETs the speed usually is slow, in VANETs, the speed is considered high. However, in IoD, the speed has a considerable variation due to the type of drone and the mission specificity, which can be low or high. Also, the possibility of moving in three dimensions is a factor that is different from other ad-hoc networks. A central difference affecting mobility is that VANETs have a predictable way of moving since staying on streets and highways is mandatory. On the other hand, it is possible to have both predetermined and non-predetermined paths in UAV networks. Airways can influence speed and direction.

When the dynamicity of nodes forces the network to organize and re-organize frequently, the management complexity is usually high. The control can be higher in IoD due to the possibility of controlling multiple drone networks. Another point that can influence the control complexity, mainly in large drone missions, is energy. Drone commonly has a low lifetime battery compared with vehicles and other mobile devices, which can affect the available time and the success of a mission. However, the possibility of recharging the nodes is expected in extended missions in IoD. In the following sections, IoD network topology types and IoD drone mobility models will be discussed in detail.

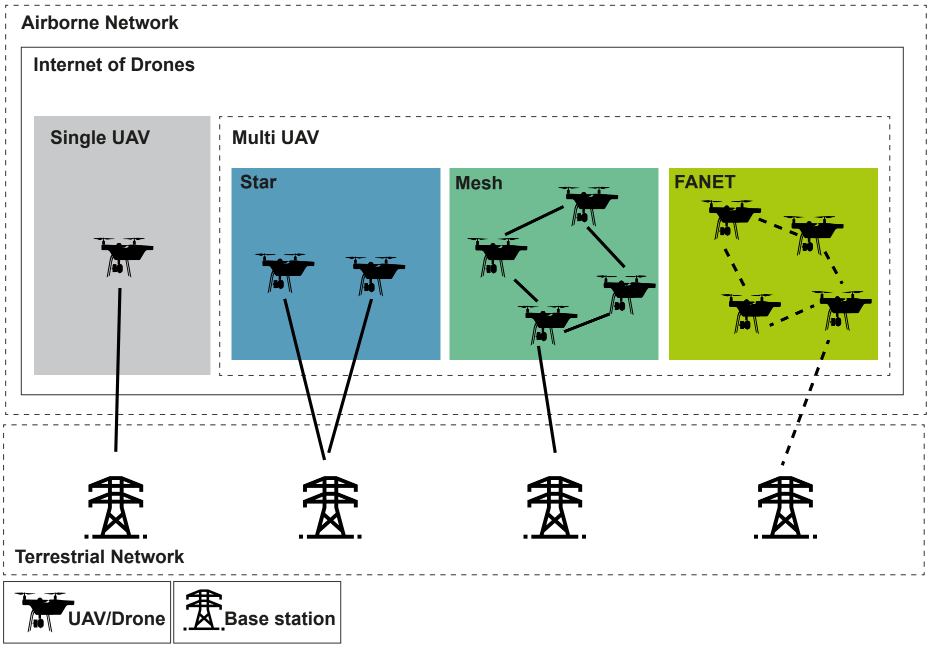
2.4.1 IoD Topology

Integration of both single-UAV and multi-UAV networks can be seamlessly incorporated into the IoD. IoD constitutes a network that comprises multiple sub-networks or swarms of drones. Each swarm can be driven by distinct objectives, such as delivery or monitoring tasks, among others. Alternatively, disparate swarms can also share a common goal yet lack the need for mutual collaboration. For instance, two companies engaged in drone-based goods delivery share the overarching objective of delivering goods; however, they operate independently without collaborative efforts to achieve this shared goal. Consequently, subgroups of drones within IoD may either cooperate or operate autonomously to attain specific aims. Nonetheless, a crucial prerequisite for their inclusion within the IoD is obligatory cooperation concerning network traffic management.

This entails averting collisions and congestion, overseeing the routing of network administration packets, and ensuring impartial utilization of the airspace. Several prevalent topologies are often employed in multi-UAV networks, including star, multi-star, mesh, and hierarchical mesh configurations [1]. Another noteworthy network architecture is the ad-hoc network, denoted as FANET (Flying Ad Hoc Network), characterized by its absence of a fixed topology. After this overview, we will delve into the definitions of single and multi-UAV networks.

- **Single UAV Networks**: The simplest UAV network. It is composed of one UAV and a controller;
- **Multi-UAV Networks**: The network consists of more than one drone. In addition, one or more base stations can be part of the network. Different topologies are adequate for multi-UAV networks, such as star, mesh, and FANET, as shown in Figure 2.4.

Figure 2.4: UAV Networks taxonomy



Source: Elaborated by the author

Multi-UAV networks have the advantage in some factors compared with Single UAV networks, such as allowing greater agility in solving tasks, greater survivability, extended scalability, and low cost for maintenance of small UAVs compared to large UAVs [95]. On the other hand, they are more complex to manage. Following, we explained the concepts related to star, mesh, and FANET networks. FANETs will be highlighted due to these networks' prominence in recent years [95, 96].

- **Star**: All UAVs must be directly connected with one or more base stations, as shown in Figure 2.4. Moreover, UAVs routed every communication among them through the ground nodes[1, 97]. A similar variation is a multi-star topology. In star topology, the UAVs form groups, and just one node of each is connected to the base station [1];

- **Mesh**: In mesh networks, the UAVs are interconnected and communicate with each other by a hop or multi-hop without ground control intermediation, as depicted in Figure 2.4 [97]. However, few drones are connected to the base station [1]. A mesh network variation is a hierarchical mesh network. In this case, the drones form groups. Besides the UAV-to-UAV communication inside the groups, they have inter-group communication, and there is a hierarchy among them;

- **FANET**: When a UAV network is ad-hoc, it is called a FANET. The UAV communication is through the UAV-to-UAV data links instead of UAV-to-infrastructure data links [95]. If a node disconnects from the network or does not establish a communication link with the infrastructure, the operation will still work using the other UAVs. Specifically, if needed, any UAV can connect with the base station. FANETs are distinct from mesh networks, in which specific and predetermined nodes can establish this connection.

All the mentioned topology types can coexist within the IoD. Thus, it is possible to have a fleet of UAVs running the star topology and another using the mesh topology. And all of them are part of the IoD and need coordination to work in the same air space and share information that can keep the network secure. For instance, information regarding the traffic (e.g., block areas and traffic congestion), weather, and alerts about some situations (e.g., disasters, attacks).

2.4.2 IoD Mobility

Real-life UAV mobility has two classifications: a predetermined path and a non-predetermined path. On the non-predetermined path, it is possible to have a pilot controlling the UAV or a random flight. On the predetermined path, we have two possibilities. First, it is possible to prepare the entire route before the flight. Second, the UAV receives the route in real time during the flight. In the latter case, we have the whole path before the mission begins. However, the system provides the UAV just a part at a time. In the literature, some mobility models are used to simulate UAV mobility in IoD [98].

Bujari et al. [98] classified the UAV mobility models into five categories randomized, time-dependent, path-planned, group, and topology-control-based mobility models. Table 2.5 summarizes the description and scenario of the use of the mobility models more suitable for UAVs.

The first category (randomized mobility models) allows mobility to be random regarding speed, direction, and time. Some possibility of this group is Random Walk, Random Waypoint, and Random Direction. In the first, a new speed and random direction are chosen at each constant time t. In the second, a new random destination and a uniformly distributed speed are selected every time the mobility node reaches a destination. Finally, in the third, a new direction is chosen when the mobile node reaches the simulation's edge. Despite these possibilities, Bujari et al. [98] mention that these options are not suitable for UAVs, meaning they do not represent a real-life drone movement. However, the Manhattan grid model described in Table 2.5 can be used in some cases in UAV applications.

The second category, time-dependent mobility models, represents the models that try to perform a smooth change of direction and speed, such as the Gauss-Markov, and Smooth Turn models, described in Table 2.5. Although these models try to avoid sharp changes, they still are not entirely adequate to represent real-life UAV movements. However, Bujari et al. [98] point out that in search and rescue missions and reconnaissance and patrolling by UAV, these mobility models are suitable, for instance, in a random search on a specified target area.

The third category contains the path-planning mobility models, such as Semi-random Circular Movement and Paparazzi model, described in Table 2.5. Path planning is one of the most used mobility models in real-life tests of UAV applications. The fourth category is group mobility, such as the Column and Pursue model. These models move the UAVs intending to complete missions quickly and efficiently. For example, if organized in columns, UAVs cover a search area more rapidly in search and rescue missions.

The last category contains topology-control-based mobility models. In real-time mobility, the speed and direction are controlled in real-time by the control station. The necessity of real-time controlling can be related to maintaining a fully connected network of UAVs at all times so that the UAVs can talk with any other [98]. Another case is the Internet of Drones scenario, proposed by Gharibi et al. [5]. In IoD, the drones might receive only a part of the entire path due to airway restrictions, and each period they receive a new segment of the path. Another instance in this category is multi-tier mobility. The multi-tier mobility considers different types of UAVs flying at different altitudes. This mobility model also can be used to simulate the IoD architecture [5]. All the models mentioned are suitable for non-airway-based scenarios. However, Smooth Turn and Paparazzi models are not adequate for airway-based environments. On the other hand, Collum and Pursue are adaptable to work in the airways.

Table 2.5: Suitable Mobility Models for Simulate UAV Movement

Mobility Model	Description	IoD Mobility Model Use Scenario
Randomized Mobility Models		
Manhattan Grid	The Manhattan Grid (MG) mobility model uses a grid road topology, which aims to describe movement in an urban area where the street layout is very regular	This model imposes geographic restrictions on mobility. It is suitable for an application that simulates the UAV airway.
Time-dependent Mobility Models		
Gauss-Markov	The Gauss-Markov (GM) mobility model uses several parameters to generate dynamic trajectories with time-varying velocity and direction [99].	It can be applied in scenarios where it is necessary simulated the UAVs flighting in airways or free flight. However, this model cannot reproduce a typical behavior of UAVs, such as turns [98].
Smooth Turn	In the Smooth Turn (ST) mobility model, each UAV selects a point in the space and then circles around it until the UAV chooses another turning point.	It can be used to represent a random mobility model for UAV and can be applied in scenarios where it is necessary, for example, a random search. Also, that is unsuitable for airways.
Path-planned Mobility Models		
Semi-Random Circular Movement	The Semi-Random Circular Movement (SRCM) mobility model simulates UAVs turning around a specific point.	It is applicable in scenarios where the UAV needs to scan or surveillance a specific area. Also, it is not proper for airways.
Paparazzi	The Paparazzi Mobility Model (PPRZM) is a stochastic mobility model based on a state machine containing six-movement pattern states: Stay-At, Eight, Oval, Scan, and Way-Point [100]	This model is adequate for cases in which is needed to determine the paths that the UAV will follow. However, this model is not applicable for airways-based scenarios.
Group Mobility Models		
Column	In the Column Mobility Model (CLMN), each mobile node moves around a reference point placed on a given line, which is moving in a forward direction [98].	This model can be used in many scanning and searching applications scenarios. Also, it is applicable in airway-based scenarios. However, the UAVs must follow the airways to flight
Pursue	In the Pursue Model (PRS), the mobile node pursues a specific target using simple relative motion.	This model is used, for example, in police pursuits or applications related to entertainment such as cinema and photography. Also, it is applicable in airway-based scenarios if the UAV pursues a target using the airways.
Topology-control-based Mobility Models		
Real-time	UAVs receive their paths in real-time from the control station.	Used for UAV real-time path control, mainly when network or mission constraints have to be continuously satisfied. It also can be used in an application that simulates the UAV airway.
Multi-tier	There are different types of UAVs flying at different altitudes [13]	Used in heterogeneous networks that need mobility models that incorporate multiple mobility patterns [13]. One use case is the IoD architecture proposed by [5].

2.5 Protocols Design for IoD

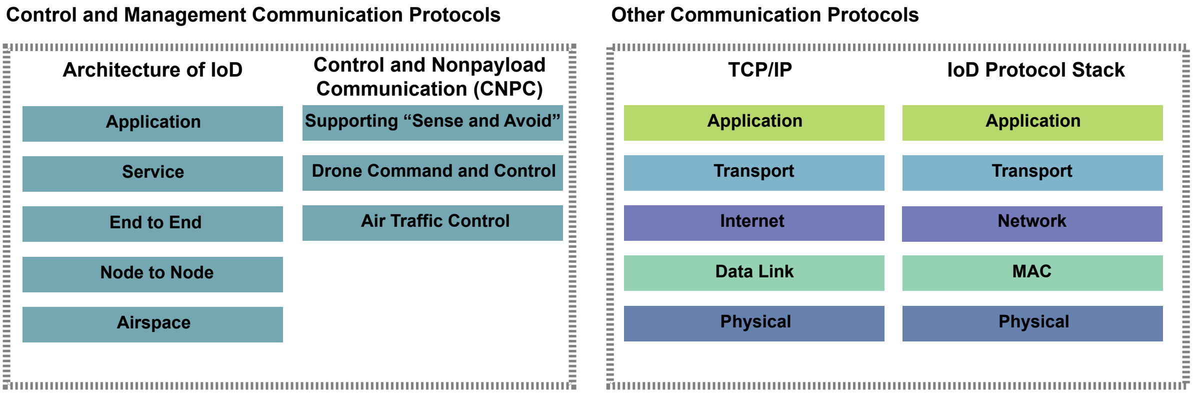
This section discusses critical aspects of the design of IoD protocols. Because IoD is an emerging area, the standards for each step of the protocol stack are still developing. The protocols for IoD have two categories: Control and Management Communication Protocols and Other Communication Protocols. Gharibi et al. [5] defined a layered architecture for IoD management. The architecture is composed of the layers:

- **Application**: applications must consider the architecture structure to use the services provided by IoD;
- **Service**: responsible for providing a common platform for broadcasting messages related to network management;
- **End-to-end**: responsible for end-to-end control, routing, handoff, and notification of congestion;
- **Node-to-node**: responsible for node-to-node control, broadcast node-to-node, plan pathway, notify congestion, manage emergencies in drone software and hardware, refuel;
- **Airspace**: liable for the network control, planning of trajectories, informing the weather condition, preventing collisions between aircraft, having the map of the places where the drones can fly, and periodically receiving the drone position.

The architecture of the Gharibi et al. [5] is related to the primary communication requirements in a drone network. As shown in Figure 2.5, this architecture seeks to support Control and Management Communication. Protocols in this category must be related to air traffic control, drone command and control, and support "sense and avoid" communication [101]. The IoD architecture is still being improved. Therefore, the development of protocols for it is still a big gap. As the IoD is established and the architecture is put into practice, the development of protocols for network management and control is expected to evolve.

The other types of network communication protocols are related to TCP/IP. Each protocol stack layer must be changed to be more suitable for IoD. However, there is a clear relationship between the two protocol stacks, as shown in Figure 2.5. Regardless of the layer, some points must be considered when developing protocols for IoD. Specifically, high node mobility, airways, energy-efficient, fairness, collaboration, location, and security. These aspects are described in Table 2.6. Thus, this section highlights the requirements for creating specific and efficient IoD protocols.

Figure 2.5: Overview of IoD architecture [5] and IoD protocol stack



Source: Figure based on [5]

2.5.1 Physical Layer

The physical layer protocol must consider 3D movement and high node mobility. Experimental communication between the drone and other devices such as remote control, base station, and IoT devices use radio waves. Communication for control and video transmission generally uses ISM (Industrial Scientific & Medical) frequencies that are not licensed, and the most used are 900 MHz, 1.3 GHz, 2.4 GHz, or 5.8 GHz [102]. The 5.8 GHz band is more common for video due to its short wavelength and significant data rate transfer capability [102].

As drone is an emergent technology, communication standards mostly related to drone-to-drone communication are still in development. Due to the lack of standards, currently, applications have been using different protocols in drone-to-drone communication, such as LTE (Long Term Evolution), WiFi, Bluetooth, and WAVE (Wireless Access in Vehicular Environments). The diversity of protocols is also because there are many types of UAVs, and each supporting different technologies. Also, it is necessary to choose the protocol according to the applications. For instance, LTE has a wide coverage area, but the energy consumption is higher, which is a problem considering that drones have a limited battery capacity. On the other hand, Bluetooth consumes less energy, but the transmissions are more subject to interference [35].

The Institute of Electrical and Electronics Engineers (IEEE) is developing future UAV standards. Due to this fact, commercial Drones do not have any specific communication interface for D2D communications [103]. IEEE started a project that seeks to standardize several drone-related issues. The IEEE P1920.1 standard defines air-to-air communications for self-organized ad hoc aerial networks. Another standard is P1920.2, which proposes the protocol for exchanging information between the vehicles for Un-

Table 2.6: Requirements for Protocols design for IoD

Requirements	Description
High nodes mobility	The protocols must consider the high mobility of the nodes. This factor provides frequent disconnections of nodes and changes in network topology.
Airways	The use of airways is a scenario that may exist in IoD. This factor provides a more controlled environment for the mobility of drones. Thus, the design and adaptation of protocols must take into account this unique characteristic.
Energy Efficient	Energy is a crucial issue in IoD. The drones usually have a limited battery. Therefore, developing protocols in this scenario must consider energy.
Fairness	As IoD is expecting a large number of drones flighting at the same time, fairness is an essential issue that protocols must guarantee for the proper functioning of the network.
Collaboration	In IoD, the collaboration between drones must be essential in the development of protocols. Whereas, drones must communicate with each other to send out congestion and collisions.
Location	In IoD drone location is an essential factor for network management. Through it, it is possible to avoid congestion and collisions of UAVs. Thus, this is a factor that is import consider in the development of protocols. It is also important that this location is not given only by Global Positioning System (GPS), but also by prediction algorithms.
Security	With a large number of drones connected to the network at the same time, security is an essential requirement in the development of protocols. It is important to consider both physical (e.g., collision avoidance, physical attacks) and data and information security that drones may be carrying.

manned Aircraft Systems.

2.5.2 MAC layer

MAC protocols should consider the different kinds of applications that can be suitable for IoD. Like other networks, in IoD, the MAC layer must also provide reliable, fair, and efficient channel access [104]. Researchers have proposed modifications to these protocols to make existing MAC protocols workable for UAV networks [105]. However, these protocols do not work correctly on UAV networks. Both traditional protocols for omnidirectional and directional antennas have challenges to overcome. In the case of Omni-direction, some issues are a short transmission range, high energy consumption, restricted network capacity for simultaneous communications, difficulty in providing fair channel access to all users, limited spatial reuse and packet transmission, and limited security[105].

In the case of directional, some challenges are the hidden terminal problem, failure to respond as the receiver beamformed in another direction due to repeatedly trying to transmit to an intended receiver(deafness), head-of-the-line (HoL) blocking, the retirement of useless packets resulting in the channel under-utilization (MAC-layer capture) [105]. These questions about using existing MAC protocols for UAVs are detailed in [105], a survey about MAC protocols for UAVs.

The challenges related to the MAC layer and UAV networks mentioned by Vashisht et al. [105] are fairness, security, energy-efficient, location awareness, and collaboration task allocation. The same points are essential to IoD architecture. IoD can enable a scenario in which there will be a large number of drones flying and communicating at the same time. As a result, attaining fairness is expected to be even more challenging in IoD, as there will be more contending nodes. Another critical issue is security. The IoD enables the sharing of air space by different companies that provide various services. Consequently, the greater the number of applications and services, the greater the sensitivity of the data that can be accessed. Therefore, security becomes even more challenging and necessary in MAC protocols.

IoD also aims to be a structure allowing drones greater energy autonomy. Thus, the protocols must have energy efficiency as a requirement. IoD also intends to enhance the integration of drones with other networks, such as vehicles and IoT devices. Therefore, the development of MAC protocols for these cases is also essential. Finally, MAC protocols for IoD must be explicitly developed for scenarios in which the IoD architecture considers the existence of airways. The use of airways can help predict and estimate the location of drones. When developing MAC protocols considering these characteristics, obtaining more optimized and reliable protocols may be possible.

2.5.3 Network layer

Routing protocols for IoD must consider specific requirements to provide reliable communication. Some of them are the high mobility of the nodes and the frequent topology changes [7]. Another important point is mobility. Specifically, if the drone flight or not over an airway. As mentioned before, in our definition, if the UAV can access the Internet, it must be part of the IoD. However, some architectures, such as Gharibi's definition [5], consider that the IoD must have airways to control access to the airspace. Therefore, we have two categories of routing protocols:

- Non-airway-based protocols: These are the protocols developed without considering the airways. Most protocols developed for UAVs did not consider the existence of airways when they were being designed. However, they can be used for IoD that does not have the airway requirement, and also they can be potential protocols for airway-based IoD;
- Airway-based protocols: These are the protocols developed considering the existence of the airways since the beginning of the development of the protocol.

These two categories can support the unicast, multicast, geocast, and broadcast paradigms. In the unicast paradigm, a source node must transmit to a destination node via multi-hop wireless communication. In the multicast/geocast paradigm, a source node must transmit to a group of nodes. In geocast, the group of nodes is in a specific geographical position. And in the broadcast paradigm, the source nodes must transmit information to all their neighbors at the same time.

Due to the similarity of IoD with MANETs and VANETs, researchers are studying the applicability of the routing protocols developed for these networks for possible use in aerial networks [1]. However, IoD has specific requirements that can differentiate how these existing routing protocols perform in aerial networks. Specifically, the energy constraint, the frequent disconnect of the nodes, the mobility patterns and speed, intermittent link management, and the access control to the airspace are factors that influence the performance of routing protocols. Considering this fact, many routing protocols developed for MANETs and VANETs were modified for UAVs. Also, new routing protocols were proposed [1, 7, 106].

As previously discussed, it is evident that as the future envisions a substantial proliferation of UAVs engaging in diverse services and competing for airspace, the imperative arises for the coordination and regulation of this aerial domain. A viable approach to address this challenge involves implementing airways [5]. Establishing airways introduces a structural constraint on UAVs, potentially necessitating adaptations to prevailing routing protocols to optimize operational efficiency. The incorporation of airways offers a potent mechanism for the control and coordination of airspace activities. However, the integration of airways introduces a novel dimension that many extant routing protocols have yet to consider. The presence of airways endows an airway-based IoD with certain similarities to VANETs. It's essential to acknowledge that the nodes in this IoD scenario are characterized by swifter movement and more frequent disconnections, setting it apart from conventional VANET scenarios.

2.5.4 Transport and Applications layers

Like VANETs, IoD also has the issue of poor performance of UDP and TCP protocols [104]. The reason is mobility at high speed and frequent topology changes. Researchers have proposed modifications for TCP and UDP protocols to overcome these issues. For instance, the use of Space Communications Protocol Specification (SCPS) - Transport Protocol (SCPS-TP)[1] for FANETs has been mentioned in [107]. This stack protocol is an extension and modification of TCP/IP.

For airway-based IoD, protocols developed and used for VANEts, such as Vehicular Transport Protocol (VNT) [108] and Mobile Control Transport Protocol (MCTP) [109], could be tested and adapted, if necessary, to be suitable for IoD. Despite the mentioned transport protocol options, IoD requires new approaches not based on traditional transport protocols. Developing trusty transport protocols is a challenging design problem since IoD is an emergent area with unique characteristics.

IoD requires the precise coordination of drones. The application layer needs to prioritize and minimize the end-to-end communication delay. So, the application protocol does not compromise the smooth functioning of the network. In addition, drones may be performing critical tasks involving life issues. Consequently, risk and emergency messages may be traveling on the IoD. In the same way that ambulances have priority on land traffic, some drones and some types of messages may require greater urgency than others.

Another issue is that the application protocols must be built to optimize the sending and collection of data considering the energy cost. Many applications related to drones perform data collection that can generate a significant expenditure of space and energy (such as videos and photos). Considering this factor, seeking solutions to minimize these effects is necessary.

---

[1] https://www.iso.org/obp/ui/#iso:std:45986:en

2.6 Joint IoD and Other Networks

In future smart cities, it is reasonable to assume that distinct networks will collaborate between them to improve network communication and safety. Prior studies have discussed how drones can aid VANETs [110, 111], Public Transportation Network (PTN)s [112], and Internet of Things (IoT) [12, 75, 113, 114]However, to the best of our knowledge, this work is the first to discuss how IoD and Terrestrial Networks can form heterogeneous networks to collaborate and benefit from each other, leveraging future services and applications for their users.

The integration and connection of autonomous vehicles have been explored extensively to make Intelligent Transport System (ITS) viable. Shortly, smart cities will have a reliable and autonomous intelligent transport system. In addition, different types of vehicles are expected to be integrated. People move around using distinct transportation modes such as buses, cars, and bicycles, and the choice depends on certain conditions such as weather, availability, and costs. Analogously, the integration between networks depends on several factors. Specifically, availability, cost, energy expenditure, weather conditions, network coverage, to name a few.

Another issue is that when people move from an origin to a destination, the path can be composed of several transportation modes. Especially, a person can travel a stretch of the route by bus, another by subway, and another by bicycle. Thus, the integration between networks can occur in the same way. More than one network can cooperate depending on the situation and conditions at a given time. In this Section, we highlight how IoD and heterogeneous networks have the synergy to improve network connectivity, energy consumption, routing, and data delivery.

2.6.1 Characteristics to Consider

Understanding the characteristics of each network allows intelligent integration so that there are mutual benefits. Thus, it is necessary to know each network's features, such as connectivity, energy consumption, resilience capacity, and the most appropriate scenarios.

VANETs are networks that, over time, may have certain predictability of behavior. For instance, at lunchtime, we have flows of vehicles more significant than during the night. Therefore, flows vary according to the hour and day of the week. Another point to consider is, for example, holidays. Thus, it is possible to know on which days a sparse or dense flow of vehicles is expected. PTNs, on the other hand, are much more predictable networks. Although they vary in the same way as VANETs, they have greater predictability, as it is possible to know the path of each vehicle on the network and in what period this will occur.

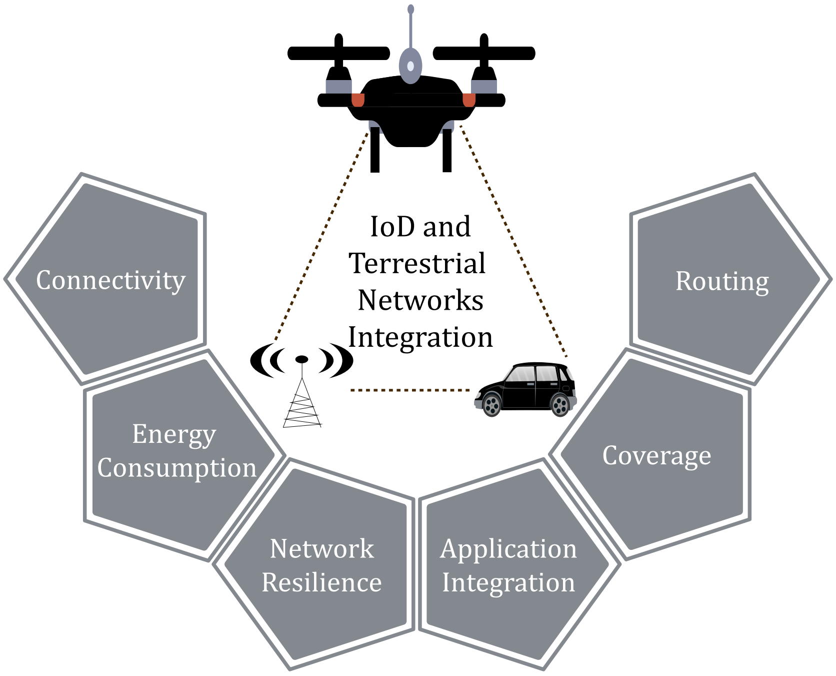
The connectivity of an IoT depends on the application and the type of node involved. It is possible to build IoT networks with a plurality of node types. If nodes are static, connectivity will be more predictable. On the other hand, if nodes are mobile, connectivity will depend on factors such as node speed, the number of nodes, and their area of operation to determine whether or not they are connected. In the case of IoD, network connectivity is not predictable as nodes move quickly and can connect and disconnect frequently. However, considering a scenario in which drones provide services, such as delivery, and there is a continuous flow of drones, it is possible to predict the behavior of the network over time, using similar characteristics of VANETs, such as time of day and day of the week.

Considering the characteristics of each network, the integration between them can increase connectivity by improving communication performance, particularly in cases where one node cannot reach another due to a sparse network, node disconnection, or failure. In addition, each network has different coverage. For example, VANETs are present on streets, avenues, and highways. PTNs are also present on streets and avenues. However, they depend on the time of day to function. IoD, probably will have better coverage in urban areas,as there is a greater demand for services (such as deliveries). However, IoD can have drones providing services in isolated applications. For example, drones can monitor crops and animals in the field. Similarly, in urban centers, the coverage is typically more significant for IoT due to having several applications working in the same space. Still, IoT is suitable in several other scenarios, such as agriculture and industry.

Another important factor in integration is energy consumption. The fact that networks such as VANETs, and PTNs do not have notable energy restrictions can improve the performance of the IoD, which has energy as one of the main limiting factors. Resilience is another crucial point in the integration between different networks. The IoD's flexibility allows it to cooperate with distinct networks to maintain resilience in adverse situations like disaster scenarios. In some cases, other networks can also contribute to the resilience of the IoD. Figure 2.6 presents the benefits that the integration of IoD and TNs can provide for both sides.

The integration can help the networks to have better coverage and connectivity. Consequently, other features like data delivery and routing will be improved. IoD can help with the integration and automation of other network components. For instance, in VANETs, IoD can contribute to the automation of roadside units, and in PTNs, IoD can automate bus stops. IoD can allow an environment where it is possible to deploy applica-

Figure 2.6: IoD and TNs partnership benefits



Source: Elaborated by the author

tions that use information from multiple networks and improve existing applications. The following details the particularities of IoD integration with VANETs, PTNs, and IoT.

2.6.2 Vehicular Ad Hoc Networks

Recently, the academic community has dedicated efforts to developing trust, reliable and secure VANETs [115]. In those networks, vehicles communicate with each other and with the infrastructure to provide security-related (e.g., accident and road conditions warnings) and non-security-related applications (e,g., information and entertainment) [115]. IoD and VANETs can collaborate to make networks more secure and reliable. Below, we detail some cooperation possibilities.

- **Filling the Gaps:** Many applications in VANETs need reliable vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication. However, this has been a significant challenge due to topology instability. One way to overcome this challenge is to use drones to assist VANETs. Drones can serve as auxiliary nodes to maintain connectivity and increase network coverage in sparse vehicle networks, thus helping VANETs [110]. The same can happen in IoD. Drones can use vehicles or roadside units as network nodes to fill gaps in the network and maintain connectivity, improving data delivery, routing, and energy efficiency. However, there is an extensive research gap in this area to the best of our knowledge.

- **Traffic-related Drone Applications:** In some applications, a collaboration between networks can bring bilateral benefits, for instance, increased network coverage and connectivity. On the other hand, there are some scenarios in VANETs that IoD can better assist vehicular networks, such as applications related to vehicle traffic. In this case, drones can send information about what is happening on the network and help keep it secure and reliable. Drones can detect accidents and send messages to roadside units and competent authorities to get help quickly. Similarly, IoD can monitor congested roads and warn roadside units. Drones can also be suitable for monitoring infrastructure conditions and informing vehicles when a critical problem is identified. They can also be used as a parking manager and notify users of available parking spaces. Another application is to have drones working as smart traffic lights for intelligent vehicles and road traffic management, which can help avoid congestion.

2.6.3 Public Transportation Networks

PTNs are essential in urban centers. They are composed of buses, subways, trams, trains, and other types of transportation that can allow the movement of people in urban centers through public vehicles. This network is part of VANETs but has some unique characteristics. The main difference is that PTNs have a more predictable mobility model, so we can know the path and estimate a particular vehicle's location at a given moment. Thus, besides predicting how the network will behave, it is also possible to know its density and node connectivity.

Public Transportation Networks can benefit from IoD as VANETs do. In the case of traffic-related applications, besides the examples mentioned above, IoD can provide public transport vehicles with personalized information for this network, such as the number of people waiting for a bus at a particular point. Also, drones can warn PTNs of significant events, such as broken buses or accidents involving buses, and schedule extra vehicles. On the other hand, IoD can benefit in the following ways:

- **Vehicles as Landing Stations:** Vehicles can serve as landing stations so drones can increase their coverage area [112]. In drone delivery, for example, drones can have their routes planned considering the vehicle routes belonging to PTNs. For instance, a drone that needs to make a delivery in which the complete path (round trip) is 30 km and only has the energy capacity to fly 20 km can make part of the route by landing on top of a bus, and save energy to make the delivery possible;

- **Vehicles as Network Nodes:** Vehicles belonging to PTNs can serve as additional nodes filling gaps where drones cannot communicate directly with each other. For example, considering that two drones need to communicate and one cannot reach the other, a vehicle belonging to the PTN can work an intermediate node option to enable communication. Due to the predictability of the nodes in PTN networks, it is possible to design the drone's path planning using the information from the PTN scale and routes.

2.6.4 Internet of Things

IoT aims to connect anything anywhere at any time to provide any service [113]. As drones in the IoD connect with the Internet, IoD is part of the IoT environment. This section discusses how non-drone IoT networks can collaborate with IoD.

- **Gathering Data:** IoT and IoD collect data for different applications [35] in different contexts, such as agriculture, environment and transportation monitoring, and smart city sensing. Some examples are animal monitoring, smart home surveillance, assisting police officers, and air quality sensing, to name a few [35]. Two forms of collaboration are possible: collecting the same information in distinct ways or collecting different information for the same application. For example, IoT and IoD networks can collaborate to perform a city monitoring system. Suppose one of the monitored elements is air quality. It is possible to have a fixed IoT structure with high monitoring demand at strategic points, such as industrial regions. Drones can collect data from other locations that do not need frequent monitoring. After, all the air quality data collected is grouped, processed, and analyzed in the cloud. On the other hand, we may have IoT devices collecting data on air quality and drones collecting information about other factors, such as noise pollution and vehicle traffic. This collaboration could create a city-sensing system;
- **Energy Transfer:** Drones have limited energy capacity. This point is also a common problem in IoT applications where devices use batteries. For instance, IoT networks in places with little or no energy infrastructure, such as monitoring rivers, crops, animals, and forests. Drones can transfer energy to IoT networks that do not have high energy capacity prolonging the network lifetime and even, in some cases,

providing energy autonomy [114]. On the other hand, using the IoT to deploy Drone charging stations is possible. In urban centers, a possible location for deploying wireless power transfer stations to charge drones is over buildings or buses. PTNs, IoD, and IoT would form a hybrid network in this case. In IoD, an infrastructure for recharging drones during missions can be crucial to allow autonomous applications such as constantly monitoring security, traffic, environment, and other factors;

- **Network Nodes:** Like other collaborations between Drones and TNs, drones can serve as additional nodes to fill gaps in IoT networks and vice versa. The cooperation has the potential to maintain network connectivity, increase data delivery, and allow better routing.

2.7 Final Remarks

This Chapter introduced the basics of IoD and discussed how to design protocols for IoD. We discussed the motivation to develop specific protocols for IoD. We highlight the challenges and potential of integrating routing protocols from IoD and other networks. In addition, we discuss important aspects of IoD protocol design and how their layered architecture is different from typical wireless networks.

Chapter 3

An IoD analysis through UC concepts and applications

This Chapter discusses how Urban Computing can help understand the IoD challenges through IoD applications for the urban context. In Section 3.2, we highlight how Urban Computing relates to the IoD. In Section 3.3, we propose a framework for IoD applications in the context of UC. Section 3.4 presents the applications related to IoD-UC considering the following topics: urban planning, transportation systems, environment, urban energy consumption, social applications, economy, and public safety and security. Finally, we give a final discussion in Section 3.6.

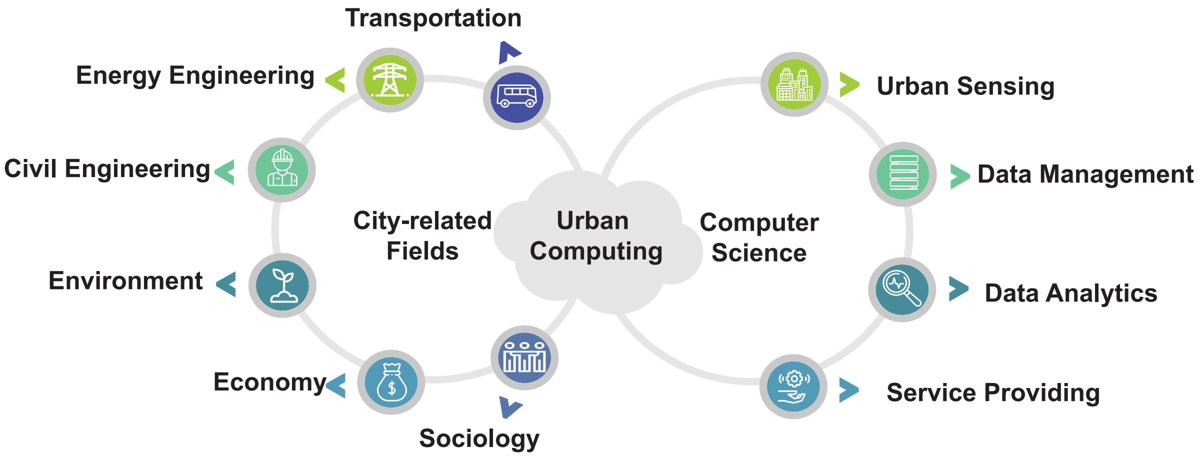
3.1 Introduction

Urban Computing (UC) represents the encounter of city-related fields such as transportation, civil engineering, economy, environment, energy engineering, and sociology with computer science [8, 116]. UC is based on computational tasks such as urban sensing, data management, data analytics, and service provisioning. It seeks unobtrusive and continuous improvement of people's lives, city operation systems, and the environment [8, 117]. Figure 3.1 gives an overview of the urban computing concept.

An important part of urban computing is data. Different data sources can be used for UC applications such as geographical, traffic, mobile phone signals, communicating, environment monitoring, Location-Based Social Networking (LBSN), economy, energy, and health care data. Zheng et al. [8] defined seven categories of urban computing scenarios, detailed in the sequence in which this data can be used.

- **Urban planning**: Applications related to the process of analyzing a set of factors such as human mobility, traffic flow, and the existing urban structures to plan and/or improve the urban scenario;

Figure 3.1: Urban Computing Concept Overview



Source: Elaborated by the author

- **Transportation systems**: Applications associated with the process of improving driving experience or transport services, both public (e.g., buses) and private (e.g., taxis and transportation apps);

- **Environment**: Applications linked to environment monitoring, such as waste, air, water, and noise;

- **Urban energy consumption**: Applications connected to the purpose of monitoring and consuming energy in a conscious, sustainable, and efficient way;

- **Social applications**: Applications to understand a society of a group of people. It seeks to understand people's lifestyles, the way they relate, and based on the location of the service they use, as well as when and what they consume;

- **Economy**: Applications that can predict or influence the economy of a place through human mobility data. For instance, determining an ideal locale for a business, indicating the trend of a stock market, and helping discovered factors that determine the value of the real estate;

- **Public Safety and Security**: Applications that help in public safety and security and also prevent it. Human mobility data can help police actions, disaster managers, and traffic control, to name a few.

3.2 Connecting IoD and Urban Computing

IoD can contribute to Urban Computing goals. UC needs data to understand the population and make people's lives more comfortable, efficient, and productive. The services provided and data collected by drones can contribute to leveraging the research related to UC. Each application must be analyzed independently and studied if it is more advantageous if carried out using drones. For instance, the collection of air quality data can be performed by static sensors and/or sensors coupled in various mobile devices such as cell phones, cars, and drones. Thus, one should consider whether using drones in this application is advantageous. Some advantages of using IoD are:

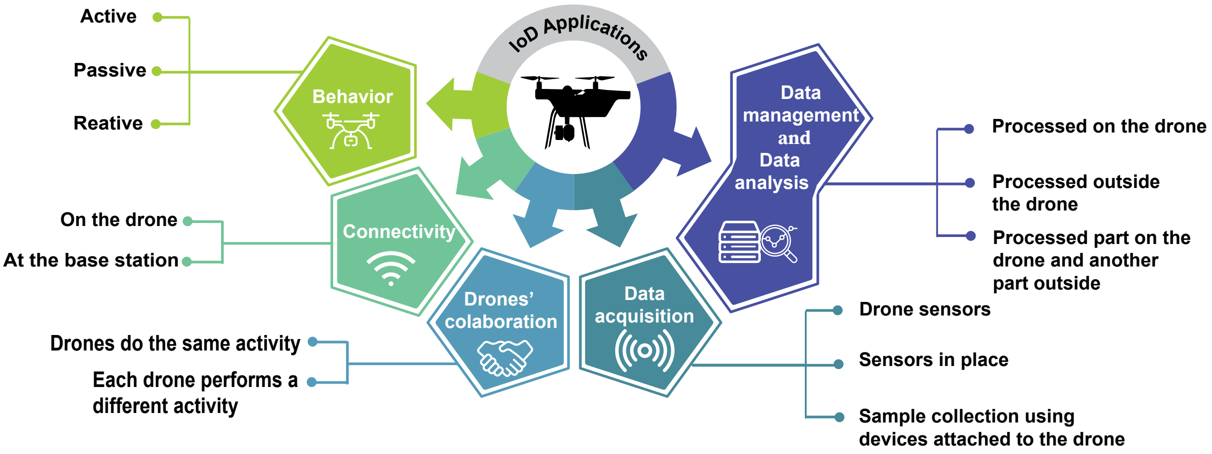
- **Cost**: considering an IoD structure, the Internet of Drones can have less or no additional cost;
- **Speed**: IoD can perform tasks and fulfill objectives faster than other devices or people;
- **Flexibility**: drones can fly at different altitudes and possibly at any time;
- **Real-time data**: IoD can gather real-time data;
- **Aerial data**: Drones can collect aerial data with flexibility, speed, and coverage that are difficult for other devices to obtain.

On the other hand, UC can contribute to IoD. IoD can benefit from more consolidated structures and data in urban computing to save cost and energy. An example is integrating drones with other networks and vehicles, such as collaboration between drones and buses used for public transport, which aims to save energy from drones in delivering goods [112]. Thus, data already used in UC, such as climate and urban mobility data, can be useful and contribute to services provided by drones and help in the planning and implementing of IoD. Hence, the connection between IoD and UC is a gap to be explored in the literature.

3.3 The Proposed Framework for Urban Computing in IoD Context

This section presents an overview of IoD applications in the UC context. UAV is an emergent technology that has many possibilities for applications. For example, applications can differ for the following reasons: how the drone is used in the application; how to connect drones to the network; different forms of collaboration and division of tasks; different ways of acquiring data; and different ways of analyzing and managing the data collected. Figure 3.2 shows the ways to develop an application in this context. In an IoD network, there are three forms to use drones in an application, namely the active, passive, and reactive forms. In the sequence, each of them is detailed.

Figure 3.2: IoD applications overview in an urban computing context



Source: Elaborated by the author

- **Active Applications**: Active applications are those that the drone uses uptime to achieve a goal. In these cases, the application's objective is already completed after the end of the operating time. UC can benefit from the application in this category using the data collected during the uptime and the drone trajectory data. For example, in the delivery of goods, the objective is delivery. The operating time of the drone is mainly to get from one point to another. It is reasonable to assume that some process of internal data analysis occurs in uptime. However, after the mission, the main objective has already been achieved. However, the resulting data, such as the drone trajectories, are convenient for other applications unrelated to the original goal (delivery of goods), such as identifying and inferring patterns about a region;

- **Passive Applications**: Passive applications are those the goal is only accomplished by an external system and usually after the drone uptime. The main objective of this type of task is to collect data. The drone uses its sensors to collect data, but the ultimate goal is reached only after its operation is complete (e.g., monitoring noise pollution in a city). During uptime, the drone collects data. However, further analysis is needed to obtain information about city noise. This kind of application commonly needs data analysis with a high processing cost that is unsuitable to perform on the drone. This scenario provides a high amount of data that is suitable for UC to improve people's lives;

- **Reactive Applications**: Reactive applications are those in which drones react to an event that appears during uptime. For instance, the drone can search for an object, animal, or person and needs to take action when some of these options are found. The main objective of this type of application is finding an event and reacting to it. One example of this is one search and rescue mission. The drone uses the time in the operation to look for a specific event, which, in this case, is to find a victim, and when this occurs, it needs to have some appropriate reaction (e.g., notifying the authorities). UC can benefit from this application to provide real-time scenarios of fields such as urban traffic and police missions. Thus, events that need immediate treatment can fit into this category, such as vigilance, traffic monitoring, police chase, and disaster scenarios, to name a few.

Applications are distinct by the way they connect to the Internet. The connection can be through the base station or each drone connected to the Internet. In addition, the collaboration between drones in the IoD may be different. Each drone, or group of drones, may be doing various tasks. For example, in an urban sensing system, some drones can collect data from the air while others from noise [65]. Another form of collaboration is for all drones to perform the same task. Thus, a larger area is covered more quickly.

Another issue is how data is collected. The most common way is for drones to collect data using their sensors. However, the drone may only get information from sensors installed in the place of interest. Another way is when the drone takes a material sample, and the analysis occurs in a suitable laboratory. This option is used for water analysis [118]. Finally, there are also different ways to process and manage the information collected, including:

- **Processed on the drone:** the data is processed on the drone before being transmitted. For example, Gu, Michanowicz, and Jia [119] use a companion computer to make the processing;

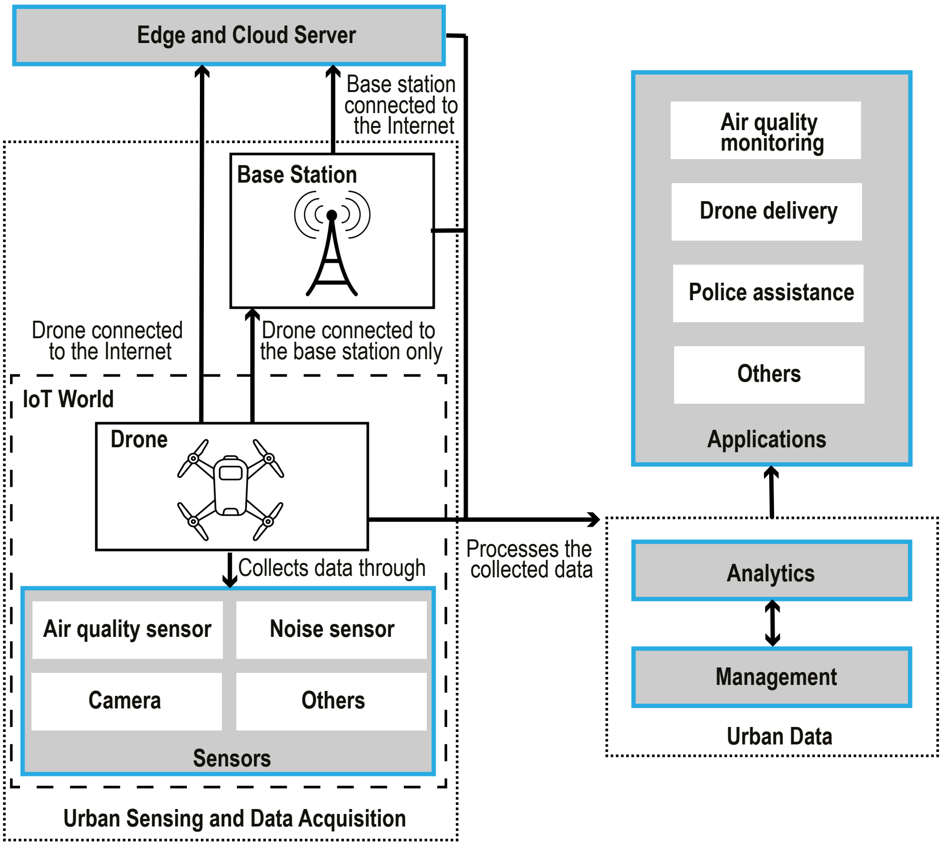
- **A processed part on the drone and another portion outside:** the drone pre-processes the data before transmitting it. For instance, Wu et al. [65] process the raw data in knowledge items that were further processed for an external system;
- **Processed outside the drone:** the drone collects the data and stores it for later analysis, or the drone transmitter the data in real-time for the base station or directly to the cloud for external processing. For example, Ghosh et al. [120] transmit data from the drone to a HUB using a module nRF24L01. Then, using a WiFi module, the HUB routed the data to the cloud to be processed.

Considering the classification of applications in Figure 3.1, in Figure 3.3 we propose a general framework for UC in the context IoD. IoD must provide a secure environment for drones to access airspace simultaneously and in a coordinated manner. Gharibi et al. [5] architecture divides airspace control into zones and each zone has one or more Zone Service Provider (ZSP). A similar approach can also be used in urban computing, dividing the city into zones. Data acquisition from each zone can be grouped as needed for each UC application. A control base in each zone can also be responsible for controlling and maintaining drones. The first step is to perform urban sensing and data acquisition. In this stage, drones can collect data using different sensors. It is common and expected to have a UAV with more than one type of sensor. For instance, cameras, air quality sensors, temperature sensors, and noise sensors, besides those related to its flight (e.g., speed, altitude, direction).

The drone is connected directly to the Internet or a base station with Internet access. As some drones can perform artificial intelligence tasks, machine learning techniques can further leverage the contribution of IoD to UC. Specifically, drones can perform real-time image processing. For instance, UAVs can monitor vehicle traffic and immediately inform the responsible authorities of an emergency scenario, such as a car accident. Also, drones can collaborate with vehicles predicting road congestion. For example, UAVs can detect dangerous regions and inform vehicles if a road is blocked due to natural matters like mudslides caused by rain.

The processing of urban data involves management and analytics. In the former activity, the data is organized, whereas the latter transforms data into knowledge [8]. In many cases, it is common for data to be managed and then analyzed. For example, typical management is data cleaning, to delete incomplete data before analysis. However, this order can be variable, and some applications may require a different sequence of data manipulation. Thus, as shown in Figure 3.3, it is possible to alternate between these steps. Finally, in the last layer, we have the IoD applications.

Figure 3.3: General framework for Urban Computing in IoD context



Source: Elaborated by the author

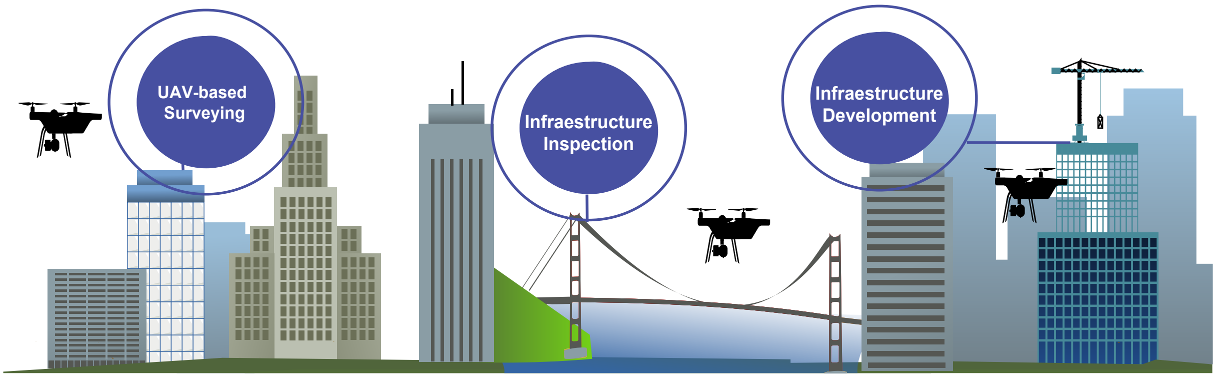
3.4 IoD-UC Applications

In this section, we discuss IoD-related applications and their contributions to UC. This section explores the following categories of applications urban planning, transportation systems, environment, urban energy consumption, social applications, economy, and public safety and security. We select some representative examples for each category.

3.4.1 Urban Planning

An essential step in building a smart city is planning. This task involves many factors, such as human mobility, traffic flow, urban infrastructure, and social issues [8]. Urban planners must collect data through traditional surveys to decide, which requires intensive labor. This task was facilitated by technological advances, increased monitoring, and the gathering of human data generated in urban spaces. Recently, UAVs have brought a new perspective to this scenario, an aerial vision. Figure 3.4. illustrates some applications that typically contribute to the UC environment.

Figure 3.4: Typical drone applications for urban planning



Source: Elaborated by the author

Urban planning applications have three main objectives: (i) map urban areas; (ii) study different factors that can be detected using relief and soil. These models can help, for example, in the generation of models for preventing landslides and floods.; and (iii) quickly detecting some imminent danger with constant monitoring. For instance, differences in soil and structures (e.g., buildings, bridges, houses), being able to prevent landslides and detect possible maintenance needs. Some typical applications for drones in urban planning are:

- **Urban UAV-based Surveying**: Urban UAV-based Surveying has many applications in the civil engineering environment. The most typical applications of UAVs in this field are the determination of a 3D model of various objects, Digital Terrain Model (DTM) Digital Surface Model (DSM), Digital Orthophoto plan [121], and Digital Elevation Model (DEM);

- **Infrastructure Inspection**: Drones can carry out the inspection of infrastructure in the city, collecting data for analysis. With this data, for example, the drone can detect damage to the infrastructure [122]. In addition, it is possible to generate 3D models that allow generating data, for instance, for restorations and maintenance;

- **Infrastructure Development**: Similar to infrastructure inspection, drones can be suitable for monitoring the construction of new engineering projects. The team can store data on the progress of the infrastructure and even allow those responsible to monitor the project remotely [123].

UAVs can be flexible, fast, accurate, inexpensive, and safe tools for geospatial and surveying activities in smart cities [124]. The UAV's goal in the mission is to gather data. These applications are generally passive, i.e., they are part of larger objectives, and the UAV is responsible for data collection only. In the case of this application, it is typical to collect photos and, in some cases, record videos. According to recent studies on UAV-based surveying in urban areas [121, 125, 126, 127], it is common to use just one UAV in this application. Besides that, some studies use an autonomous flight, a system necessary for path planning. On the other hand, other works use a manual flight with a certified flight operator [125], mostly when the place may have people present during the flight time.

When the UAV completes its mission, all collected data are transmitted to an external system for the processing and analysis stage. High image processing is usually required in this situation, which also involves computer vision and machine learning techniques. The UAV is typically a way of collecting data in these applications. UAVs give an efficient and cost-effective survey tool [128], providing an opportunity for different applications that require 3D models, DEMs, DTMs, DSMs, and orthophotos in urban scenarios.

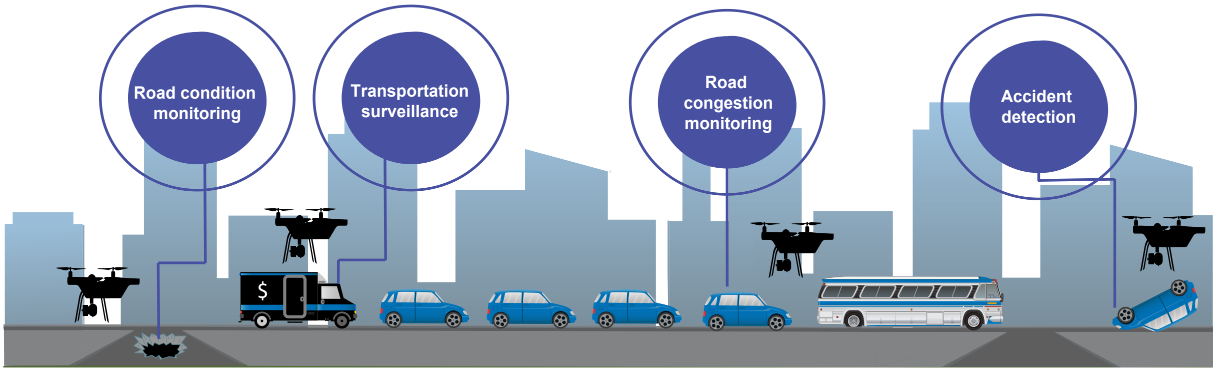
These applications and the data collected through them can be helpful to UC. With IoD, applications can evolve from a single drone to multiple drones. Specifically, these applications can allow UAV-based surveying, infrastructure monitoring, and other applications related to urban planning to be more intelligent, integrated, and efficient. For example, the city may have a group of drones on the network responsible for regularly monitoring public infrastructures, such as bridges and viaducts. This monitoring can be more frequent in the rainy season, always aiming at the safety of citizens. Drones that perform other roles in the city can carry out this programmed monitoring according to their location and energy. In addition to the urban applications explored in this section, UAV surveying also has been used in other fields such as agriculture [129], underground mine cavities[130], volcano-tectonics [131], and archaeology [132].

3.4.2 Transportation Systems

The aerial view of the drone is an ally in applications related to the control, monitoring, and improvement of urban traffic. Specifically, this drone feature allows a different view from other types of monitoring, such as security cameras, data from the cars themselves, and data from the car users (cellular data, like GPS). In addition, transport systems automation cannot be achieved by automating only vehicles [111]. It is necessary that other system components, such as traffic police, road surveys, and rescue teams, also need to be automated. Some suitable applications for drones in this context are illustrated in

Figure 3.5 and detailed below:

Figure 3.5: Typical drone applications for transportation systems



Source: Elaborated by the author

- **Road Congestion Monitoring**: Drones can identify congestion and the reasons for the congestion, as well as possible solutions to improve traffic [133];
- **Road Condition Monitoring**: Drones can contribute to regular road maintenance by identifying stretches that need repair [134]. They can also identify damage in bridges and viaducts, as presented in Section 3.4.1. In addition to these issues, drones can monitor other elements that are part of the transportation system, such as bus stops and pedestrian walkways;
- **Accident Detection**: Similar to congestion identification, drones can identify accidents. Often, accidents can be the cause of congestion. In this case, in addition to seeking solutions to congestion, the drone can be helpful in the rescue team at the scene of the accident and help the team to reach the accident scene within the shortest time [111];
- **Transportation Surveillance**: Security can also benefit from drone applications. Drones can monitor specific cars and special operations, such as transporting authorities. They can serve as a support to the security team, identifying possible threats and problems;
- **Road Weather Monitoring**: Weather elements such as wind, rain, fog, and storms exert a significant impact on road transportation, leading to speed and capacity reductions [135]. Current weather service systems are limited in their ability to offer precise local weather nowcasting due to the predictive nature of their processes and the absence of real-time data from the atmospheric boundary layer. Drone-based mobile automatic weather stations can play a pivotal role in enhancing road weather management by delivering accurate and up-to-date atmospheric information.

Transport-related drone applications have many examples of situations in which it requires a reaction from the drone. In congestion, infrastructure, accidents monitoring, and transportation surveillance, it is expected that the drone reacts to some event, informing the responsible authorities of the data collected. Thus, in this case, applications tend to be reactive. Due to the characteristics of these applications, the most used sensor is cameras. Another point is that, due to the need to react, it may be usual for the drone to perform at least a pre-processing of the collected information.

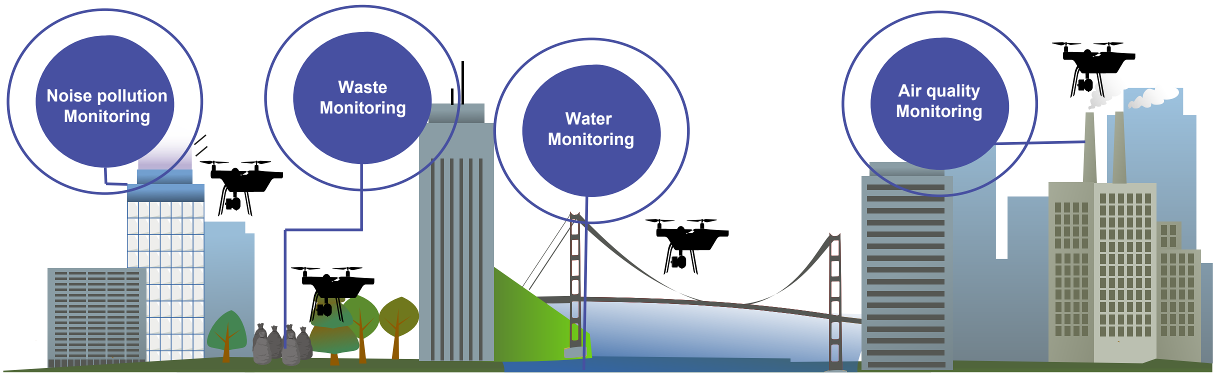
In the literature, some studies seek to study the use of multiple drones in the monitoring of traffic [136, 137, 138, 139]. The mentioned studies do not clarify if there is communication between drones and how it is done. Thus, despite the academy's interest in studying drone networks related to transportation systems, the area is still in its beginning. Wu et al. [65] [34] present examples of communication between drones. In this study, different swarms cooperate to perform urban sensing. Each UAV swarm is responsible for collecting data in a region, and the communication between them allows the aggregation of information and the disposal of redundant information.

Transport-related drone applications have many examples of situations requiring a reaction from the drone. In congestion, infrastructure, accident monitoring, and transportation surveillance, the drone is expected to react to some event, informing the responsible authorities of the data collected. Thus, in this case, applications tend to be reactive. Due to the characteristics of these applications, the most used sensor is cameras. Another point is that, due to the need to react, it may be usual for the drone to perform at least a pre-processing of the collected information.

3.4.3 Environment

Cities consume different resources to produce materials and energy [140]. The consume-produce process on a large scale can generate diverse kinds of pollution in the city, such as air, noise, waste, and water pollution. Besides that, this process can affect both regional aspects, such as human health, agricultural/ecosystem productivity, and global change issues, like climate, ozone depletion, and oxidative capacity [140]. Added to these factors, the growing increase in the world population and urban agglomerations make pollution issues in urban centers increasingly worrying. Considering these issues, the academic community has been studying ways to monitor the environment on an ongoing basis. Drones contain several sensors and travel great distances and, for this reason, are adequate for environment monitoring. Figure 3.6 presents the main applications considered.

Figure 3.6: Typical drone applications for environment



Source: Elaborated by the author

Monitoring the environment in urban areas has three main objectives: (i) comply with the legislation – many countries have targets for reducing polluting gases, noise, and cleaning up rivers. Constant monitoring can help to meet the objectives. (ii) studying the effects of pollution on health and the environment – monitoring can help understand the relationship between diseases and changes in the ecosystem with air, noise, and water pollution. (iii) detecting some imminent danger – constant monitoring can help to detect, for example, a flammable gas leak. Some typical applications for drones in the environment are:

- **Air Monitoring**: Drone applications for quality monitoring have two approaches. The first and more common one deals with using sensors to measure air quality, usually with three objectives: comply with the legislation, study the effects of pollution on health and the environment, and detect quickly some gas that has imminent danger. The second one uses aerial images to identify possible causes of air pollution;

- **Noise Monitoring**: Drones offer a viable option for urban noise monitoring [141]. In this context, investigations aim to discern the acoustic emissions from the drone. Employing digital processing techniques, these studies endeavor to segregate the drone-generated noise from the ambient acoustic environment, thereby enabling the accurate identification of authentic urban noise sources [142];

- **Water Monitoring**: Drones have the potential to perform this kind of task in two ways: the first consists of using the camera to use aerial images to determine possible pollution sites or trying to identify water quality through these images. The second is to use some device attached to the drone to collect water data in the place or to carry a sampling to analyze (water sampling payload);

- **Waste Monitoring**: Drone applications for waste management include identifying potential illicit waste disposal areas via aerial imagery. Additionally, drones find utility in administrating and supervising municipal solid waste landfills and dumpsites [143].

It is relatively common to find in the literature studies that use only a drone to monitor some factors related to the environment in the city [119, 120, 144, 145].

Generally, these systems use commercial drones with some specific sensors for the intended purpose, such as gas sensors for air monitoring and noise sensors for sound monitoring. These applications have the most sensor variances. In these simpler applications, drones are manually controlled, or a system is used to plan the drone's path. Therefore, communication in these cases is also relatively simple as it generally uses a 2.4/5.8 GHz transmitter to establish communication between a base station and the drone. However, researchers have dedicated themselves to creating systems with more complex communications to control drones to monitor the environment in urban spaces. Table 3.1 summarizes these studies.

Table 3.1: Recent studies related to environment

Ref.	**UAV objective**	**Theme**	**Information processing**	**Highlighted**
[65] (2017)	Collect data and disseminate information to other drones	Air, Noise, and Traffic	Data are processed by the drone into a set of knowledge items. The data can be stored in the drone memory for further analysis or transmitted through a cellular tower for real-time use	A holistic middleware solution for drone swarms executing urban sensing tasks
[146] (2018)	Collect data and disseminate information to other drones and the base station	Air	The drone transmitted data to the base server, which processes the data. The acquired information is stored in an online database and it's used to create a pollution map.	A pollution monitoring system, with two autonomous drones
[147] (2020)	Collect data and disseminate information to the monitoring station	Air	Each drone is sent to a different location to collect air data. The acquired information is transmitted to the monitoring station and it's used to create an air quality map.	Multiple drones are used to generate Air Quality Health Index map

We can identify the two types of drone collaboration by observing Table 3.1. In the first type, different drone swarms collect data about more than one environment element and are still working together[65]. In the second one, the drones have the same goal: collecting air data [146, 147]. Usually, applications related to the environment have two main goals:

- **Gathering data**: the most common drone use in this context. The data are collected in different ways, either by sensors of the drone or sensors already installed in the place of interest or utilizing samples collected through devices attached to the drone;

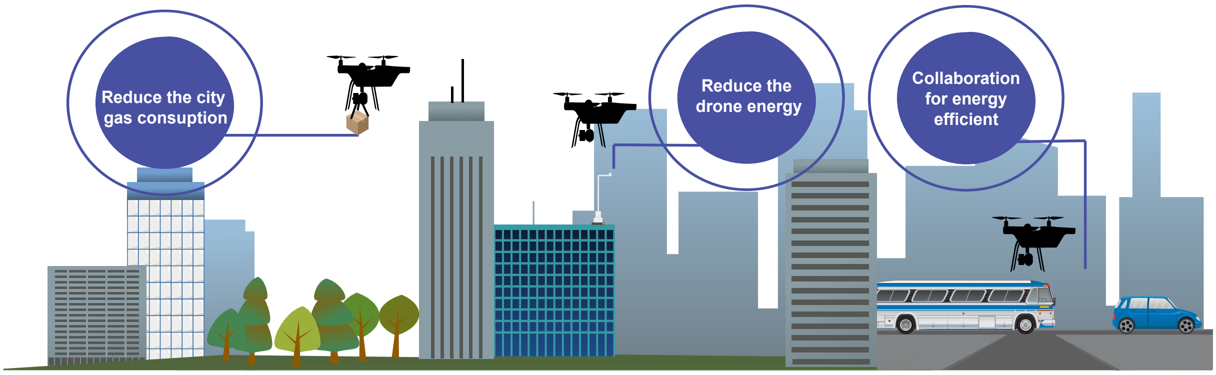
- **Search for possible pollution sources**: the drone identifies and/or tracks potential illegal discharges of pollutants and waste into the water, the air, and the soil.

Environmental applications are generally classified as passive applications. The drone only collects the information and passes it on for storage. Environment-related drone applications seek to make more intelligent sensing of the city when compared to projects that use only one drone. The development of smart cities must be linked to environmental protection in urban landscapes. Increasingly, researchers and officials are well-aware of natural and man-made hazards in cities. Protecting the environment while modernizing people's lives is of supreme importance in UC [8], and IoD has the potential to help these applications become more efficient and integrated. In addition to this issue, protecting urban ecosystems improves people's quality of life by fighting and preventing diseases.

3.4.4 Urban Energy Consumption

Due to the rapid progress of urbanization, there was an increase in demand for energy consumption, requiring technologies that can detect consumption, improve energy infrastructure and reduce energy use [8]. IoD can play an essential role in advancing technology to achieve part of these goals, considering the approaches illustrated in Figure 3.7. Each one of the approaches is detailed in the sequence:

Figure 3.7: Typical IoD approaches to reduce urban energy consumption



Source: Elaborated by the author

- **Reduce the city's energy and gas consumption**: As discussed, one of the most promising applications is the delivery of goods by drones. Considering the current technological scenario related to drones, if carefully deployed, drone delivery can be more advantageous in terms of gas emissions and energy use in the freight sector than delivery by trucks [148];

- **Reduce the drone energy consumption**: Initial research shows that there are scenarios where the drones use can be more advantageous in terms of energy [148]. However, there may be scenarios in which drones use is not recommended due to their battery limitation. In addition, the energy issue is a significant limitation to be overcome due to its implications on flight time, influence on the success of missions, and safety for the population. In the case of swarms, perpetual flight, when a UAV swarm has enough energy to fly forever, has been a subject of study to overcome the flight time limitation barrier [149]. Nevertheless, it is also necessary to investigate the reduction in drone energy consumption, that is, to fly longer with less energy or even study sustainable battery recharging sources. Decreasing drone energy consumption helps to reduce the city's energy consumption as a whole;

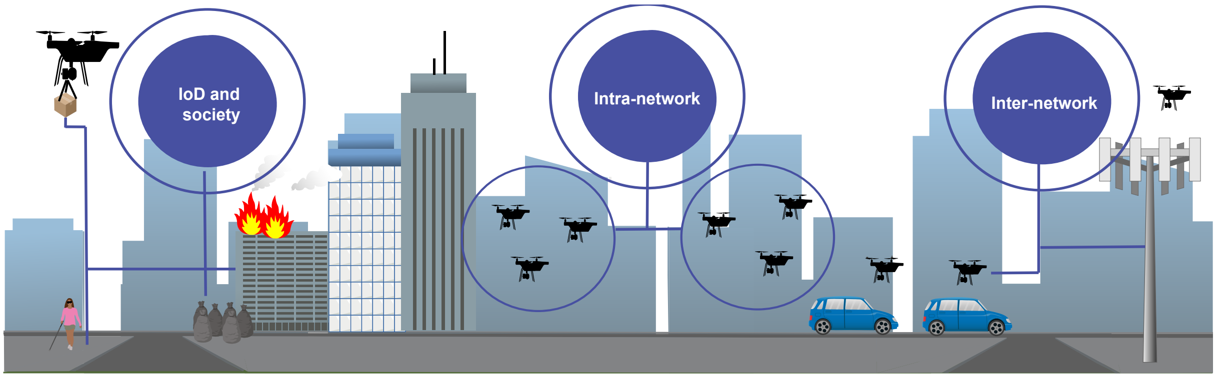
- **Drone collaboration for energy efficiency**: The drone can collaborate to increase the energy efficiency of other devices. For example, drones can collaborate to enable green IoT, which minimizes energy consumption in IoT devices [150]. These devices need a high transmission power to send information over long distances. The drone can physically go to a location close to the IoT device to collect, process, and transmit data, reducing power consumption [150]. A different approach is proposed by Mitcheson et al. [151], in which the authors present the end-to-end energy-autonomous sensor systems. This system consists of one or more drones with devices capable of transferring inductive energy. In this initial research, the drone can transfer energy to sensors by collecting data via a coil attached to the UAV that receives power from a charging platform. Another point of view is to use collaboration to conserve the drone's energy, increasing its efficiency. An example is using a strategy where drones hop between public transit vehicles (e.g., buses and trams) to increase the drone's flight range in applications that drones have to deliver packages [112].

Urban energy consumption-relation applications are a point in the relationship between drones and urban computing that only makes sense when a drone swarm that works in favor of urban space exists. In addition, energy efficiency can be improved when you have good resource management. This fact can be achieved through IoD, which seeks to connect different drone swarms services and applications in the urban space.

3.4.5 Social Applications

The social aspect of IoD can be discussed considering two perspectives: IoD's relationship with other networks and the drone's relationship with people. Also, from the first perspective, it is possible to discuss from inter-network, that is, the collaboration of the IoD with other types of networks and intra-network, with which the subnetworks IoD collaborate. These approaches are illustrated in Figure 3.8 and will be discussed in the following sections.

Figure 3.8: Typical IoD approaches for social applications



Source: Elaborated by the author

3.4.5.1 Inter-network

IoD can aid other networks and provide resilient scenarios for urban environments. Communication between different types of networks can contribute to balancing services, prioritizing what is most advantageous for each network. Such a system can lead to energy savings making the city more sustainable while becoming intelligent. Some possibilities of internetworks explored in the literature are:

- **IoD-assisted VANETs**: IoD can collaborate with VANETs in two ways. First, drones provide services related to the transportation system, such as the applications mentioned in Section 3.4.2. Second, drones can form hybrid networks with VANETs and provide, for instance, safety message broadcasts, ubiquitous internet access, and additional spectrum provision [152]. Drones can help overcome some difficulties found in VANETs, such as high node mobility, frequent path failures, and various network obstructions [153];

- **IoD-aided IoT**: IoD is part of IoT. Also, hybrid networks with different types of mobile nodes (terrestrial and airborne) and static nodes can be formed, seeking greater resilience, efficiency, and energy balance. This is a comprehensive area explored by various studies [12, 111, 124, 150, 154]. Each has different needs related to drones, depending on each application, objective, and complexity. However, it is possible to highlight some general points that need advances, such as security (hardware and software), the drone's capacity and energy efficiency, and the management of hybrid networks efficiently;
- **IoD-enable resilient communications**: IoD can play a crucial role in smart cities related to the resilience and overload of other networks. In disaster scenarios, communication is likely to be affected and IoD can fill the gap and allow essential services to be maintained. Specifically, the IoD can quickly restore limited connectivity in the system being substituted for the damaged by on-ground infrastructure [155]. Also, IoD can act as extra nodes in big events (e.g., soccer, concerts), maintaining network connectivity and quality.

The academic community has studied the integration between drones and VANETs in recent years [110, 156], focusing mainly on creating a routing protocol that may be suitable for this scenario. Although many protocols have been developed [110], most proposals do not deal with the D2D communication that may exist in this network. Some recent studies consider the communication between drones more directly routing protocols development [153, 157, 158]. Due to the difficulty of setting up a network with these specificities, the authors use simulators to carry out the experiments such as NS-2. However, they do not discuss drone networks' hardware and software requirements to act in this environment. Another point is that it is fundamental to deepen studies that would perform the management of drones in this scenario. For instance, consider each drone's coverage, network density, and uptime, which will demand recharging the battery.

3.4.5.2 Intra-network

IoD is formed by several drone networks, each with a purpose. Intra-network in IoD refers to the collaboration between these different drone networks. This collaboration can have different levels based on the context of these networks. For instance, several companies are predicted to start offering drone delivery services. In this case, there may be a collaboration between applications from different companies due to the service provided. Another type of collaboration is related to performing distinct tasks but with a common goal. An example of this scenario is the study of Wu et al. [65], in which different groups of drones perform tasks such as monitoring the air, traffic, and noise with a common objective: sensing the city.

Besides, in IoD, there is also a collaboration between all the active drones in the network. This fact is mainly in issues related to security and emergencies. Collaboration between different networks can also be significant for the energy balance of the network, making IoD more viable, intelligent, economical, and valuable to improve people's lives. However, this topic still needs to be explored in the literature.

3.4.5.3 Drones and Society

Drones have been gaining different roles in societies, from leisure applications, such as receiving a Valentine's Day message via drone [159], to more significant applications, such as the aid of drones to control the Covid-19 pandemic [160].

- **Disease control and combat**: IoD can help in pandemic scenarios to combat diseases. Some applications are public announcements of the government guidelines; detecting the disease symptoms through screening masses; spraying disinfectants or other suitable products; delivering medical supplies and other essentials; and crowd surveillance to identify a possible unwarranted situation [161]. Besides that, IoD can help fight diseases related to a specific cause. An example is using UAV images to identify potential breeding sites for the dengue mosquito in Brazilian cities [162, 163];

- **Healthcare and drone delivery**: IoD can improve the health system using mainly the drone's ability to perform fast transport. Some examples of situations in which IoD can collaborate are the transport of biological materials samples (such as blood for analysis); and delivery of blood, vaccines, medicines, organs, life-saving medical supplies, and equipment [164];

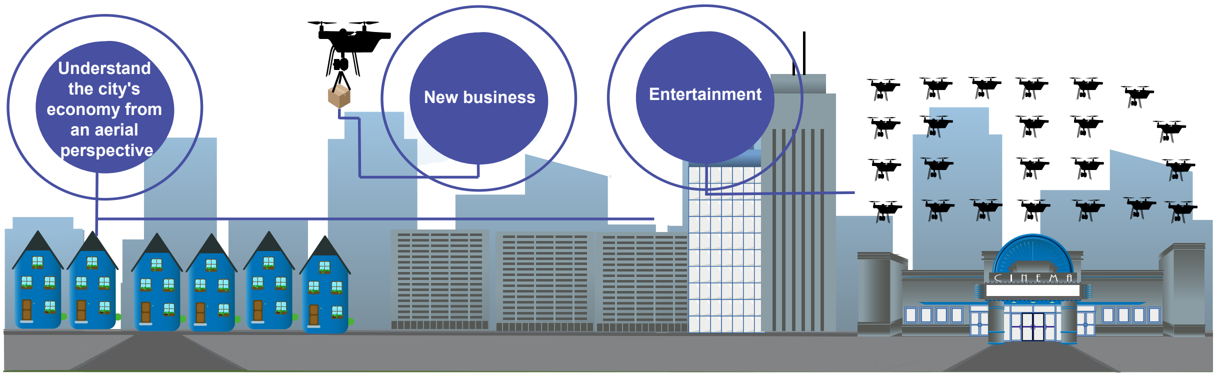
- **Assisted IoD**: Drones can also be used as assistive technology. Some studies, for example, seek to use drones to help the visually impaired [165, 166, 167]. However, each person has a drone that will guide them. The personal drone can be used for different types of assistance to people, adapting to each person's needs. For example, drones can notify the responsible authorities if they detect a person lying on the street. These personal drones can be linked to IoD and pass information more quickly. However, an approach that has not yet been explored in the literature is the creation of a network of drones, and not the use of personal drones, which can help assist in several cases, such as when an elderly person falls on the street or when someone experiences fainting. For applications like this, it will be necessary to discuss rules and define regulations, mainly considering ethics;

- **Social Media**: Social media (e.g., Twitter, Instagram) can report real-time events, especially during and after disasters. However, it is often necessary to separate accurate and false reports. Thus, IoD can be used to verify the integrity of the information posted on social media and provide a quick and effective response in emergencies [168].

3.4.6 Economy

Drones are considered a driving force behind smart cities, encouraging new businesses that can bring more jobs, improve productivity and optimize cities' resources [124]. Some crucial points of the economy and its relationship with IoD are illustrated in Figure 3.9. The most impactful change related to the economy is that it will be possible to understand it from an aerial view. Through the applications and the collected data, it will be possible to draw an aerial profile of the city. In addition, building an air mobility signature using drone traces will be possible. In the same way that we use urban mobility traces to understand society and improve people's lives, it will be possible to add to these analyses the perspective and data of drones.

Figure 3.9: IoD impact on the economy



Source: Elaborated by the author

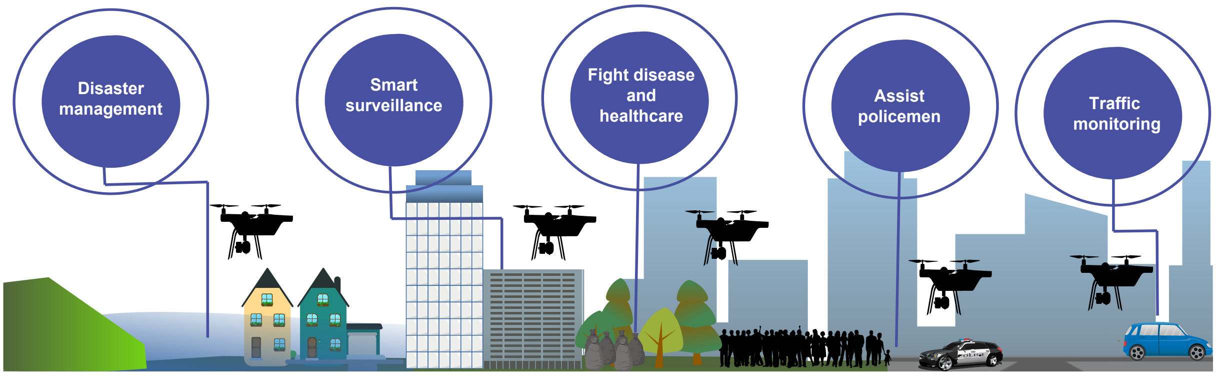
Also, the popularization of IoD can bring new applications that can affect the economy in different ways. For example, the most promising of applications, drone delivery, can change the delivery market. However, it will not completely replace the way we receive deliveries today. Some factors will contribute to drone delivery, such as cost, desired speed, and feasibility of delivering a particular product using a drone. Also, once the IoD is stabilized and providing services in the urban environment, it will be possible to check which services are used, which are on the rise, and which are not. Thus, it will be possible to understand the market of different sectors.

Another example affecting the economy is using drones in entertainment, tourism, and sports. With the popularization of drones, the film, tourism, and television industry has already been using drones to produce content from an aerial perspective for television and movies, and to broadcast sports games from different angles. With IoD, these applications could be connected to the network and bring more information and possibilities. For example, tourist attractions may have a drone network that assists tourists with information and guided tourism. Drones can also be used to attract tourists. For instance, a drone swarm was used in presentations in China to create a New Year's show [169]. This application could be the future of New Year's tourism, in a mixture of fireworks with drones or even their complete replacement.

3.4.7 Public Safety and Security

Safety and security are factors that most contribute to raising people's quality of life. Drones can include numerous sensors, which makes them helpful in monitoring essential indicators for public safety [65]. Many of the applications mentioned here are related to other topics discussed in the application section. This fact only shows the importance of this context and that the contribution of IoD can be highly relevant to the area. Some typical applications are illustrated in Figure 3.10, and details are in sequence.

Figure 3.10: Typical IoD applications for public safety and security



Source: Elaborated by the author

- **Smart surveillance**: Drones can intelligently monitor homes, businesses, buildings, and public places;

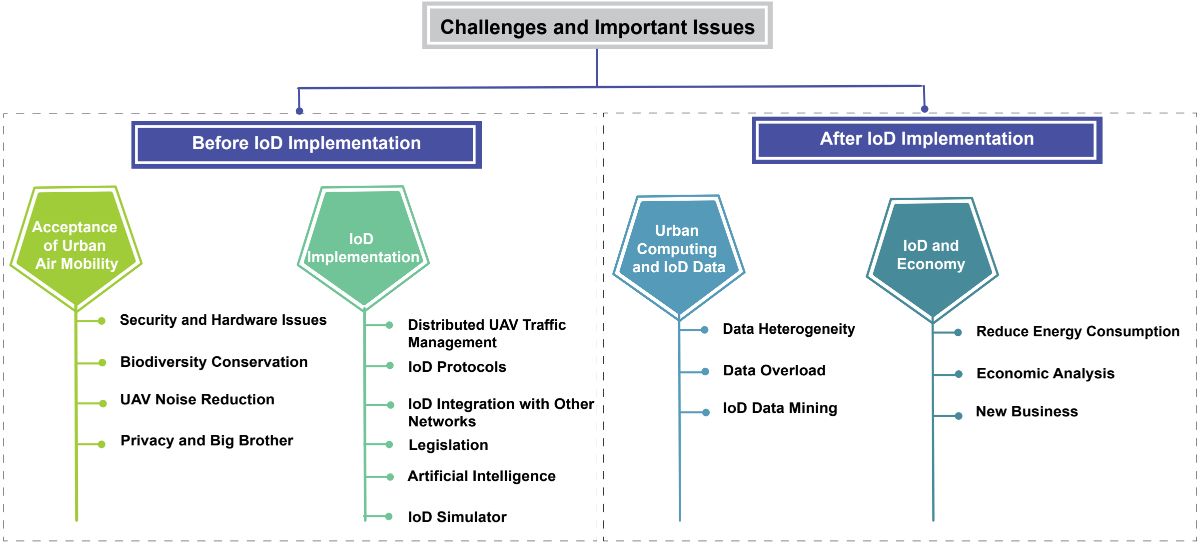
- **Disaster Management**: Drones can contribute in numerous ways to disaster management. Some instances are monitoring, forecast, early warning systems, situational awareness and logistics, damage assessment, standalone communication systems, search and rescue missions, and disaster information fusion [170];
- **Police assistance**: Drones can help the police in various situations, such as crowd monitoring (which can happen at big events), police chases, and transportation of important goods and authorities. In general, the drone can be used as police eyes, identifying possible breaches of rules in traffic and in various situations that may occur [111];
- **Fight disease and Healthcare**: Thus, as mentioned in Section 3.4.5.3, drones can play an important role in fighting diseases, pandemics, and assisting in the healthcare of the population as a whole.;
- **Traffic Monitoring**: Traffic monitoring can contribute to improving security in urban centers. Thus, the applications described in Section 3.4.2 also relate to safety and security. Another example related to this context and traffic monitoring is the identification of Driving While Intoxicated (DWI) drivers by notifying the responsible authorities to take appropriate measures [171].

Regarding IoD public safety and security applications, in many cases, they are suitable for reactive applications. For instance, a drone can react when finding a victim in an urban disaster or identifying a vehicle crash in traffic monitoring. Also, this is an environment where drones can perform different activities with a common goal. For example, in a big event such as a football match, police drones can work together and share messages forming an event monitoring. However, each fleet of drones can perform a different activity. For instance, it is possible to have drones monitoring people searching for fights, and at the same time, other drones can identify situations where people can invade the field. Besides that, all the data collected in this situation can be used to form statistics about the city, such as traffic-related maps, disease-related maps, and police-related maps, to name a few. These maps can help UC produce data where it is possible to identify locations that need more security or improve road conditions, enhancing the city and people's lives.

3.5 Challenges and Important Issues

For IoD and Urban Computing to connect, it is necessary to face some challenges. First, for people to accept air mobility, it is essential to have a safe environment. Second, IoD needs to be implemented, which also involves numerous challenges. Other related challenges may arise from the integration of IoD with UC. Therefore, the challenges are divided into issues that need to be considered before the implementation of IoD (and continuously improved) and topics that need to be addressed after the implementation of IoD (see Figure 3.11). Some points to be highlighted are IoD data management and its effective use, leading to economic employment in terms of financial and energy costs. Each of these challenges will be discussed in the following sections.

Figure 3.11: IoD Challenges and Important Issues overview



Source: Elaborated by the author

3.5.1 Acceptance of Urban Air Mobility

The acceptance of urban air mobility is essential for drone-related applications to evolve. Some fears may arise related to drone-hardware issues, such as colliding with another drone or a bird, or a drone may crash and fall. Likewise, it is possible to have data privacy issues. Besides, biodiversity conservation and public well-being must also be considered for public acceptance of drones. In the sequence, these factors are detailed.

- **Security and Hardware Issues:** Hardware plays an essential role in the acceptance of drones in society. The main reason is that with many drones providing services in an urban environment, there is a good chance that the drone will fall. A drone can drop for several reasons, such as attacks, collisions with other drones, buildings, structures, or animals, lack of energy, and general failure. The fall of drones can cause fear in people and, consequently, a rejection of the service. Thus, providing a safe and reliable system is a great challenge. Numerous studies are related to this issue, such as safety, collision prevention, increased energy efficiency, and precise and effective control of each drone. This control should include the drone's purpose in completing the service, its path, whether it has enough energy, how long the service will take, and weather conditions. The weather is one of the issues that can most influence the safe use of drones since most drones cannot fly during rain and strong winds;

- **Biodiversity conservation:** The environmental issue is also a factor that can influence the acceptance of air mobility. Some studies suggest the use of drones for biodiversity conservation [172]. Specifically in monitoring and counting species [173], fighting fires [174], and spreading seeds for reforestation [175]. However, there are still no studies on the impacts of drones on animals in the urban environment. Considering the scenario in which many drones provide services to society, it remains to be seen whether this will impact the lives of animals, such as birds that live in the urban environment. In addition, as each ecosystem can vary widely for each region, some regions may be more affected than others;

- **UAV noise reduction:** Drones can make considerable noise [176]. As IoD can be composed of large amounts of drones, the noise produced can be a nuisance for people, increasing noise pollution. This noise can be generated even by small drones. Specifically, the noise made by models like the Mavic Air is around 99 dB [176]. Thus, this issue must be considered from the beginning of the IoD design. So that noise pollution caused by IoD does not become an obstacle, new strategies to reduce the noise of drones effectively will be necessary;

- **Privacy and Big Brother:** Privacy is one of the biggest challenges related to the acceptance of air mobility. With several drones providing services, many people can get the feeling of being watched all the time. Currently, applications that monitor people have already been used. One example is the use of drones during the Covid-19 pandemic. Applications in which drones search for people who are circumventing the quarantine or where the drone identifies people with symptoms such as fever and coughing show how drones can be a great ally in fighting diseases [161]. However, they also show that drones may be watching people. Thus, ethical issues must be established to distinguish between acceptable and the invasion of privacy, avoiding a "Big Brother" style society.

3.5.2 IoD Implementation

Many challenges are related to the effective implementation of IoD. Initially, it is necessary to establish how the air space will be organized. Another issue is testing and developing new communication protocols to make IoD secure. It is also necessary to integrate IoD with other existing networks. In addition, the IoD must respect the flight laws of each country in which it operates. Each of these questions will be detailed in sequence.

- **Distributed UAV Traffic Management:** The first major challenge to implementing IoD is effective, safe, fair, and reliable network control. Due to the characteristics of IoD, traffic management is expected to be distributed. One way to carry out this control is to use the architecture introduced by Gharibi et al. [5]. This architecture is based on three networks: the Internet, cellular, and air traffic control. To carry out distributed management, zones and several service providers (ZSPs - Zone Service Providers) are established;

- **IoD Protocols:** A crucial factor in the use of IoD is communication protocols. Protocols developed for communication between UAVs until then do not consider that other UAV swarms may be carrying out close missions. Thus, as part of airspace control, it is necessary to develop specific protocols for this scenario. Another essential point is that different UAV swarms must collaborate to maintain network security and reliability. The protocols developed need to be tested in scenarios that take into account several UAV swarms working at the same time. Another point is that there are no communication protocols that consider the existence of airways. Hence, the existing protocols must be tested considering the airways. Besides, new protocols must be developed considering all the characteristics of IoD (in both cases, with and without airways). Thus, it will be possible to build more effective protocols for IoD;

- **IoD Integration with Other Networks:** Another issue is that IoD needs to be integrated with other networks. Some studies show that drones can support VANETs. Similarly, other networks can benefit from the help of UAVs. It is also possible that the opposite occurs, i.e., services provided by UAVs can benefit from the assistance of other networks, such as satellite networks, VANETs, and IoT device networks. Integration with other networks can support IoD in several ways. Satellites can provide weather data to help you decide whether or not to fly. VANETs and IoT devices can collaborate for energy savings and Internet access for data transfer. Therefore, a major challenge will be the integration so that it is generally adequate for energy and cost savings for providing services;

- **Legislation:** Legislation can be a great challenge for using drones in different applications. The Federal Aviation Administration (FAA)[1], responsible for regulating drones in the USA, has been demanding, for example, that drones be registered and report their position frequently [2]. IoD architecture is expected to facilitate the creation of laws and regulations to be followed by drone-related manufacturers and service providers. The use of airways, for example, can ease the monitoring of drones, limiting the locations through which they can travel. In addition, it may also be possible to know the drone's location with greater precision periodically. Another critical point is that IoD can facilitate the autonomous flight of drones (without a pilot) and flight over people, as it allows greater control of each flight. Recent changes to the FAA regulations aim to enable commercial drones to fly over people[3]. This fact is seen as a significant step towards providing services such as delivery. However, IoD can supply a safer environment for autonomous and personal flights, allowing the drone to fly further and more drones to operate simultaneously;

- **Artificial Intelligence:** Artificial intelligence has been studied in research related to military drones [177]. In particular, the objective has been to provide a more far-reaching, accurate, coordinated, intelligent, and speedy interaction between warfare drones [177]. These same interaction characteristics must be present in the IoD. Thus, artificial intelligence can also be used to provide intelligence in the IoD, preventing and predicting problems that may occur on the network;

- **IoD Simulator:** In the same way, other networks still in development, such as VANETs, use simulators to develop protocols and applications. IoD also needs a robust simulator to advance its development. Some requirements for an IoD simulator are communication, mobility (airway-based or not), control, coordination, and energy. Some topics can be even deeper and more complex as scenarios that consider the climate (e.g., rain, wind speed), drone real and natural mobility, and energy consumption of the UAVs. Despite the existence of some of these topics (such as communication) in simulators known as NS3[4] and OMNET++[5], the researcher

---

[1] https://www.faa.gov/uas/

[2] https://www.faa.gov/uas/getting_started/remote_id/

[3] https://bit.ly/3Pt7YzD

[4] https://www.nsnam.org/

[5] https://omnetpp.org/

always needs to prepare their environment from scratch. Thus, there is a need for a more robust simulator consolidated by the academic community.

3.5.3 Urban Computing and IoD Data

IoD can generate a lot of data that can be useful in Urban Computing. In Urban Computing, for instance, vehicle data is used to understand human behavior to create new services and improve the quality of life for people in urban centers. Likewise, the data generated by IoD can contribute to expanding the effects of Urban Computing on people's lives. With this, some challenges arise, such as the heterogeneity of the data and data overload. On the other hand, new perspectives may emerge with this data, enabling the creation of new areas of study, such as IoD Data Mining.

- **Data Heterogeneity:** The enormous data heterogeneity is already a problem facing Urban Computing. The more data is collected, the greater the challenge becomes to aggregate them and, consequently, the more challenging the data analysis. Thus, the use of drones for the collection helps to increase the heterogeneity of the data, even when they are on the same theme. However, aggregating this data is vital for more complex analyses to be carried out to improve services, applications, and quality of life;

- **Data Overload:** In urban computing, data is collected in different ways. For example, data on air quality can be obtained by cell phones, cars, and static nodes at strategic points in urban centers. Drones can also acquire air quality data. Thus, a large amount of data is generated, and there may be data redundancies and problems with storage. One of the challenges is to analyze and select which data is instrumental;

- **IoD Data Mining:** An aerial view can bring new perspectives to understand cities. Thus, creating a new city profile based on the trajectory of aerial and terrestrial vehicles will be possible. The UC seeks to understand various aspects of the city through data about people's mobility and use this data to improve people's lives [8]. An example is identifying points of interest using urban mobility data [8]. This process can, for example, determine a trendy restaurant in a region. Using drones as service providers can change how people behave in urban centers. Thus, it will also be possible to identify through the drone trajectory data establishments that are being used a lot and, therefore, understand and improve that region. For instance, suggesting new businesses to meet a region's demand and identifying points of interest where people prefer to go in person and where they prefer to ask for some service. IoD enables new segments that connect IoD and urban computing, such as IoD Data Mining and Drone Trajectory Data Mining.

3.5.4 IoD and Economy

As already mentioned, many applications that use drones have emerged. In most applications, drones perform activities that other devices or people previously did. Many researchers justify using drones in these cases due to speed, flexibility, and low cost [12, 35]. However, research needs to demonstrate these advantages effectively. This challenge includes the effective reduction of energy expenditure in large cities. For example, if the delivery of goods by drones is more beneficial to the environment in terms of energy and CO_2 emissions, it would be interesting if this type of delivery were also less costly. Heterogeneous network use should also be explored. Another option for savings is using the same drone node to perform different activities in the same mission. So it is a great challenge to study, understand and integrate IoD with other networks so that real savings happen.

3.6 Final Remarks

This chapter discussed the IoD application in the urban context. We highlighted how IoD and Urban Computing are connected. Also, we introduced a framework for UC in IoD Context. In the sequence, we discussed the IoD-UC Applications. The IoD-UC applications can bring insights into the design of IoD protocols because they reveal the dimension, needs, and challenges of the IoD-UC environment. Last, we summarize the challenges and important issues observed in this context.

Chapter 4

Path Planning for IoD

A unique feature of IoD is the shared airspace between multiple drone applications. Thus, when designing a path planning for a monitoring application, other drones are expected to perform other services such as delivering goods. Considering the number of drones, different applications, and companies that can be IoD users, it is more realistic to have a decentralized path planning and build an online collision-avoidance mechanism. Thus, the IoD resembles VANETs forming a constant flow of drones.

First, in Section 4.1 we propose an airspace and airway topological architecture intending to organize the environment to allow multiple drones and applications. In Section 4.2 we we conducted a study on the advantages of food drone delivery compared to traditional delivery. After, in Section 4.3 we developed a Coverage Path Planning for IoD. We consider the airway scenario previously established to design the algorithms. Section 4.4 brings the final remarks of this chapter.

4.1 Airspace and Airway Topological Architecture

In this Section, we propose an airway topological architecture for IoD. As mentioned, it is expected that soon drones will play a greater role in smart cities [12]. The drone delivery of goods will probably lead to a higher traffic flow. Also, other applications such as monitoring and smart cities sensing will share the same airspace. Moreover, some applications can provide services more urgently than others in this scenario. Gharibi et al. [5] proposed the use of airways uses to organize the airspace by forming a graph. However, the authors do not discuss how to organize the airways in the airspace. Table 4.1 summarizes important issues in airway topological architecture.

One of the big challenges in IoD is to determine where the airways can pass. One way to organize the airways is to overlap them on the landways. In this case, the airways cannot pass over houses and buildings. However, some exceptions can occur as squares and parks. Another way is to keep a free-position airway. Specifically, airways are

Table 4.1: Important issues in airway topological architecture

Feature	**Highlighted**
Airways parallel to Terrestrial Roads	Determine where airways can pass. Helps to share and coordinate drone access to the airspace.
Overlaps Airways	Provides airway expansion by overlapping them at different altitudes.
Priority Airways	Provide a way of priority for some health applications, such as blood and medicine delivery.

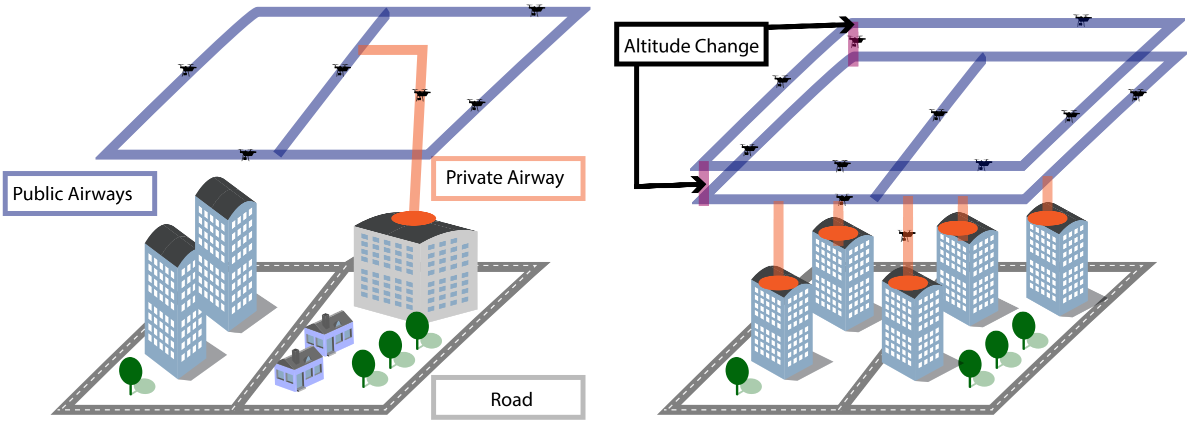
anywhere except in some areas considered unsuitable for drone flight, such as airports, government buildings, prisons, to name a few. In this case, the airways can pass over houses, buildings, and squares. Table 4.2 presents the benefits and drawbacks of each airway position approach.

Table 4.2: Benefits and drawbacks of each airway position approach

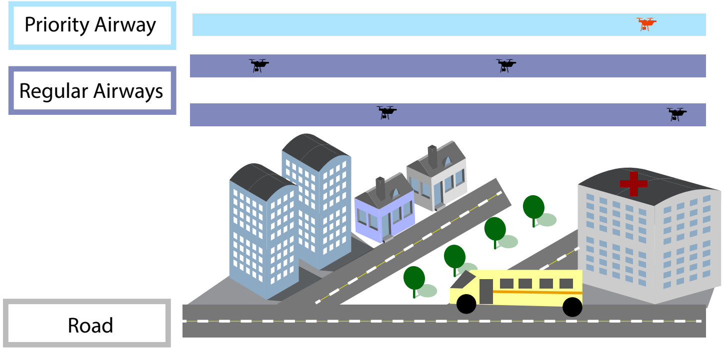
Airway Position	**Benefits**	**Drawbacks**
Over landways	• Easier public acceptance • Making regulations and laws easier • Easier collaboration with VANET and PTNs	• Airspace is not used in the most efficient way
Free	• Possibility of shorter paths	• More difficult to have public acceptance • More difficult to create regulations and laws

To overcome the airways challenge we propose parallelling the airways to existing terrestrial roads. Thus, drones would only pass through permitted public areas. This approach can help organize drone traffic and facilitate the development of laws that regulate traffic. Furthermore, this structure can help accept drones by society, as people will know precisely where drones can go. It is reasonable to assume that the population can more readily accept a network of drones with a constant flow in the city if the airways do not pass over their homes. The risk of accidents with falling drones can make the free-standing airway approach not accepted by the population. Thus, despite the disadvantage of not efficiently taking advantage of the airspace, positioning the airways on the existing landways seems the most suitable way to organize the airspace. Also, this approach allows a more efficient collaboration between IoD and some terrestrial networks such as VANETs and PTNs. Figure 4.1a exemplifies airways parallel to terrestrial routes.

Figure 4.1: Airways architecture illustration



(a) Parallel airways (b) Overlaps airways



(c) Priority airways

Source: Elaborated by the author

Figure 4.1a also classifies the roads between public and private. As with streets and roads, companies, individuals, and government agencies may have airways within their property to allow the delivery of goods. Hence, only authorized drones can enter that route. Therefore, this configuration helps protect areas where drone overflight is prohibited, such as airports, military areas, and government buildings.

Another point is that central areas of cities with many delivery areas can have heavy drone traffic. This fact can cause drone traffic congestion and affect road safety. It is expected that the bigger the traffic, the bigger the chances of unexpected accidents happening. Hence, in some city areas, more airways will be needed to accommodate drone traffic. As illustrated in Figure 4.1b, these airways can be overlapped on existing airways forming new paths. In this architecture, drones can change altitude using specific airways. Also, this architecture still follows the rule of allowing airways only parallel to terrestrial roads.

Drones have been used in applications so diverse as microbiology and infectious diseases [164, 178, 179]. In particular, some uses are the delivery of medicine and other things related to health, such as blood, organs, life-saving medical supplies, and equipment [164]. Drones have also been used in the fight against the Covid-19 pandemic [180]. For example, in monitoring areas affected by the disease, aerial spray and, disinfection, and delivery of blood and medicines [181, 182, 183]. This type of delivery is related to saving lives. Thus, some delivery categories should prioritize considering other delivery services and other applications. Also, different kinds of applications can benefit from the priority to use airways, for instance, in search and rescue missions. Figure 4.1c illustrates the priority and regular airways.

4.2 Drone Delivery Path Planning

Despite significant advancements in drone delivery, it remains essential to understand the circumstances where drone delivery offers advantages in terms of why, where, and when it should be employed. This section introduces a comprehensive guideline for developing decision systems for drone delivery within the IoD. Specifically, we present a systematic approach to the drone delivery decision problem that tackles those questions (why, where, and when).

4.2.1 Preliminaries

In this section, we introduce the relationship between IoD and drone delivery. Also, we provide an overview of related studies conducted in this field.

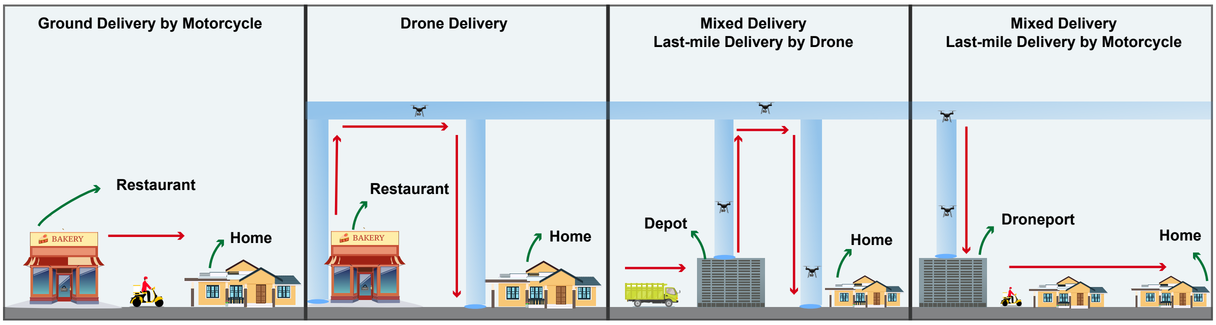
4.2.1.1 Drone delivery and the Internet of Drones

Drone delivery encompasses various categories, including the transportation of healthcare products, food, and small everyday goods. The drone delivery of healthcare products primarily focuses on expedited transportation due to the critical nature of these essential items for saving lives. This category has witnessed significant advancements in recent years [184]. Blood [184], medications [185], and human organs [186] are among the most significant items considered for this type of delivery.

Small everyday goods delivery is a highly sought-after category for companies investing in drone delivery services. Amazon is an example of a company venturing into this field, despite the challenges associated with service regulation [187]. They have initiated testing in select cities in the United States to explore the feasibility of using drones for delivering such goods [188]. Food delivery is another prominent category, which encompasses various scenarios. It involves both time-sensitive deliveries for immediate consumption, such as restaurant orders and small grocery purchases. In recent years, the demand for contactless food delivery has surged, driven by the Covid-19 pandemic [189]. Drones have garnered significant attention as a viable solution for this type of delivery. To integrate drone delivery seamlessly into our daily lives, it becomes crucial to establish organized airspace that regulates and coordinates drone access. One approach to achieving this organization is through the IoD architecture, where drones utilize well-defined airways to form a continuous flow network in the sky, similar to a vehicular network. Moreover, these airways can overlap ground routes, enabling easier identification and regulation of airspace.

In the first part of Figure 4.2, we depict the conventional delivery process where a delivery person collects the order from the seller and transports it to the destination. Conversely, the second part represents end-to-end drone delivery, where the vehicle is replaced by a drone that collects the order from the seller and delivers it directly to the customer's home. Furthermore, mixed delivery approaches involve a combination of vehicle types. For instance, a depot could receive orders, and drones could transport the products to the end consumers. This approach, known as last-mile delivery, has garnered significant attention in recent literature [190]. Given the challenges associated with ensuring a safe landing spot for drones at each residence [191], one possible solution is to leverage drones for a portion of the delivery process. The drone would transport the order to designated droneports, and the last-mile delivery could be completed using other vehicles, such as motorcycles.

Figure 4.2: Deliveries approaches overview



Source: Elaborated by the author

4.2.1.2 Related Work

Research in the field of drone delivery primarily revolves around studying and forecasting the proliferation of drones within urban environments [192]. Narkus-Kramer [193], for instance, predicted that the number of drones in Washington could reach 1 million by 2050. As drones are still an emerging technology, significant attention is given to understanding the factors influencing drone delivery service acceptance [189, 194, 195, 196]. It is important to note that acceptance can evolve. A prime example of this is the impact of Covid-19, which has brought a new perspective to this form of delivery as a means of contactless goods transportation to end consumers [189]. Consequently, many individuals have recognized the advantages and the necessity of this type of delivery. Furthermore, certain studies [187, 197] aim to elucidate the reasons behind the slower-than-expected progress of drone delivery, highlighting the challenges and factors contributing to this phenomenon.

Considering this scenario, several related questions have emerged regarding why, when, and where drone delivery benefits society. To comprehend the benefits of drone delivery, certain studies concentrate on healthcare delivery and examine the transportation of specific items such as medicine [198, 199], human organs [186], and blood [184]. In these cases, swift delivery is crucial as human lives can be contingent on the successful arrival of these items. Drone delivery of blood, for instance, is already being implemented, offering the advantage of rapid transportation [184]. Furthermore, these studies also assess whether drone delivery can help mitigate waste, considering that such products require specific storage conditions [184]. Other research endeavors conduct a comprehensive analysis of drone delivery, focusing on the advantages in terms of energy efficiency [200], cost-effectiveness [201, 202], sustainability [202], and delivery time [184, 203]. These studies can be classified into two categories: those that aim to develop more efficient algorithms and models for path planning and drone systems [202, 203], and those that compare drone delivery with ground vehicles [184, 200].

Table 4.3 provides an overview of the relevant studies conducted in the field. Zeng et al. [191] focused on estimating and allocating droneports based on e-commerce data specifically from Singapore. However, they did not consider the comparative advantages of utilizing drones in different scenarios. Jiang and Ren [192] compared drone delivery with traditional rider-based delivery, but did not incorporate actual delivery data. On the other hand, Nisingizwe et al. [184] evaluated the use of drones for delivering blood in Rwanda, utilizing data from existing operational drone deliveries. However, that study primarily focused on providing specific products to hospitals, which may not encompass broader scenarios that aim to serve the entire population, such as delivering food and everyday goods. Therefore, this work aims to comprehensively compare delivery times, considering various delivery options, including ground delivery, drone delivery, and a combination of both (mixed delivery), utilizing real delivery data.

Table 4.3: Related studies

Year	Ref.	Category	Source	Main Goal
2020	[191]	Everyday Goods	The data of total e-commerce demand for Singapore was estimated from Statista Portal, Shopee and Amazon	Introduce a practical and effective optimization model that addresses the placement of droneports
2020	[192]	Food	A survey made by the authors about expected drone delivery time in China	Compare the expected delivery time with the real delivery time and analyze the conditions in which the drone delivery outperforms the rider delivery
2022	[184]	Health	Rwanda's Health Management Information System (HMIS)	Evaluate the effect of using drones in terms of time and waste to deliver blood products to remote health facilities
2023		**Food**	**Aiqfome, a Brazilian company operating in the field of online meal delivery**[1]	**Propose guidelines for drone delivery decision in a real city, taking into consideration ground delivery, drone delivery, and mixed delivery**

4.2.2 Deliveries Approaches

In this section, we introduce the dataset employed in this study and utilize it to examine three different delivery approaches: ground delivery, drone delivery, and mixed delivery.

4.2.2.1 Study Setting

Aiqfome[1] is a Brazilian food delivery company founded in 2007, exclusively operating within Brazil. Presently, the company boasts a user base of over 5 million individuals and collaborates with approximately 25,000 registered restaurants across around 700 cities in Brazil. Aiqfome operates by facilitating connections between restaurants, deliverers, and consumers. Table 4.4 provides an overview of the collected data from deliveries conducted in six cities where Aiqfome operates, covering the period between July 22, 2022, and May 19, 2023.

The Aiqfome dataset comprises a collection of motorcycle delivery records, including information such as the restaurant's coordinates, the departure time of the delivery person from the restaurant, the coordinates of the delivery destination, the arrival time

---

[1] https://bit.ly/43oADwo (in Portuguese)

Table 4.4: Aiqfome dataset overview covering the period between July 22, 2022, and May 19, 2023

City	**State**	**Population (thousands) [204]**	**# Deliveries**	**Average real driving distance (km)**	**Average estimated Google driving distance (km)**	**Average real driving time (min)**	**Average real driving speed (km/h)**
SORRISO	MT	94.941	3525	8.36 ± 5.62	6.65 ± 5.22	13.84 ± 9.58	41.22 ± 25.81
CERQUILHO	SP	50.631	2096	4.77 ± 2.71	3.45 ± 2.15	7.69 ± 5.02	41.32 ± 19.42
TIETÊ	SP	42.946	1736	6.30 ± 4.25	5.23 ± 4.12	10.51 ± 7.02	40.22 ± 31.51
MARINGÁ	PR	436.472	1459	7.86 ± 4.20	4.99 ± 2.90	13.51 ± 7.64	38.93 ± 22.96
CONSELHEIRO LAFAIETE	MG	130.584	1356	5.66 ± 3.34	3.73 ± 2.32	10.96 ± 7.51	35.83 ± 20.41
COLATINA	ES	124.283	1139	8.34 ± 5.45	5.52 ± 3.69	15.12 ± 12.05	40.29 ± 31.67

of the delivery person at the destination, the distance in kilometers of the route suggested by Google from the restaurant to the delivery location, and the distance in kilometers traveled by the delivery person. All analyses and algorithms presented in this study were implemented using Python 3.11. Table 4.4 summarizes this data and displays the estimated population figures for each city in the dataset.

Table 4.5 highlights the key distinctions between motorcycle-based delivery and drone delivery. We considered data from the drones used in Amazon's Prime Air[2] service to conduct this comparison. Drones for delivery typically transport only one package at a time, whereas motorcycles can accommodate multiple packages. Regarding food delivery, it is common for a motorcycle courier to handle various deliveries simultaneously. Regarding capacity differences, motorcycles can travel around 40 km per liter and typically have a fuel tank capacity of 9 liters. In comparison, Amazon drones have a range of 12 km for a round trip.

Table 4.5: Motorcycle × drone delivery

	Motorcycle	**Drone (based on Prime Air [2])**
Capacity	More than one package	Usually 1 package (2.2 kg)
Autonomy	40 km/l (tanque 9 liters)	12 km
Autonomous	No	Yes
Speed	40 km/h (average based on Table 4.4)	80 km/h (maximum)
Safety	Subject to traffic accidents	Subject to attacks and hardware problems
Weather	Usually unaffected by rain	Strong winds can shut down the service

One advantage of drones is their autonomy, as they operate without a human operator, which can be cost-effective. Although motorcycles have a higher maximum speed in theory, traffic congestion and urban speed limits often result in lower average rates than drones, which can reach a maximum speed of 80 km/h. Safety is another

---

[2] bit.ly/3prbX7B

differentiating factor between these vehicles. While motorcycles are susceptible to traffic accidents, drones may face issues such as attacks. Lastly, weather conditions are crucial, as rain and particularly strong winds can affect or even interrupt drone services.

4.2.2.2 Ground Delivery

Different types of vehicles, such as cars, motorcycles, and trucks, can perform Ground Delivery. The AiqFome dataset represents real data from deliveries performed by motorcycles. It is worth noting that in Brazil, most deliveries are carried out using motorcycles. This choice of vehicle is due to factors such as the climate, ease of parking, and cost-effectiveness in terms of fuel for the deliverers. In this section, we analyze and summarize the data from this dataset for comparison with other types of deliveries.

Table 4.4 presents the average delivery time, considering the departure time from the restaurant and the arrival time at the destination. By analyzing the time spent on deliveries and the distance traveled, we calculated the average speed of the deliverers. Also, we computed the average distance covered by the deliverers and the average distance recommended by Google. Google tends to propose shorter or more efficient routes, considering factors such as traffic conditions and permitted speeds on the roads. However, the average distance the deliverers takes exceeds the average length suggested by Google. There are various reasons for this discrepancy, including the deliverers choosing their routes for convenience, avoiding unsafe areas, and even simultaneously making multiple deliveries using different delivery applications.

In the dataset, for the most populous city (Maringá), the average delivery time is approximately 13.51 min, covering a distance of 4.99 km at an average speed of 38.93 km/h. On the other hand, for Tietê, the smallest city in the dataset, the average delivery time is 10.51 min, with the delivery person traveling 5.23 km at an average speed of 40.22 km/h. When examining the relationship between these population statistics and the average distance traveled during deliveries, it becomes apparent that city size alone does not necessarily dictate the distance covered in deliveries. Several factors come into play when determining the distance traveled by deliverers. Factors such as the size of the city, the layout of neighborhoods, the presence of commercial quarters, and the types of restaurants registered in the application all contribute to this variation. In larger cities, for instance, restaurants may only offer delivery services within specific areas, whereas in smaller cities, restaurants might cater to a broader geographic range.

4.2.2.3 Drone Delivery

In this section, we delve into goods delivery utilizing drones. Specifically, we explore the concept of end-to-end drone delivery. Numerous companies may be involved in providing drone delivery services (e.g., Amazon, Google, DHL, Uber), simultaneously managing a significant fleet of drones within the airspace. Therefore, we examine the IoD architecture, wherein drones have coordinated access to airspace. We employ well-defined airways that closely align with existing land routes to facilitate drone airspace coordination. This approach ensures that drones avoid unauthorized areas such as airports and government buildings while simplifying navigation in densely populated areas with buildings, towers, and antennas. Additionally, this strategy promotes public acceptance by providing clear localization of drone flight paths and avoiding drones directly over residential areas.

By considering the IoD concept with well-defined airways, we conducted simulations of deliveries using drones on the Aiqfome dataset. The distance results are presented in Table 4.6. Notably, in the city of Maringá, it is observed that a drone with a 12 km range (similar to Amazon's Prime Air drone) may face challenges in completing the delivery and returning to the point of origin. Another point is a significant disparity in the distance covered by drones compared to real distance. This disparity is quantified by the distance shortening rate (DSR), which represents the percentage by which the drone delivery distance is smaller than the actual distance traveled by the delivery person. On average, the drone covers 61% less distance compared to the courier's actual travel distance. Colatina exhibits the largest disparity with an average reduction of 75%, while Maringá showcases the smallest difference, with the drone route being 44% shorter than the actual route.

Table 4.6: Comparision of ground delivery with end-to-end drone delivery

City	**Real driving distance (km)**	**Drone delivery distance (km)**	**DSR - real driving to Google (%)**	**DSR - real driving to drone (%)**	**Real driving time (min)**	**Drone delivery time (RDS) (min)**	**Drone delivery time (MDS) (min)**
SORRISO	8.36 ± 5.62	1.89 ± 2.01	21.59 ± 26.90	69.18 ± 29.84	13.84 ± 9.58	2.83 ± 3.26	1.42 ± 1.51
CERQUILHO	4.77 ± 2.71	1.65 ± 1.16	25.67 ± 29.48	57.84 ± 29.20	7.69 ± 5.02	2.67 ± 2.11	1.24 ± 0.87
TIETÊ	6.30 ± 4.25	1.90 ± 1.71	16.02 ± 40.85	61.85 ± 30.60	10.51 ± 7.02	3.02 ± 2.80	1.42 ± 0.87
MARINGÁ	7.86 ± 4.20	4.07 ± 2.23	34.01 ± 22.54	44.28 ± 22.56	13.51 ± 7.64	7.33 ± 4.63	3.05 ± 1.28
CONSELHEIRO LAFAIETE	5.66 ± 3.34	1.69 ± 1.31	30.02 ± 27.61	62.96 ± 27.57	10.96 ± 7.51	3.24 ± 2.93	1.26 ± 0.98
COLATINA	8.34 ± 5.45	1.66 ± 1.72	28.01 ± 30.57	75.04 ± 23.80	15.12 ± 12.05	2.77 ± 3.24	1.24 ± 1.29

DSR - Distance Shortening Rate; RDS - Real Delivery Speed; MDS - Maximum Delivery Speed

We also compared the actual route couriers took and the distance Google suggested.

Aiqfome relies on a route provided by Google's API for the delivery person, and this distance is recorded to determine if the courier followed a distance similar to the suggested one. Table 4.6 provides the DSR, comparing the distance couriers traveled with the route Google suggested. We observed that the route suggested by Google is approximately 25% shorter than the actual route taken by the couriers. This indicates that even if couriers followed the suggested route, drone delivery would still be shorter in terms of distance. The distance traveled also has an impact on delivery time. Table 4.6 also shows the delivery time in each of the analyzed cities. We compared the actual delivery time with the projected delivery time of drones assuming they operated at the same average speed as ground delivery. Additionally, we presented the delivery time if the drones operated at their maximum speed, given that the reference drone has a maximum speed of 80 km/h. Consequently, if the drone operated at the average speed of motorcycles, the delivery time would already be faster.

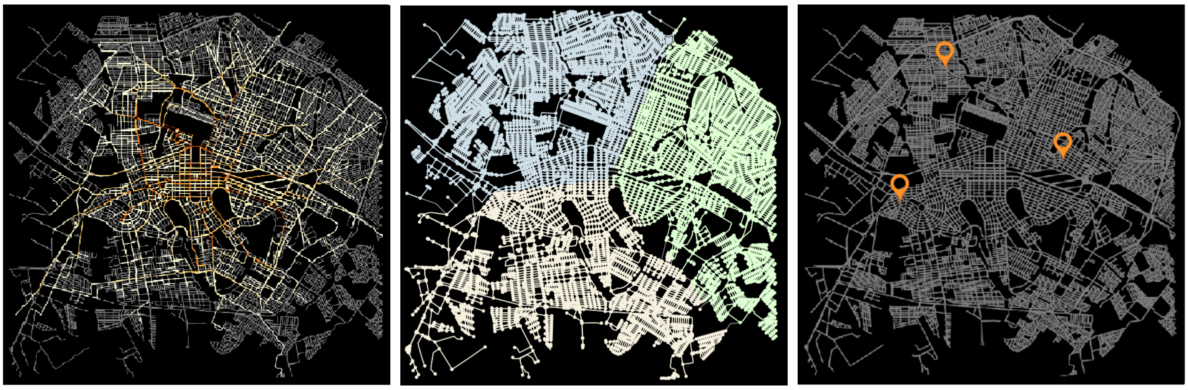
4.2.2.4 Mixed Delivery

The concept of mixed delivery involves utilizing multiple vehicle types for product delivery. Given that our dataset primarily consists of deliveries made by motorcycles, we consider both motorcycles and drones in our study. Some previous studies suggest that drones are well-suited for last-mile delivery scenarios due to their limited autonomy imposed by battery constraints [190]. Therefore, we examine the possibility of motorcycles transporting goods to a central depot, where drones take over for the last-mile delivery. However, it is worth noting that other studies emphasize the need for safe landing areas for drones, known as droneports [191]. In certain urban settings, individual homes may not provide a secure landing spot for drones. To address this, one possibility is to establish droneports strategically located throughout the city, while motorcycles handle the last-mile delivery. This section first delves into distributing droneports or depots within a city. Subsequently, we simulate mixed-delivery scenarios, where drones are responsible for the last-mile delivery in one scenario, and motorcycles handle the last-mile delivery in the other scenario.

► **Depots and Droneports Distribution**. In the mixed delivery scenario, we focused on the city of Maringá. This city is the largest population within the dataset and presents unique characteristics in the end-to-end drone delivery results. Specifically, it demonstrated the potential for routes exceeding the 12 km round trip capacity of the reference drone used in this study. To address this, we established a flight autonomy of 10 km to ensure an appropriate safety margin. Consequently, we considered that the drone could cover a radius of 5 km, encompassing an area of approximately 78 km². According to the Brazilian Institute of Geography and Statistics, Maringá has an urban area of 112.70 km². Hence, we determined that three droneports or depots would be sufficient to fulfill the city's delivery demand. We consider this because drones need to follow the airways and probably will need to fly a considerable distance to achieve a point compared to free flight.

While certain studies propose the distribution of droneports [191, 205], they rely on e-commerce data unavailable for the Brazilian cities examined in this research. Additionally, none of these studies consider a real delivery dataset within an IoD scenario where dedicated airways are present, resembling terrestrial routes. We employed the K-means algorithm to partition the Maringá graph into three sections [206]. The resulting segmentation can be observed in Figure 4.3b. We analyzed the vehicular flow on each road segment to identify the suitable location for the depot or droneport. Specifically, we examined the flow of vehicles for each delivery within the Maringá dataset, considering the shortest route between the origin and destination points. This approach enabled us to identify roads that potentially experience higher traffic levels, as illustrated in Figure 4.3a.

Figure 4.3: Maringá results overview



(a) Flow of vehicles (b) K-means (c) Algorithm 1

Source: Elaborated by the author

Algorithm 1 outlines the process of distributing points designated as droneports or depots. Each node possesses two attributes: *flow*, representing the frequency of a node appearing in all routes within the Aiqfome dataset, and *egc*, denoting the eigenvector centrality [207] that signifies the influence power of a node based on its neighboring nodes. On lineS 1-2 of the algorithm, the mean and standard deviation of the *flow* attribute is calculated across all nodes. Subsequently, in lines 5 to 8, nodes with a *flow* greater than the mean and less than the mean plus the standard deviation are added to the set T. Finally, on line 9, the node with the highest *egc* value is appended to the set of nodes that will serve as droneports or depots. Table 4.7 presents the results of

Algorithm 1. The displayed distance is the sum of the distance from the departure point to the depot/droneport plus the distance from the depot/droneport to the destination. We compare the analysis using only the point closest to the mean and the point with the highest *egc* and we verify that our method that uses the metrics together produces the best result.

Table 4.7: Last-mile delivery results

Route	**Distance (km)**			**Last-mile Delivery by Drone (min)**	**Last-mile Delivery by Motorcycle (min)**
	Average	**EGC**	**Average + EGC**	**Average + EGC**	
Source to Droneport/Depot	3.34 ± 1.67	4.70 ± 1.61	3.27 ± 1.46	6.16 ± 3.81	2.45 ± 1.09
Droneport/Depot to Destination	3.82 ± 1.87	4.13 ± 1.80	3.45 ± 1.58	2.59 ± 1.19	6.35 ± 3.77
Total	7.16 ± 2.06	8.84 ± 1.89	6.73 ± 1.72	8.75 ± 3.61	8.80 ± 3.65

```
Algorithm 1: Depots/Droneports distribution
Input  : graphs
Output: nodes
nodes ← {}
for (G in graphs) do
    $\alpha \leftarrow average(G.nodes.flow)$
    $\beta \leftarrow ste(G.nodes.flow)$
    $T \leftarrow \{\}$
    for (n in G.nodes) do
        if $(n.flow > \alpha \land n.flow < \alpha + \beta)$ then
            $T \leftarrow T \cup n$
    $nodes \leftarrow nodes \cup max(T.egc)$
return nodes
```

- **Last-mile Delivery**. Table 4.7 presents the results of delivery times for the last-mile delivery performed by drones or motorcycles, taking into account the previously determined locations of the depots or droneports. For this analysis, we assume that the section carried out by motorcycles maintains the average speed of real deliveries (see Table 4.4) In contrast, the segment carried out by drones operates at a speed of 80 km/h. The time required for the drone's ascent and descent is calculated as 0.005 min. In this calculation, we consider that the drone ascends to a maximum altitude of 200 m. The results show that the average delivery time for drones last-mile delivery is 2.59 ± 1.19 min, while a delivery conducted solely by the drone at maximum speed takes approximately 3.05 ± 1.67 min. Although the mean delivery time is slightly lower for the mixed-delivery approach, the difference is not significant when considering the standard deviation. The results are similar when last-mile delivery is done by motorcycles.

Therefore, it is evident that for a city like Maringá, which is considered medium-sized in Brazil, the mixed-delivery approach with last-mile delivery does not provide significant time advantages. However, it is essential to consider other factors that may influence the decision-making process. For instance, the analysis of the Maringá delivery dataset reveals instances where the delivery distances exceed the reference drone's autonomy of 12 km. In such cases, last-mile delivery can be beneficial as the maximum distances traveled by the drones remain within the threshold.

In last-mile delivery carried out by drones, motorcycles can transport multiple orders to depots in a single trip, thereby reducing the overall travel time. On the other hand, in last-mile delivery conducted by motorcycles, drones cannot consolidate packages for efficiency during the initial leg, whereas motorcycles can deliver multiple packages in a single route. Although this second method is not extensively discussed in the literature, it is reasonable to assume that in urban areas with a high housing density, drones may face challenges in finding suitable locations for package delivery, thereby hindering the feasibility of last-mile delivery by drones.

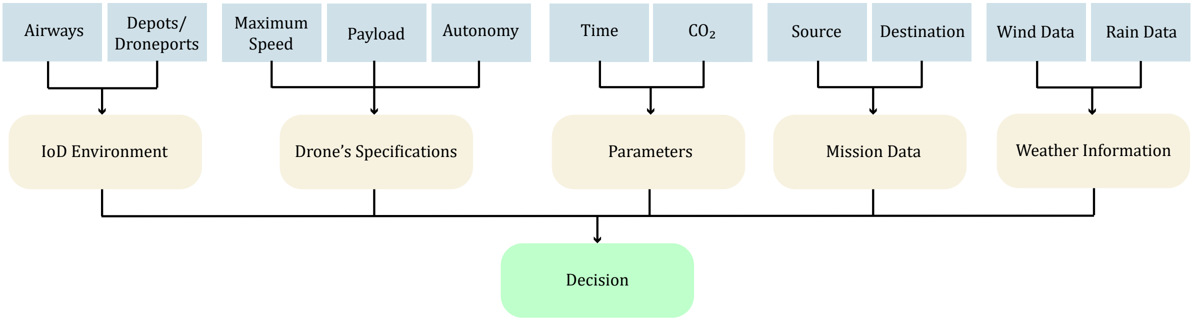
4.2.3 Drone Delivery Guidelines for IoD

This section introduces a guideline for drone delivery in the IoD context. Additionally, we conducted a case study in the city of Maringá to assess the effectiveness of the proposed guideline.

4.2.3.1 Guidelines Overview

Figure 4.4 provides an overview of the drone delivery guide in the context of the IoD. The yellow boxes represent essential factors for decision-making, while the blue boxes represent examples of data that can be utilized. The initial step in determining the most favorable delivery type involves gathering environmental data, including airway information, depot and droneport locations. Collecting drone-specific data such as autonomy, speed, and payload capacity is also crucial. Another requirement is selecting the factor to consider when choosing routes, such as time, cost, or CO_2 emissions. Different priorities can yield different solutions; for instance, the fastest delivery may not always be the most cost-effective or environmentally friendly option. Additionally, the delivery decision necessitates knowledge of the delivery origin and destination, along with weather conditions.

Figure 4.4: Drone Delivery Guidelines for IoD



Source: Elaborated by the author

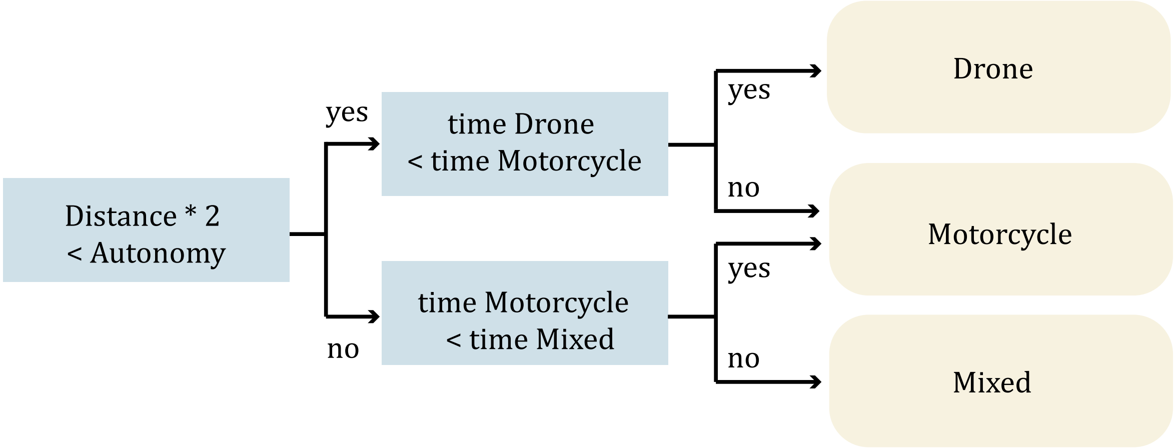
4.2.3.2 Study Case

Aiming to verify the effectiveness of our proposed guideline, we applied it to design an IoD drone decision system for the city of Maringá. The step-by-step process is presented as follows.

- **IoD Environment**: We consider airways overlapping the roads for the urban area of Maringá (area of Figure 4.3). The airways form a graph where the intersections are the nodes, and the connections are the edges. We also consider the depots created by Algorithm 1, as shown in Figure 4.3c.
- **Drone's Specifications**: We consider the drone's specifications in Table 4.5.
- **Parameters**: Our goal was to opt for the fastest delivery method. We do not consider cost or other elements such as sustainability.
- **Mission Data**: We simulate all deliveries for Maringá available in the Aiqfome dataset.
- **Weather Information**: We consider only good conditions in this decision process.

The decision-making process for choosing the delivery method is based on the scheme illustrated in Figure 4.5. Firstly, we calculate the distance between the departure and destination points. To ensure the feasibility of using drones, the mission distance (calculated as twice the distance between the points) should be within the range of the drone's autonomy. If it is possible to complete the entire delivery using drones, we compare the estimated delivery time for both drone and motorcycle deliveries, considering the airways in the IoD scenario. If the full delivery cannot be accomplished with drones, we assess whether a mixed delivery approach or solely using motorcycles would be more advantageous, considering the estimated delivery times for each option. In the case of mixed delivery, only the last-mile delivery by drones is considered. Also, the distance estimation for motorcycle delivery is determined based on the Aiqfome dataset using Google distance.

Figure 4.5: Drone Delivery Decision System for Maringá



Source: Elaborated by the author

Using the system of Figure 4.5, we determine the most time-efficient delivery method based on the distance. The results indicate that out of the 1459 trips, 19 should be conducted by motorcycles, 997 by drones, and 443 through mixed delivery. Table 4.8 displays the results regarding time and distance, and compares them with those obtained when using a single delivery method for all trips. It is evident that with the decision system, the overall average total delivery time is reduced compared to using only one delivery method for all trips. Furthermore, this method ensures that all deliveries assigned to drones are feasible, as the maximum autonomy limit of 12 km (round trip) is considered.

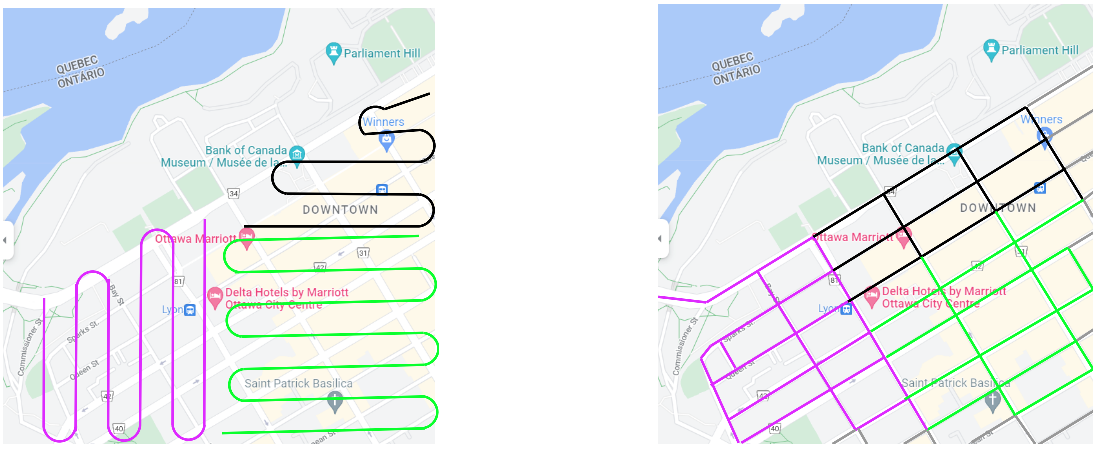
Table 4.8: Summary of results for distance and time for Maringá

	Single delivery strategy			**Multi delivery strategy**			
	Motorcycle	**Drones**	**Mixed**	**Motorcycle**	**Drone**	**Mixed**	**All**
Distance (km)	7.86 ± 4.20	4.07 ± 2.23	6.73 ± 1.72	4.73 ± 2.24	3.09 ± 1.43	7.70 ± 1.76	**4.51** ± **2.62**
Time (min)	13.51 ± 7.64	7.33 ± 4.63	8.75 ± 3.61	9.84 ± 4.31	5.81 ± 3.43	9.51 ± 3.48	**6.99** ± **3.86**

4.3 Coverage Path Planning for IoD

In this section, we proposed a Coverage Path Planning for IoD considering the previous airway scenario established. Overall, the CPP for drones considers two phases. The first executes the decomposition of the area, and the second performs the coverage by the decomposed area. The decomposition phase allows for greater flexibility in the path planning when compared to airway-based IoD. This work focus on coverage path planning applications for IoD based on airways. Thus, drones need to cover a given region using the available airways. Figure 4.6 shows the difference between typical CPP and CPP-IoD.

Figure 4.6: Difference between typical CPP and CPP-IoD



(a) Typical CPP (b) CPP-IoD

Source: Elaborated by the author

Airways can be arranged parallel to landways. This type of organization can facilitate the acceptance of drones by the population and the creation of regulations. Without airways, drones can travel to any space in the delimited area, as shown in Figure 4.6a. In IoD, drones must travel through the delimited space using the available airways, as shown in Figure 4.6b. The first step in CPP is to delimit the area of interest to be covered. The area can have different shapes, such as rectangular, concave, and convex polygons [208]. Next, some decomposition technique is used to ensure full coverage of the scenario. This step has crucial differences from the CPP-IoD, in which the objective is to cover all the airways within the delimited area. The two main differences are described as follows. First, in traditional CPP, the goal is to cover the entire delimited area, while in CPP-IoD, the goal is to cover all airways segments in the coverage area; Second, in traditional CPP, drones can fly in any direction and at any altitude, while in CPP-IoD, drones must follow the airways altitudes and directions.

4.3.1 Related Work

In recent years, the academic community has made significant efforts to develop work related to CPP for terrestrial robots and drones [208, 209, 210, 211, 212]. CPP finds applications in various fields, including mapping, monitoring, search and rescue missions, and industrial inspections, to name a few [210]. These applications require scanning a particular area to achieve their respective objectives. Most studies consider scenarios where the scanned site may have obstacles. However, when barriers are disregarded, unrestricted movement is assumed for terrestrial robots and drones. The following sections will detail related proposals on CPP for drones and robotics.

4.3.1.1 CPP for Drone Networks

Recent surveys on CPP for drones [208, 213, 214] highlight that drones have several factors that need to be considered in addition to those already commonly used by land robots. These factors include the physical characteristics of the vehicle, energy limitations, restricted payload, environmental conditions, and others [208]. Table 4.9 summarizes recent work related to CPP for drones.

In recent years, the academic community developed different CPP methods for single [46, 215, 218, 219, 222, 223] and multi-drone [44, 45, 214, 216, 217, 220, 221, 224, 225]. As shown in Table 4.9, many studies have considered an explicit focus on a single application, such as search and rescue [45, 216], surveying [219], aerial imagery [220], surveillance [221], structural inspection [223], spraying [46] and precision farming [44]. CPP development in these proposals regards specific requirements, such as considering obstacles (e.g., buildings and antennas) in urban applications. Despite their focus, many of these studies are suitable for other applications with a few modifications. However, some studies do not focus on a particular application and aim to develop more generic solutions [214, 215, 217, 218, 222, 224, 225].

A significant limitation of some of these studies is that they do not consider any kind of energy constraint [44, 45, 46, 214, 216, 218, 219, 221, 223, 224, 225]. Energy is a critical issue for drones considering the limited battery. The energy is viewed in different ways by the CPP works in Table 4.9. Path optimization based on drone energy consumption is the most common way of considering energy [215, 220, 222]. This is the most efficient way of expending less energy because the algorithms search for solutions that the drone paths are more energetically economical. However, Ghaddar and Merei [217]

Table 4.9: Recent proposals related to UAV coverage path planning

Year	Ref.	Type	IoD	CPP Approach	Energy Approach	Application	Flight barriers
2020	[215]	Single		Genetic Algorithm	Path optimization considering the drone energy consumption	Generic	
2020	[216]	Multi		Genetic Algorithm		Search and Rescue	
2020	[217]	Multi		Grid-based Method	An energy-aware algorithm to avoid obstacles	Generic	Obstacles
2020	[218]	Single		Deep Reinforcement Learning		Generic	No-fly zone
2020	[219]	Single		Rotating Calipers Path Planner		Surveying	
2020	[220]	Multi		Mixed Integer Linear Programming	Path optimization considering the drone energy consumption	Aerial Imagery	
2021	[221]	Multi		Grid-based Method		Surveillance	Obstacles
2021	[214]	Multi		Fast Marching Square		Generic	Obstacles
2021	[45]	Multi		Mixed Integer Linear Programming		Search and Rescue	Obstacles
2021	[222]	Single		Heuristic	Path optimization considering the drone energy consumption	Generic	
2021	[223]	Single		Meta-heuristic		Structural Inspection	Obstacles
2022	[224]	Multi		Mixed Integer Linear Programming		Generic	
2022	[46]	Single		A Method Based on Back-and-forth Paths		Spraying	
2022	[225]	Multi		Ant Colony		Generic	
2022	**CPP-IoD**	**Multi**	✓	**Heuristic**	**Balancing the traveled distance between the drones**	**Generic**	**Airways**
2023	[44]	Multi		Genetic Algorithm		Precision Farming	
2023	**AntIoD**	**Multi**	✓	**Ant Colony**	**Path optimization considering the drone energy consumption**	**Generic**	**Airways**

developed a method to avoid obstacles considering energy.

IoD is a unique scenario that allows many drones in the sky to simultaneously perming different applications and services. One crucial aspect of the IoD is providing a safe and secure environment for many drones to fly [37, 226]. To achieve this, it is expected that an organization will be used to coordinate drone access to airspace. One possible way to organize this is through the use of airways. The use of airways has been considered in various proposals related to IoD [5, 226, 227, 228], and it is a common practice for organizing airspace for airplanes. Therefore, it is reasonable to assume that in large urban centers, airways will be utilized for better coordination, organization, and security of drone traffic in the context of IoD. Although some studies considered obstacles such as buildings and antennas or no-fly zones, none considered using airways. So, in this work, we propose an ant colony-inspired coverage path planning that considers an airway-based IoD to manage the drones.

4.3.1.2 CPP for Robotics

Traditionally, robot CPP algorithms involve dividing the target space into non-overlapping cells. There are various methods for achieving space division, including Morse-based cellular decomposition, landmark-based topological coverage, grid-based, and graph-based methods [209]. Many CPP methods have been developed for different scenarios, such as indoor [229, 230], outdoor [210], and 3D coverage [222, 231]. In our work, as we deal with airway constraints, we focus on comparable scenarios of CPP IoD with CPP robotics. Our analysis concludes that scenarios featuring road constraints are the most similar to the IoD scenario utilized in our work. Table 4.10 shows recent proposals that have employed road constraints in CPP for robotics.

Table 4.10: Recent proposals related to robots coverage path planning

Year	Ref.	CPP Approach	Energy	Barriers	Environment	Application
2021	[232]	Ant Colony		Avoid static and non-static obstacles	Road	Surveillance
2022	[231]	Metaheuristic			Parking lot, City crossing, Bridge	Autonomous Road Sweeping
2022	[233]	Metaheuristic		Avoid obstacles such as sidewalks, flowerbeds and trees	Parking lot	Autonomous Road Sweepers
2022	[234]	Ant Colony			Road	Patrol

Based on Table 4.10, the featured studies are applicable for scanning a region, including surveillance [232], patrol [234], and road sweeping [231, 233]. These scenarios can be divided into two types. In the first type, the robot only needs to travel along the central street to cover the entire region (patrol and surveillance). The second type must cover the whole region delimited by the road. For example, in the case of sweepers, the robot needs to clean the entire area of the street, not just the center. It should be noted that recent proposals that utilize road constraints for coverage do not prioritize energy optimization as a crucial factor for their applications. These studies often assume that the robots have sufficient energy capacity and, therefore, do not consider energy consumption during path planning. However, energy consumption is a critical requirement in IoD, making it unsuitable for applying these techniques to drones directly. Additionally, the unique characteristics of drones, such as their ability to move in three dimensions and collaborate, must also be considered.

4.3.1.3 CPP for IoD and Discussion

This work assumes that the IoD scenario is based on airways, meaning that drones can only navigate within predetermined airways. As a result, we focus on applications where airways are the only feasible route for drone flight. These airways are established to run parallel to landways and are kept free from stationary obstructions. However, mobile and temporary obstacles, such as the presence of animals, particularly birds, may occur within the airways. Moreover, the airways are directed, meaning that some situations require revisiting previously traveled paths to ensure complete coverage of the regions.

CPP for IoD poses unique requirements, combining elements of CPP for both drones and ground robots. Specifically, it must account for 3D movement and drone collaboration while accommodating road network constraints, including overlapping roads. Additionally, energy consumption is a crucial factor due to the limited battery life of drones. Traditional CPP drone techniques often rely on geometric approaches to divide space into cells [225], which are not necessary for the IoD context. Furthermore, the IoD scenario is highly dynamic, requiring adaptive algorithms such as bio-inspired techniques. To address these challenges, this work proposes an ant-colony optimization-based algorithm for CPP in IoD, considering the energy expenditure of drones.

4.3.2 System Model

The antIoD system is formally defined by the following notation.

- $G = \{N, A\}$ is the graph that describes the IoD flyable airspace, where N is a set of waypoint nodes, and A is a set of arcs;
- $D = \{1, \ldots, K\}$ is the set of drones such that K is the number of drones;
- e^k_{zxy} represents the energy coefficient of each pair of arcs $(z, x) \rightarrow (x, y)$, describing the energy required for a flight from node x to node y given the previously visited node z by drone k, (x, y) and $(z, x) \in A$, and $k \in D$. e^k_{zxy} is a sum of three factors $(eF^k_{xy} + eT^k_{zxy} + eV^k_{xy})$, where eF^k_{xy} is the energy coefficient for flights straight in the arc (x, y), eT^k_{xy} is the energy coefficient for make a turn between the arcs (z, x) and (x, y), and eV^k_{xy} is an energy coefficient for flights up or down in the arc (x, y). We assume that antIoD knows if there is a turn when transitioning from arc (z, x) to arc (x, y);

- $P^k = \langle \rho_0, \ldots, \rho_n \rangle$ is the path planning taken for drone $k \in K$, in which any $\rho \in P^k$ represents an arc $(x, y) \in A$, composing a sequence of arcs;
- $p^k_{zxy} \in 0, 1$ denotes a flight of a drone $k \in D$ from arc $(z, x) \in A$ to $(x, y) \in A$. If this flight occurs, p^k_{zxy} is set as 1, otherwise, 0.

Given this notation, antIoD wants to minimize Equation 4.1.

Minimize

$$\sum_{k \in D} \sum_{(x,y) \in A} \sum_{(z,x) \in A} e^k_{zxy} p^k_{zxy} \tag{4.1}$$

Subject to

$$\sum_{k \in D} \sum_{z \in N} p^k_{zxy} > 0 \ \forall (x, y) \in A \tag{4.2}$$

$$\rho^y_0 = \rho^x_1 \ \forall \langle \rho_0, \rho_1 \rangle \hookrightarrow P^k, \forall k \in K \tag{4.3}$$

$$P^x_0 = P^y_n \ \forall P^k, k \in K, n = |P^k| \tag{4.4}$$

$$p^k_{zxy} \in \{0, 1\} (\forall (x, y), (z, x) \in A, \forall k \in D) \tag{4.5}$$

The optimization goal of AntIoD, as stated in Equation 4.1, is to minimize the total energy required for a given mission, taking all drones into account. Equation 4.2 ensures that each arc in the graph must be visited at least once due to the constraints imposed by the airspace. Equation 4.3 guarantees that two consecutive arcs in a drone path must be transactable, meaning that there should be an edge $(z, x) \in A$ and another edge $(x, y) \in A$. Additionally, each drone path must start and end at the same depot to form a proper route, as expressed in Equation 4.4. Lastly, Equation 4.5 imposes the integrity constraint on p^k_{zxy}.

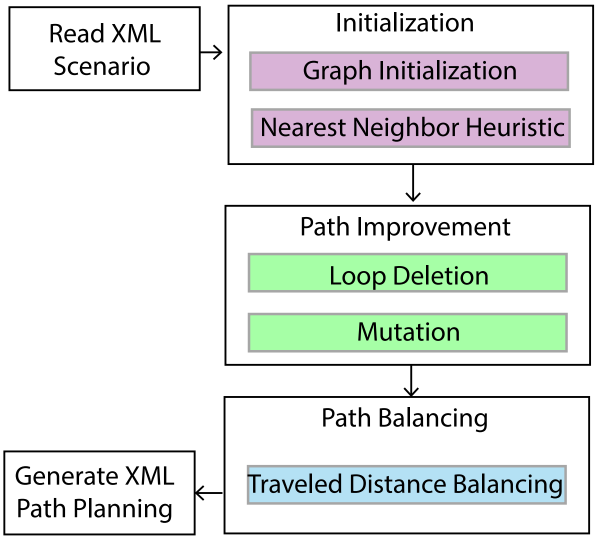
4.3.3 CPP Method

In this section, we propose a method of coverage path planning for IoD (CPP-IoD). Specifically, we introduce a technique that considers the IoD environment and has the objective of balancing the energy expenditure between the drones.

4.3.3.1 CPP Algorithm

This section describes the method used in CPP-IoD as depicted in Figure 4.7. In the first step, we read a scenario in XML format. The initialization step consists of generating the graph and creating a valid solution. We transform the input map into a directed graph in the initialization. Specifically, the intersections between the airways are the nodes, and the connections between them are the edges. Since the coverage problem is to traverse all edges, we change the graph for the edges to become the nodes. Next, we use the nearest neighbor heuristic to create a good solution.

Figure 4.7: CPP-IoD method overview



Source: Elaborated by the author

After initialization, we start the method of improving and balancing the paths. The improvement part comprises two processes: the deletion of loops and the mutation of existing paths. The improvement stage is responsible for standardizing the distances traveled by the drones. This step's purpose is to make the drones' paths similarly sized paths. Finally, the generated paths are saved in XML format. Each path represents the sequence of nodes for drones to follow in the map graph. The following sections detail the initialization, path improvement, and path balance steps.

- **Initialization**. The first part is to generate the map graph using the XML scenario. The idea is to construct all drone paths simultaneously using the nearest neighbor heuristic. Despite being a typical heuristic for creating paths, it is necessary to modify it to work with multiple drones. In addition, considering the IoD environment, we have the constraint that each node might be covered more than once to reach a given segment. Algorithm 2 presents the proposed heuristic.

Algorithm 2: Nearest neighbor heuristic for multiple drones

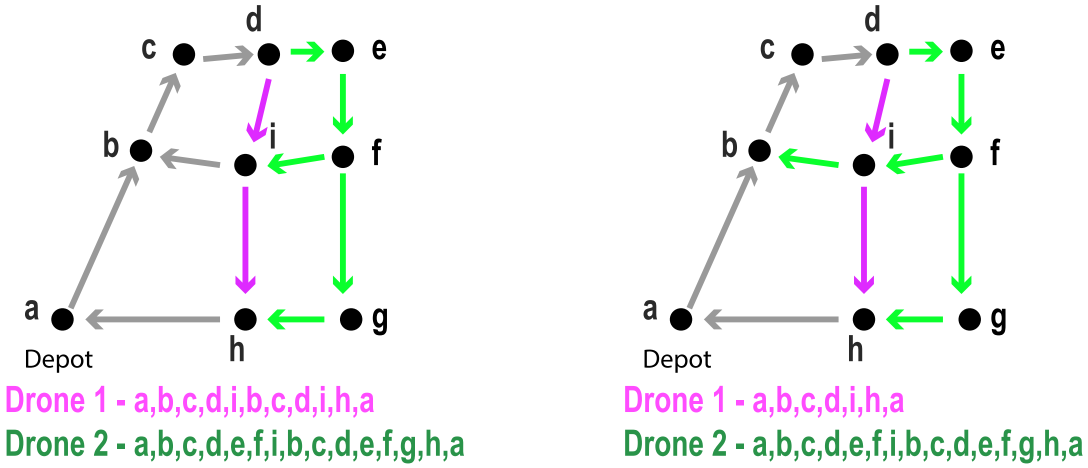
```
   Input  : numberDrones $D$, graph $G$
   Output: $paths$
1  $s \leftarrow randomNode(G)$
2  $C \leftarrow \{\}$
3  $C \leftarrow C \cup \{s\}$
4  $paths \leftarrow$ pathsInicialization$(D, s)$
5  while $(|C| \neq |G.N|)$ do
6      for $(p \in paths)$ do
7          if $(|C| \neq |G.N|)$ then
8              $currentNode \leftarrow p[p.size - 1]$
9              $nextNode \leftarrow$ getNearestNeighbor()
10             if $(nextNode)$ then
11                 $p \leftarrow p \cup \{nextNode\}$
12                 $C \leftarrow C \cup \{nextNode\}$
13             else
14                 $nextNode \leftarrow$ getNearestNode()
15                 $nextPath \leftarrow$ buildPath$(currentNode, nextNode)$
16                 $p \leftarrow p \cup \{nextPath\}$
17                 $C \leftarrow C \cup \{nextPath\}$
18 for $(p \in paths)$ do
19     $currentNode \leftarrow p[p.size - 1]$
20     $p \leftarrow p \cup$ buildPath$(currentNode, s)$
21 return $paths$
```

The algorithm works as follows. First, we randomly choose a node $s \in G$ to be the depot. Next, we initialize a path for each drone (Line 4), with node s added as the start of all paths. The algorithm continues until all nodes are traversed (Line 5). For each drone path (Line 6), we verify the nearest neighbor of the last node (Lines 8 and 9). Before, it is essential to check if existing nodes are available (Line 7). It is important to check if there are still nodes available before adding another one to each path, as the number of nodes may not be divisible by the number of drones. If a neighbor not yet covered exists (Line 10), it is added to the current path (Lines 11 and 12). If no neighbor is available, we choose the closest node not yet covered (Line 14).

As the chosen node is not adjacent to the current node, it is necessary to build the path from the current node to the selected node (Line 15). Thus, we add the shortest path between the current node and the chosen one to the current drone path (Lines 16 and 17). In the end, for all paths, we sum the shortest path to the last node included in the depot. The time complexity of this algorithm is described as follows. The conditional clause of Lines 7–17 has a procedure that follows the construction of a path (Line 15). This procedure uses the traditional Dijkstra's algorithm, with time complexity $\mathcal{O}(A+N \log N)$. This procedure dominates the operations in Lines 9 and 14, being both $\mathcal{O}(N)$. The loop structure of Line 6 iterates over all paths whose size is equal to the number of drones D. Hence, its time complexity is $\mathcal{O}(D(A + N \log N))$. The while loop of Line 5 occurs until the $|C| = |N|$. In the worst case, the size increase will occur one element by iteration. Therefore, the time complexity is given by $\mathcal{O}(ND(A + N \log N))$. Also, the iteration of Line 18 is dominated by the while of Line 5 since its time complexity is $\mathcal{O}(D(A+N \log N))$.

- **Loop Deletion**. After the initial path construction, it is possible to identify some loops in the path planning that are unnecessary as depicted in Figure 4.8. Upon reaching node i, the pink drone has two edges not yet traveled (ih and ib). Due to the greedy strategy, the pink drone makes a decision that leads to a loop (steps i, b, c, d, i). In the end, the green drone traverses the edge ib. Thus, the loop performed by the pink drone to cover the ib edge becomes unusable, and it is possible to delete it. Although the green drone also has loops, none is delectable as it always happens that an edge belonging to the loop is only traversed once. Therefore, deleting the loop would cause the solution to become invalid.

Figure 4.8: Loop deletion example



(a) Before loop deletion (b) After loop deletion

Source: Elaborated by the author

Algorithm 3 details the loop exclusion. We initialize the hash table R using the graph nodes as a key (Line 1). For each node, we count how many times the node is traversed in the solution (Lines 2 and 3). For each path, the loops are searched (Line 5). For each identified loop (Line 8), we check if its deletion is possible. Specifically, we scan each loop node verifying the amount in the node hash table (Lines 9–12). If it is 1, it cannot be deleted as the solution will become invalid. Otherwise, we delete the loop (Lines 13–15).

The time complexity of Algorithm 3 is described as follows. The loop of Lines 2–3 iterates over all the nodes, counting how many times the node appears in the paths.

Algorithm 3: Loop Deletion algorithm

```
   Input  : $paths$, graph $G$
   Output: $paths$
 1 $R \leftarrow$ hashTableInicialization($G$)
 2 for ($node \in G.N$) do
 3     $R[node]$ = countNodePaths($node, paths$)
 4 for ($p \in paths$) do
 5     $loops \leftarrow$ getLoops($p$)
 6     $exclude \leftarrow$ true
 7     $paths \leftarrow paths - \{p\}$
 8     for ($l \in loops$) do
 9         for ($node \in l$) do
10             if ($R[node] = 1$) then
11                 $exclude \leftarrow$ false
12                 break
13     if ($exclude$) then
14         $p \leftarrow$ "remove the loop $l$ from $p$"
15         $paths \leftarrow paths \cup \{p\}$
16 return $paths$
```

We have D paths, and each one has N^2 elements in the worst case. Thus, the loop time complexity is $\mathcal{O}(DN^3)$. Next, the time complexity of the loop of Lines 4–15. The *for* operation of Lines 4–15 iterates D times. The operation of Line 5 iterates over all elements of p, whose size is N^2 in the worst case. We will analyze the two inner loops of Lines 8 and 9 in an aggregate way. The worst case occurs when all path elements belong to a loop (always visited once). Hence, the time complexity is $\mathcal{O}(N^2)$ since the maximum number of elements in a path is N^2. Therefore, the total time complexity of Algorithm 3 is $\mathcal{O}(DN^3 + DN^2) = \mathcal{O}(DN^3)$. Although the time complexity has a high degree, we note that the main factor is the path length. However, the path length being close to N^2 will hardly occur in practical terms. In a real scenario, the airways are arranged so that a path between two nodes does not need to go through all existing nodes, considerably reducing the total path size.

► **Mutation**. Each path has nodes not deletable because they are covered by one track. However, the path between two unique nodes can change if other drones cover the intermediate nodes. Algorithm 4 illustrates the process of mutating a path. Initially, it is necessary to find out for each path which nodes are unique to each (Lines 1 and 2). Next, it checks for each pair of exclusive nodes if a path is shorter than the current one (Lines 6–8). If it exists, we verify if the new path between the pairs of vertices returns a valid solution (Lines 9 and 10). If positive, there is a change in the path (Line 11). The process continues for all unique node pairs and all paths.

The time complexity of Algorithm 4 is given as follows. We have in total D paths (Line 1). The operation of Line 2 iterates over all elements from the path, being $\mathcal{O}(N^2)$. The number of unique nodes is at most N. Therefore, the inner loops of Lines 3 and 4 will occur at most N^2. However, we must analyze the inner operations of Lines 5–11. The operation of Line 6 follows the time complexity of the Dijkstra algorithm, being $\mathcal{O}(A + N \log N)$. The operations of Lines 7, 9, and 10 iterates over all the elements of the path p. Thus, their complexity time is $\mathcal{O}(N^2)$. Hence, the total complexity time of the inner loops is $\mathcal{O}(N^2 N^2) = \mathcal{O}(N^4)$. This way, the time complexity of Algorithm 4 is $\mathcal{O}(DN^4)$. Again, the main factor affecting the algorithm's complexity is the path length N^2. However, as the graph represents an actual map, the path between two nodes will always be shorter than traversing all the map nodes. For instance, if someone wants to go from one intersection to another, it is unnecessary to go through all the city's crossings.

Algorithm 4: Mutation algorithm

```
    Input  : $paths$
    Output: $paths$
 1  for ($p \in paths$) do
 2      $U \leftarrow$ checkUniqueNodes($p$)
 3      for ($i \in U$) do
 4          for ($j \in U$) do
 5              if ($i \neq j$) then
 6                  $newDist \leftarrow$ shortestPath($i, j$)
 7                  $currentDist \leftarrow$ distancePath($p, i, j$)
 8                  if ($newDist < currentDist$) then
 9                      $newPath \leftarrow$ changePath($p, i, j$)
10                      if ($checkAnswer(newPath, paths)$) then
11                          $p \leftarrow newPath$
12  return $paths$
```

► **Path Balancing**. The last step of the algorithm is to balance the path taken by each drone. Algorithm 5 details path balancing, an essential process that helps balance energy consumption. In addition, the coverage process becomes faster, as the maximum time depends on the drone that travels the greatest distance. For this, we assume that the speed of drones is similar, which is reasonable given this scenario. The balancing algorithm works as follows. Initially, the algorithm finds the longest and shortest paths (Lines 2 and 3, respectively). Then the algorithm creates a new set of paths with all drone trajectories except the smallest and largest ones (Line 5).

Next, we discover the loops in the longest path *longerPath* (Line 7). Loops are best suited for transferring the path from the *longerPath* drone to the *shortestPath* drone, as

Algorithm 5: Distance traveled balancing algorithm

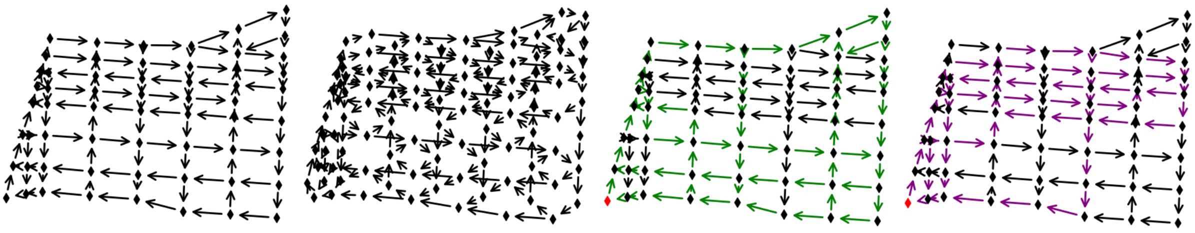
```
   Input  : $paths$, $maxUpdate$
   Output: $paths$
1  while ($cont < maxUpdate$) do
2      $longerPath \leftarrow \max(paths)$
3      $shortestPath \leftarrow \min(paths)$
4      $newPaths \leftarrow \{\}$
5      $newPaths \leftarrow paths - shortestPath - longerPath$
6      $loops \leftarrow \mathrm{getLoops}(longerPath)$
7      $distanceMax \leftarrow (longerPath.distance - shortestPath.distance)/2$
8      $CP \leftarrow \{\}$
9      for ($l \in loops$) do
10         if ($l[0] \in shortesPath$) then
11             $dist \leftarrow \mathrm{computeDistance}(l)$
12             if ($dist < distanceMax$) then
13                 $CP \leftarrow CP \cup \{l\}$
14     $p \leftarrow \mathrm{getPathLonger}(CP)$
15     $newPathA \leftarrow \mathrm{add}(p, shortesPath)$
16     $newPathB \leftarrow \mathrm{delete}(p, longerPath)$
17     $newPaths \leftarrow newPaths \cup newpathA \cup newpathB$
18     if ($newPaths.std < paths.std$) then
19         $paths \leftarrow newPaths$
20     else
21         break
22     $cont \leftarrow += 1$
23 return $paths$
```

it is only necessary to check that the first node of a loop found in *longerPath* is part of the path in *shortestPath*. The algorithm checks this step for each loop found (Lines 9 and 10). The maximum distance the loop can have is half the difference between the length of the longest and shortest paths (equation in Line 7). Thus, all loops that fit the rules are added to the *CP* list (Lines 11, 12, and 13). Next, the longer path is added to *shortestPath*, creating *newPathA*, and deleted from *longerPath*, creating *newPathB* (Lines 16 and 17). Finally, the algorithm checks whether the new paths returned a smaller standard deviation than the old ones (Lines 17 and 18). If the standard deviation is smaller, the new paths are replaced with the original trajectories. The algorithm continues until it is no longer possible to balance or until a maximum number of retries is reached. The complexity of Algorithm 5 is given as follows. The operation of Line 6 returns all loops in the longer path, being $\mathcal{O}(N^2)$. The loop of Lines 9–13 has time complexity $\mathcal{O}(N^2)$ since we can check if a node is in the shortest path in $\mathcal{O}(1)$. The operations of Lines 15–17 has time complexity $\mathcal{O}(N^2)$. Therefore, the Algorithm 5 complexity is dominated by $\mathcal{O}(maxUpdate \times N^2)$, where *maxUpdate* is the maximum number of times the algorithm can try to update the balance of the paths.

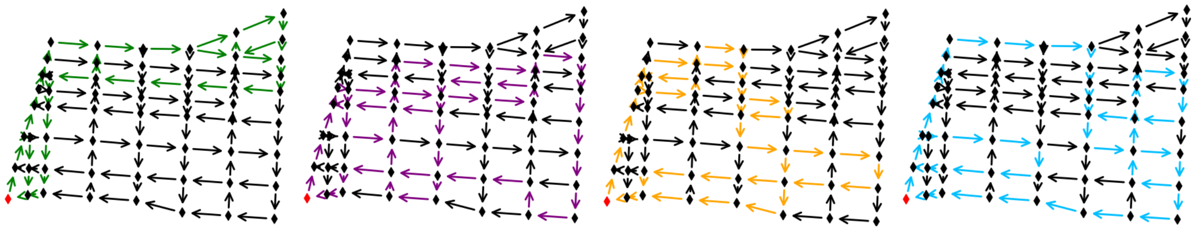
4.3.3.2 Performance Analysis

To simulate this work, we used part of the neighborhood of Ipanema beach, Rio de Janeiro, Brazil. Figure 4.9 illustrates some resulting graphs. Figure 4.9a represents the directed graph created from the original street directions. The scenario used has approximately 15,150 m of airways. As the intention is to cover the airways, Figure 4.9b represents the inverted graph transformation, in which the airways become the nodes. We perform the tests considering 2 to 10 drones. For each number of drones used, 30 simulations were performed, varying the depot location.

Figure 4.9: Graph used (a,b), path planning examples for two drones (c and d) and four drones (e, f, g, and h)



(a) Graph of Ipanema (b) Inverted Graph (c) 2 drones ex. - d 1 (d) 2 drones ex. - d 2



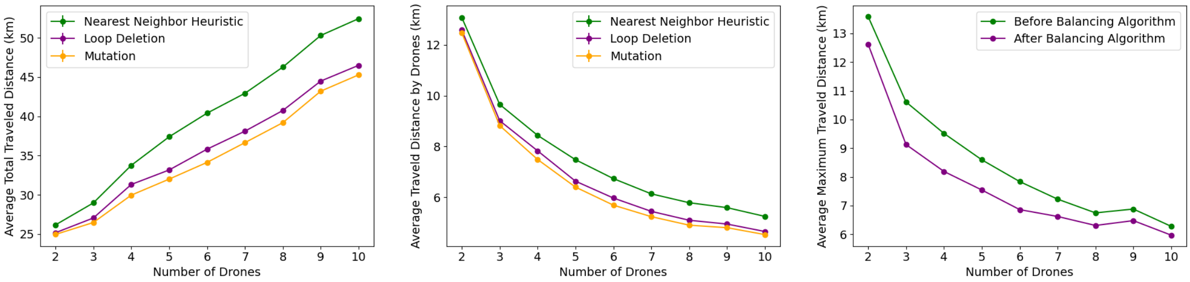
(e) 4 drones ex. - d 1 (f) 4 drones ex. - d 2 (g) 4 drones ex. - d 3 (h) 4 drones ex. - d 4

Source: Elaborated by the author

Figure 4.9 illustrates two examples of path planning results. The first one is the CPP for two drones. Figure 4.9c shows the path planning result for the first drone and Figure 4.9d for the second one. The first example shows that drone 1 has most of its path at the bottom of the map and drone 2 at the top. The second example is the CPP for four drones. Figures 4.9e, 4.9f, 4.9g, and 4.9h represent the drones 1 to 4, respectively. The red dot represents the depot.

- **Path Planning Analysis**. Figure 4.10 presents the results of the analysis carried out for the generation of the CPP and the path balancing. To evaluate the path planning, we considered two metrics: average total traveled distance and average distance traveled by drone. Figure 4.10a shows the results of the total average traveled distance considering all drones used. We observe that the more drones we add, the greater the total path traveled. Another point is that for all cases, the removal of loops decreases by an average of 9.59% (see Table 4.11) compared to the nearest neighbor heuristic. Next, we applied the mutation algorithm after removing loops. With this application, the paths decreased by an average of 12.46%, as shown in Table 4.11, compared to the baseline.

Figure 4.10: CPP-IoD results



(a) Total traveled distance (b) Distance traveled by drone (c) Max. traveled distance

Source: Elaborated by the author

Table 4.11: Reduction of distance traveled

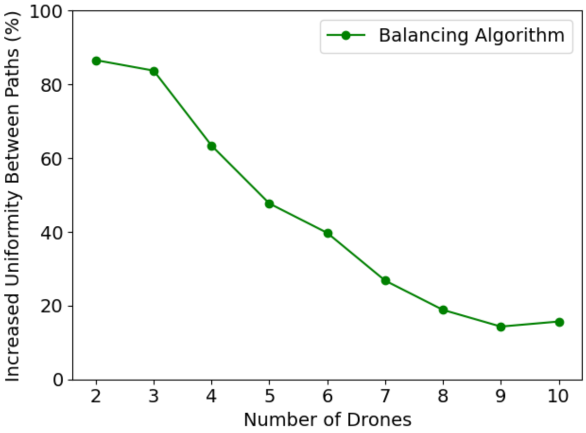
# Drones	**Loop deletion (%)**	**Mutation (%)**
2	3.80	4.64
3	6.63	8.62
4	7.22	11.25
5	11.30	14.42
6	11.35	15.52
7	11.24	14.64
8	11.94	15.30
9	11.55	14.11
10	11.34	13.65
Average	**9.59**	**12.46**

Figure 4.10b shows the average distance traveled by drones. This result indicates that by adding drones, the length of the path taken by each drone decreases. The increase of drones can lead to better coverage in a shorter time and enable drones with greater energy capacity to perform spatial applications. However, the overall cost of the application becomes higher as more drones are needed. Furthermore, considering an IoD scenario in which other applications use the airways, more drones can congest the airways. Thus, the number of drones used depends on the application and purpose. Applications related to safety, health, and saving human lives may require a greater number of drones, not being relevant to the higher economic and energy cost.

- **Path Balance Analysis**. The second step of the CPP-IoD is to balance the length of the path taken by each drone. Figure 4.10c shows the longest existing path before and after balancing. Since drone speeds are similar, the total mission time tends to be similar to the time needed to travel the longest path. Therefore, decreasing the longest path diminishes the full-time drones take to cover the delimited area. Consequently, the increased path uniformity represents a distribution of energy consumption between the drones.

Figure 4.11 presents the metric of how much path uniformity increased after using the balancing algorithm. The greater the number of drones, the greater the difficulty in balancing the paths, i.e., there is a greater difficulty in finding trajectories suitable to be transferred to other drones. Thus, in this scenario, the rate increases more than 80% by using up to three drones. However, the uniformity rate of seven or more drones increases by less than 30%.

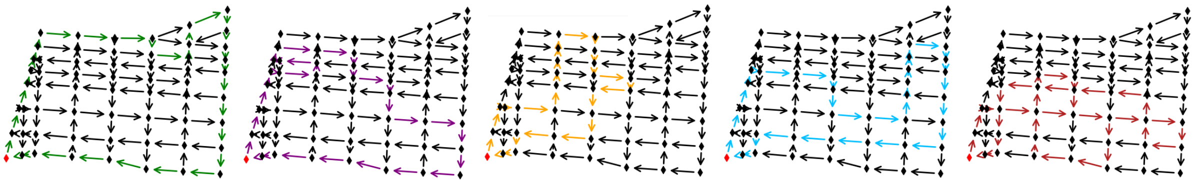
Figure 4.11: Increased uniformity between paths



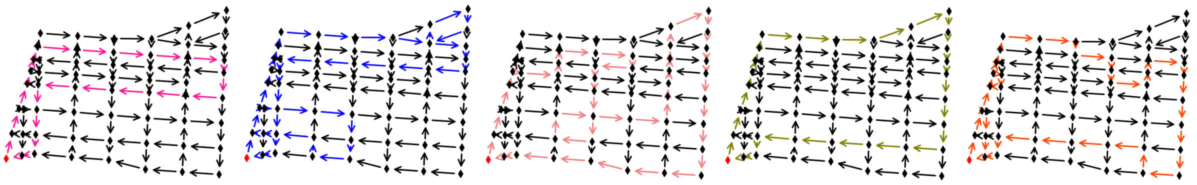
Source: Elaborated by the author

Figure 4.12 presents an example of CPP for ten drones. We observe that the balance of distance through loops is more challenging to be performed. This is because some paths do not have secondary loops, such as drone 1 (see Figure 4.12a), which performs a circuit from the depot until it returns to the same exit point. Another observation in this example is that the more drones, the more edges are repeated, such as the arrival and departure edges of the depot used by all drones. The edge repetition also happens because the edges are directed. In future work, we will carry out an analysis of undirected and overlapping airways.

Figure 4.12: Path planning example for ten drones before path balancing



(a) drone 1 - 4220.83 m (b) drone 2 - 4246.88 m (c) drone 3 - 2908.67 m (d) drone 4 - 4156.48 m (e) drone 5 - 5366.82 m



(f) drone 6 - 3640.59 m (g) drone 7 - 6089.71 m (h) drone 8 - 5909.42 m (i) drone 9 - 3661.45 m (j) drone 10 - 5096.75 m

Source: Elaborated by the author

4.3.4 AntIoD Method

In this section, we propose an Ant Colony Optimization (ACO) Algorithm for CPP in IoD (AntIoD), considering energy consumption. Our goal is to enhance our previous work using ACO metaheuristic.

4.3.4.1 Energy Consumption Model

Abeywickrama et al. [235] proposed an energy model for drones used in this work. They conducted experiments using an Intel Aero Ready to Fly Drone with Dualsky 4000 mAh 4S 25c, ECO-S LiPo batteries to develop an energy consumption model (ECM). The ECM includes on-ground power consumption (idle and armed mode), take-off, horizontal and vertical movements (upward and downward), hovering, and the payload. However, in this work, hovering is not required for any movement. Therefore, the energy consumption model for the IoD scenario only considers horizontal and vertical movements. It is also assumed that all drones require the same amount of energy to take off, and this value is not included in the energy optimization since it is only performed once. Hence, Equation 4.6 represents the energy consumption for a drone during a straight flight (horizontal movement).

$$E_{straight} = 308.709t - 0.852 \tag{4.6}$$

where t represents the time and $E_{straight}$ represents the energy in joules. Equations 4.7 and 4.8 represent the energy in joules to climbing and descending, respectively.

$$E_{up} = 315d_1 - 211.261 \tag{4.7}$$

$$E_{down} = 68.956d_2 - 65.183 \tag{4.8}$$

where d_1 represents the vertical distance traveled upward in meters, and d_2 is the distance traveled downward. We also consider that all drones will have the same payload, and, thus, for this reason, this expenditure was not considered. To summarize, to design a coverage path planning in an airway-based IoD, the most important movements are flinging straight, down, and up. Another important point is turning. Turning can expend a significant amount of energy. Abeywickrama et al. [235] did not consider the expense of turning a drone. However, For this reason, we consider the turn model of [236]. The value of E_{turn} is given by Equation 4.9.

$$E_{turn} = P_{turn}\frac{\Delta\theta}{\omega_{turn}} \tag{4.9}$$

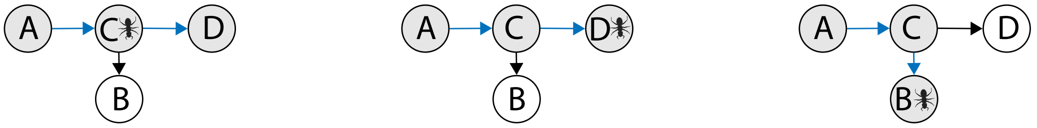
where P_{turn} is equal to 225 W /s, ω_{turn} is 2.1 rad /s, and $\Delta\theta$ is the angle of the turn. Although we have considered the energy consumption values, using other energy models as input in the AntIoD algorithm is possible.

4.3.4.2 Mathematical Formulation

AntioD can simultaneously control the movement of K drones, each represented by an ant. To ensure that all edges are covered while adhering to directional airway constraints, ants may need to reuse edges that they or other ants have previously traversed. Moreover, to address energy consumption concerns, the previous vertex is crucial in determining whether a drone will change direction. For instance, if ant F is located at node C (see Figure 4.13a), there are two possible paths to take: edge $(C, D) = path_1 = \{A, C, D\}$ (Figure 4.13b) or edge $(C, B) = path_1 = \{A, C, B\}$ (Figure 4.13c). While traversing $path_2$, it is essential to account for the energy required to turn in edge (C, D).

It is only possible to know that a curve will occur by knowing the previous vertex that the forming traversed. This situation affects the probability that an ant selects the

Figure 4.13: Example of turn influence in a path



(a) Ant F in node C (b) Ant F in node D (c) Ant F in node B

Source: Elaborated by the author

next node. Thus, the P^k_{zxy} transition probability equation that ant k selects the next vertex y from the vertex x considering the ant already taken vertex z is defined by Equation 4.10.

$$P^k_{zxy} = \frac{[\tau_{zxy}(t)]^\alpha [\eta_{zxy}(t)]^\beta}{\sum_{l \in N^k_{zx}} [\tau_{zxy}(t)]^\alpha [\eta_{zxy}(t)]^\beta} \tag{4.10}$$

where $\tau_{zxy}(t)$ is the pheromone quantity of the edge (x, y) given that the ant has already traversed the edge (z, x) in the iteration t and $\eta_{zxy}(t)$ is the pheromone value. N^k_{zx} is a set of vertices where an ant k can move from vertex x given that ant traversed the node z. The pheromone update after each t iteration is given by Equation 4.11 and Equation 4.12.

$$\tau^k_{zxy}(t) = (1 - \sigma)\tau^k_{zxy}(t - 1) + \sum_{k=1}^{K} \Delta\tau^k_{zxy}(t) \tag{4.11}$$

$$\Delta\tau^k_{zxy}(t) = \begin{cases} \frac{1}{E_k} & (z, x, y) \in p_k \\ 0 & \text{otherwise} \end{cases} \tag{4.12}$$

where σ is the pheromone evaporation coefficient, p_k is the path of ant k, $\Delta\tau^k_{zxy}(t)$ represents the pheromone deposited in edge (x, y) given edge (z, x) by ant k in iteration t, and E_k is the energy cost of the path constructed by ant k. E_k is calculated considering the equations in Section 4.3.4.1.

4.3.4.3 AntIoD Algorithm

Algorithm 37 details the ACO considering energy for IoD. The algorithm takes in a graph G and an initial pheromone value λ, then initializes the pheromone table *tableP* (Line 1). Each triple $\langle z, x, y\rangle$ in the table has a pheromone value and a selection probability. The while loop in Line 4 controls the number of iterations the ACO algorithm runs for, and the algorithm terminates if there is no improvement in the result after T iterations.

Each drone path starts with a randomly chosen edge (Lines 7–10). Set C keeps track of all edges that have been traversed. The algorithm ensures that every drone takes

Algorithm 6: AntIoD

```
    Input  : numberDrones $D$, numberIterations $T$, graph $G$, initialPheromone $\lambda$, evaporationCoefficient $\sigma$
    Output: $paths$
 1  $tableP \leftarrow$ tablePheromoneInicialization($G, \lambda$)
 2  $bestPaths \leftarrow \{\}$
 3  $count \leftarrow 0$
 4  while ($i < I$ or $count < T$) do
 5      $paths \leftarrow \{\}$
 6      $C \leftarrow \{\}$
 7      for ($k \in D$) do
 8          $s \leftarrow$ randomEdge($G$)
 9          $paths[k] \leftarrow paths[k] \cup s$
10          $C \leftarrow C \cup s$
11      while ($|C| \neq |G.E|$) do
12          for ($k \in D$) do
13              if ($|C| \neq |G.E|$) then
14                  $x \leftarrow$ lastNode($path[k]$)
15                  $possibleEdges \leftarrow \{\}$
16                  for ($y \in G.adj[x]$) do
17                      if ($y \notin C$) then
18                          $possibleEdges \cup y$
19                  if ($possibleEdges \neq \varnothing$)
20                   then
21                      $ChosenEdge \leftarrow$ RWS ($path$[k], $possibleEdges$, $tableP$)
22                      $path[k] \leftarrow path[k] \cup ChosenEdge$
23                      $C \leftarrow C \cup ChosenEdge$
24                  else
25                      $W \leftarrow$ BFS($path[k], C$)
26                      $ChosenPathEdge \leftarrow$ RWS($path[k]$, $W$, $tableP$)
27                      $path[k] \leftarrow path$[k] $\cup$ $ChosenPathEdge$
28                      $C \leftarrow C \cup ChosenPathEdge$
29      $paths \leftarrow$ buildPath($paths$)
30      if ($sum(paths) < sum(bestPaths)$) then
31          $bestPaths \leftarrow paths$
32          $count \leftarrow 0$
33          $tableP \leftarrow$ tableUpdate($G, paths, \sigma, tableP$)
34      $count \leftarrow count + 1$
35      $i \leftarrow i + 1$
36  return $bestPaths$
37  Source: Elaborated by the author
```

at least one unused edge. The while loop in Line 11 controls if all the edges were added in set C. For each drone k (Line 12), the algorithm gets the last node in its path (Line 14). In Line 15, the algorithm checks if the node has any adjacent edges that any drone has not traversed. If such an edge exists, it adds it to the set *possibleEdges* (Lines 16–18). The algorithm selects an edge from the set *possibleEdges* for the drone k. The selection uses a roulette wheel selection, which considers the last node, the set of *possibleEdges*, and the pheromone table *tableP* (Line 21). The edge selected is added in set C (Line 22).

If no adjacent edges are found, a Breadth First Search (BFS) is performed to find all edges at the same distance as the last node added to the path of drone k, and these edges are added to a set W (Line 25). The new edge $w \in W$, $w \notin C$ is selected using the pheromone table and roulette wheel selection, considering all nodes and edges from the last node to the new edge w (Lines 26–28). This selection will result in a path from the last node added to the new edge w (*ChosenPathEdge*). The drone k path and the set C are updated, and if the current solution is better than the current best solution, *tableP* is updated with new pheromone values and probabilities based on the selected edges (Line 29). Finally, the best path is returned when the algorithm terminates.

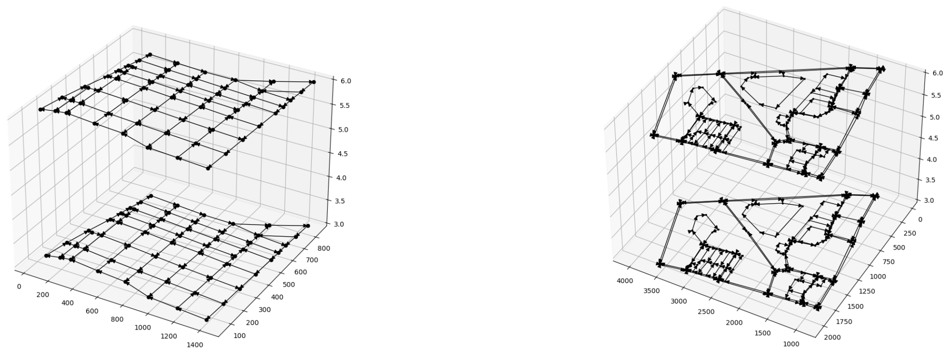
If the current node has no untraversed adjacent edges, we use a Breadth First Search (BFS) to discover nearby unexplored edges close to the last node added to the path of drone k (Line 25). We consider all edges equidistant to the last node added to the path of drone k and add them to a set W. Using the table of pheromones, we choose a new edge $w \in W$, such that $w \notin C$. In Line 26, we make this selection using a roulette wheel based on the probabilities of each edge $w \in W$ being selected in the table of pheromones. Specifically, we consider the entire path from the last node to the new edge w to calculate the probability of w being selected. Then, we update the path of drone k and set C (Lines 27–28). In Line 29, for all paths, we sum the energetically least costly path from the last included node to the source depot using the Dijkstra algorithm. At the end (Line 30), we check if the current solution is better than the current best solution. If it is, we update the *tableP* with new values of pheromones and probabilities based on the evaporation coefficient σ and the edges selected in the paths. Finally, the best path is returned at the end of the algorithm.

The time complexity of Algorithm 37 can be analyzed as follows. The outer while loop in Line 4 runs for I iterations. The while loop in Line 7 iterates over the set of drones $K = D$, which has a maximum size of $|K|$. Lines 11–13 are executed $|E|$ times in total. Within this loop, the for loop in Line 12 iterates over all adjacent nodes to the current node, which has a maximum size of $|N|$. In the worst case, the BFS in Line 25 takes $O(N + E)$ time to find all adjacent nodes to the current node that have not yet been visited. To compute the return path for each drone using Dijkstra's algorithm in Line 29, the time complexity is $K\mathcal{O}(E + N \log N)$. Therefore, the total time complexity of the algorithm is $\mathcal{O}(I(K + E(N + E) + K(E + N \log N)))$, which can be simplified to $\mathcal{O}(I(E^2 + E(N + \log N)))$. Since the number of edges E dominates the number of nodes N and drones K, we can simplify the time complexity to $\mathcal{O}(E^2)$.

4.3.4.4 Simulation Setup

AntIoD was subjected to evaluation in two distinct scenarios. In the first scenario, a section of the Ipanema beach neighborhood in Rio de Janeiro, Brazil, was utilized (Figure 4.14a). The Ipanema neighborhood was selected due to its resemblance to a grid-like structure, a common characteristic among many cities. The second scenario involved the Riverside and Ridgemont neighborhoods in Ottawa, Ontario, Canada (as depicted in Figure 4.14b). This region was chosen specifically because it does not exhibit a grid-like topology, thus enabling the assessment of AntIoD's performance in different situations. In both scenarios, two overlapped altitudes were used for the airways.

Figure 4.14: Scenarios overview



(a) Ipanema, Rio de Janeiro, Brazil with two altitudes of airways

(b) Riverside neighborhood in Ottawa, Ontario, Canada with two altitudes of airways

Source: Elaborated by the author

The drones must use edges connecting the different altitude levels to transition between altitudes. Each node was connected to both altitude levels via a direct edge, but certain edges were designated for drones ascending or descending. The direction of each edge was randomly assigned at the start of each simulation, and the same scenarios were utilized throughout all simulations conducted. We conducted 30 simulations of each algorithm for both scenarios, and the results were determined based on the average value of these 30 runs, with a 95% confidence interval.

Table 4.12 provides detailed information on the parameter values used in AntIoD, which were also utilized in the ACO comparison algorithm. These parameter values were consistent across all simulations. The energy consumption was calculated using the equations presented in Section 4.3.4.1. The drone speeds were set at 3 m/s to conform to the constraints of this model. Additionally, the first altitude was established at 3 m, while the second altitude was set at 6 m. However, for other ECMs that do not have altitude or speed constraints, AntIoD can be customized to operate at any altitude or drone speed. Each drone was initialized on a different node and must return to its origin node. However, we can easily modify the algorithm so that all drones begin the mission at the same depot (only one position is randomly chosen for all drones to start) or end at different depots (the return path is not added to the solution).

Table 4.12: ACO paramenters

Parameter	Symbol	Value
Pheromone Initial	λ	0.1
Evaporation Coefficient	σ	0.001
Iterations	I	3000
Threshold	T	1000

4.3.4.5 Performance Analysis

This section analyzes AntIoD compared to recent algorithms adapted for the IoD scenario. Firstly, we outline the algorithm used for comparison and the metrics employed. Subsequently, we discuss the results of the energy consumption of the entire path, turn movement energy consumption, and vertical movement energy consumption. To perform our simulations, we use Python 3.9.13. We compare AntIoD with three other algorithms, as detailed below.

- ACO: the ACO comparison method employed in this study was based on the approach presented in [234]. This choice was made because Fan et al. [234] proposed a multi-robot path planning approach that considers constraints imposed by the road network, making it similar to the IoD scenario. However, they used an undirected graph and assumed that two robots could not occupy the same node simultaneously. To adapt this algorithm to the IoD scenario, we modified it to use a directed graph and allow multiple nodes to be occupied at the same time since our objective is only to construct the path offline;
- BA*: BA* is based on Boustrephedon motions, a kind of zigzag motion [231]. We chose this heuristic because it is widely used in constructing metaheuristics for CPP. We adapted the idea proposed in [231] for the IoD scenario. For comparison, we did not use the Inward Spiral stage, as we consider that it does not apply to the airway scenario.
- CPP-IoD: We compared AntIoD and CPP-IoD which also focuses on the IoD scenario. We selected this work as it has the most comparable scenario for our evaluation. However, our previous work does not address energy optimization based on path optimization, which is the main focus of our proposed algorithm.

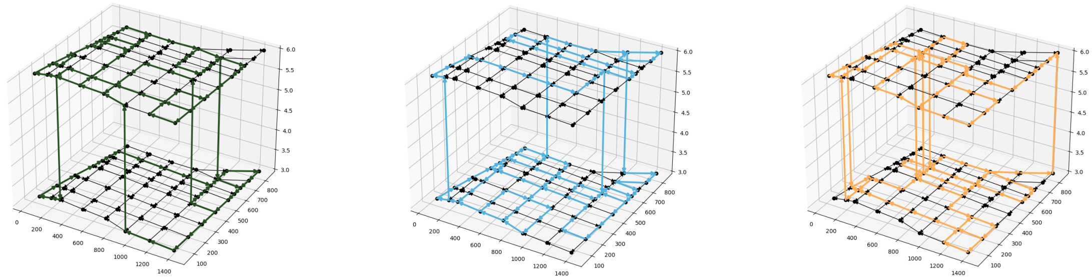
Our performance evaluation primarily focuses on analyzing the energy consumption of drone movement. Specifically, our results are based solely on the energy expenditure associated with drone displacement and do not consider other potential energy costs that may arise during a drone mission. For the performance evaluation, the following metrics were considered:

- Average path planning energy consumption: evaluates the average energy consumed by all drones in a cover mission;
- Average path planning energy consumption per drone: evaluates the average energy consumed by drones in a cover mission;
- Average turning movement energy consumption: evaluates the average energy consumed for the turning movement by all drones in a cover mission;
- Average number of turning movements: evaluates the average number of turns made for all drones in a cover mission;
- Average vertical movement energy consumption: evaluates the average energy for drones flight upward and downward consumed by all drones in a cover mission;
- Average number of vertical movements: evaluates the average number of times that drones change airways in a cover mission.

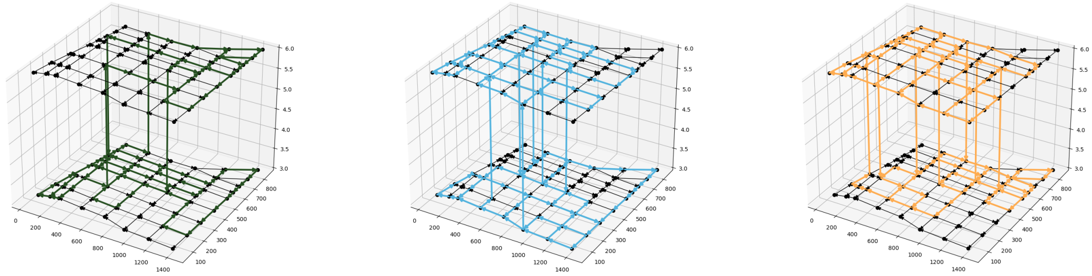
► **Path Planning Energy Consumption**. This section presents an energy consumption analysis for a coverage mission in two scenarios, namely, Ipanema and Riverside. The number of drones used for the mission varied from 2 to 10. Figure 4.15 illustrates an example of the result for all algorithms in the Ipanema scenario considering 3 drones. Upon a detailed analysis of the results, specific characteristics were observed in the generated paths by the algorithms, as depicted in Figure 4.15. The ACO algorithm faced challenges in generating paths with fewer edge repetitions and showed a frequent movement of descent or ascent. While the BA* algorithm produced satisfactory results for drone 1, it struggled to find efficient paths for the other drones due to the unavailability of easy routes for zigzag movement. The CPP algorithm displayed a higher frequency of turning movements compared to the first generated path to the other paths.

Figure 4.16a illustrates the results obtained for the Ipanema scenario. We observed that the ACO algorithm had the worst performance compared to the other algorithms for varying numbers of drones. This can be attributed to the fact that the ACO algorithm selects the next edge to be covered based on a random selection, which may result in suboptimal paths. Moreover, the CPP-IoD algorithm showed a rapid increase in energy consumption as the number of drones increased, surpassing the ACO algorithm's performance. This is likely due to the restrictions in the scenario, which make it challenging for the CPP algorithm to operate efficiently, such as the scenario being almost a complete grid.

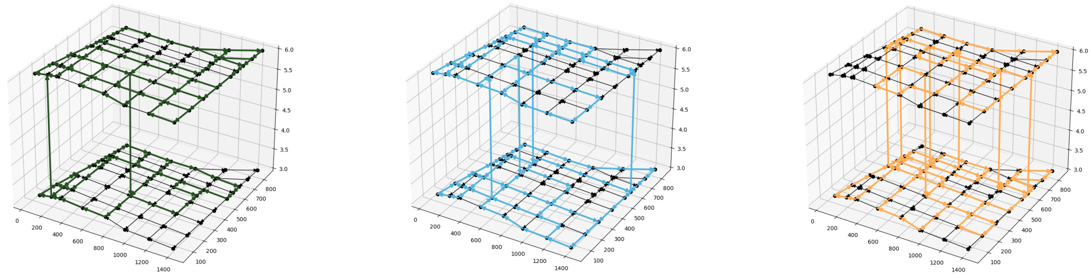
Figure 4.15: Example of results for Ipanema using 3 drones



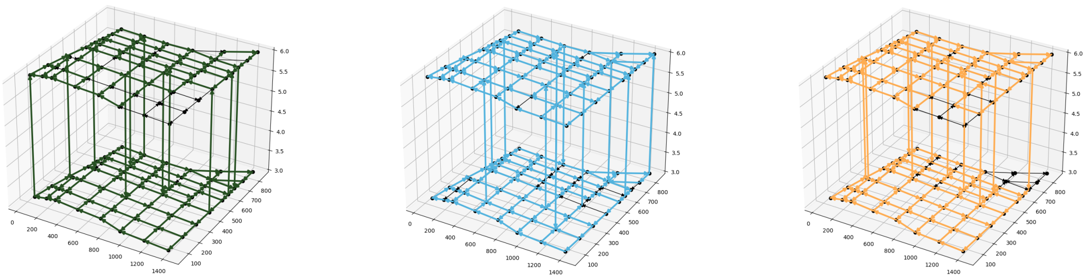
(a) AntIoD - Ant 1 (b) AntIoD - Ant 2 (c) AntIoD - Ant 3



(d) BA* - Ant 1 (e) BA* - Ant 2 (f) BA* - Ant 3



(g) CPP-IoD - Ant 1 (h) CPP-IoD - Ant 2 (i) CPP-IoD - Ant 3

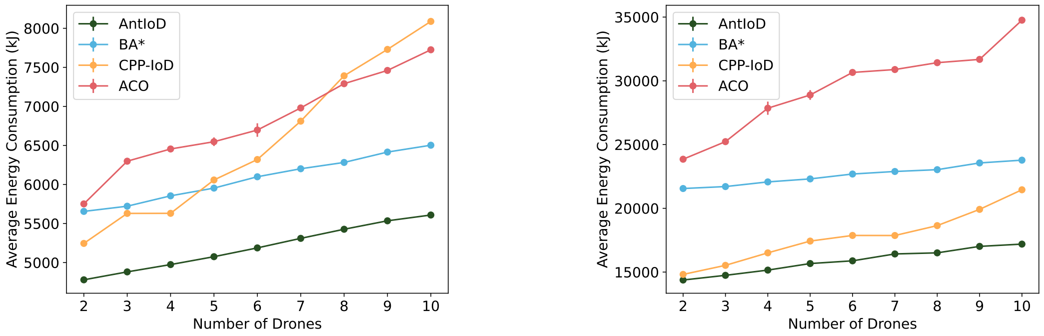


(j) ACO - Ant 1 (k) ACO - Ant 2 (l) ACO - Ant 3

Source: Elaborated by the author

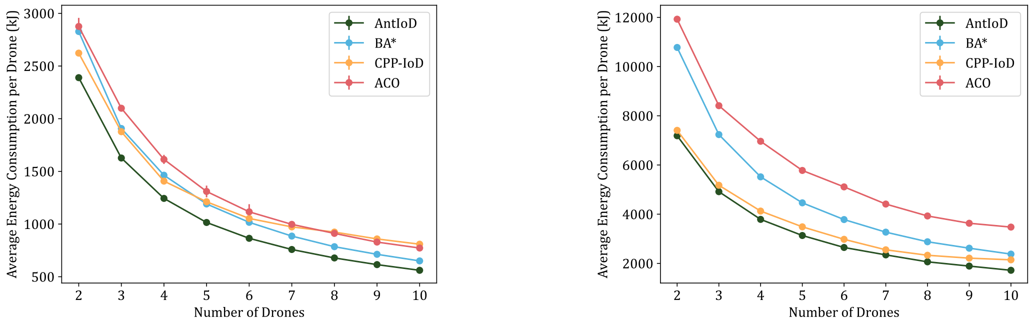
On the contrary, this scenario proves to be favorable for the performance of BA* since it allows the construction of zigzag paths to cover the designated area efficiently. However, neither ACO and BA*, nor CPP-IoD algorithms consider the energy consumption involved in horizontal, vertical, and turning movements during path construction. In contrast, our proposed algorithm outperforms all other algorithms in all drone tests conducted in the Ipanema scenario since it considers energy expenditure in its optimization process.

Figure 4.16: Path planning results for both Ipanema and Riverside scenarios



(a) Average energy consumption (kJ) × number of drones in Ipanema

(b) Average energy consumption (kJ) × number of drones in Riverside



(c) Average energy consumption per drone (kJ) × number of drones in Ipanema

(d) Average energy consumption per drone (kJ) × number of drones in Riverside

Source: Elaborated by the author

Figure 4.16b illustrates the results obtained for the Riverside scenario. In contrast to the Ipanema scenario, the Riverside scenario is unsuitable for the BA* algorithm because it does not have a grid structure, making it challenging to generate zigzag paths. However, the CPP-IoD algorithm performed better in this scenario because it prioritizes finding the nearest neighbor, and there are fewer nodes to cover. On the other hand, the ACO algorithm had the worst results. This occurs because the ACO does not use energy as a parameter and randomly chooses new areas to be covered. The AntIoD algorithm consistently outperformed all other algorithms for all numbers of drones.

Figures 4.16c and 4.16d show the average energy expenditure by drones in Ipanema and Riverside respectively. We observed that as the number of drones increased, the average energy expenditure per drone became closer. In the Riverside scenario, the difference in the average energy expenditure per drone generated by the AntIoD algorithm is more pronounced when compared to the other algorithms. This is because the Riverside scenario has specific design features, such as the mapping of road crossings, and is different from a grid-like structure.

The average improvement for each algorithm concerning AntIoD is tabulated in Table 4.13. A significant improvement of 45.95% was observed for the Riverside scenario compared to the ACO algorithm. On the other hand, a smaller improvement of 10.15% was observed for the CPP algorithm in the Riverside scenario. However, all the algorithms resulted in an improvement in the energy consumption for the given mission.

Table 4.13: Reduction of energy consumption

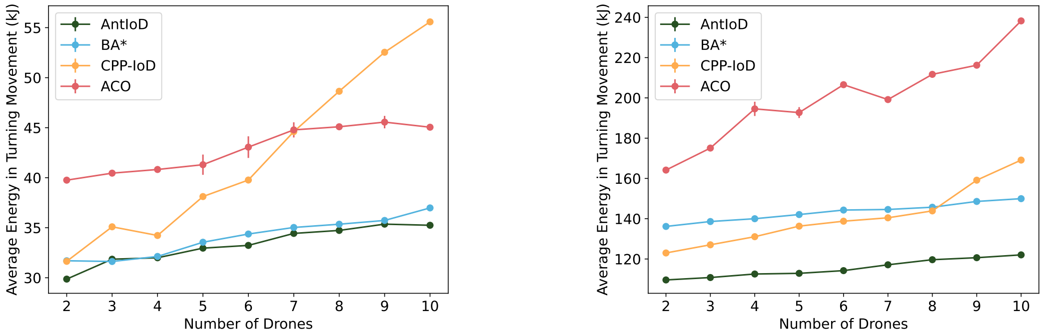
# Drones	**Ipanema (%)**			**Riverside (%)**		
	BA*	CPP-IoD	ACO	BA*	CPP-IoD	ACO
2	15.49	8.89	16.90	33.30	2.96	39.72
3	14.70	13.30	22.51	32.07	5.04	41.56
4	15.04	11.65	22.94	31.34	8.20	45.59
5	14.75	16.22	22.49	29.76	10.11	45.76
6	14.94	17.91	22.54	30.00	11.14	48.20
7	14.38	22.04	23.93	28.26	8.09	46.83
8	13.62	26.58	25.57	28.33	11.46	47.47
9	13.72	28.40	25.81	27.78	14.57	47.94
10	13.74	30.65	27.39	27.69	19.86	50.53
Average	**14.48**	**19.51**	**23.34**	**29.83**	**10.15**	**45.95**

► **Turning Movement Energy Consumption**. We analyzed the energy costs of turning movements, calculated by Equation 4.9. For each angle of turning, distinct energy consumption is associated. Figure 4.17a shows the average energy expended by all drones for turning movements in the Ipanema scenario while varying the number of drones used in the mission. We observed that BA* and AntIoD algorithms achieved similar results, likely because the grid-like structure of the scenario facilitates zigzag movements in the BA* heuristic, thereby avoiding many turns. Despite this advantage, the AntIoD algorithm also achieved good results in this metric. However, the CPP-IoD and ACO algorithms spent significantly more energy in turning movements, as they prioritize finding paths based on distance rather than energy cost.

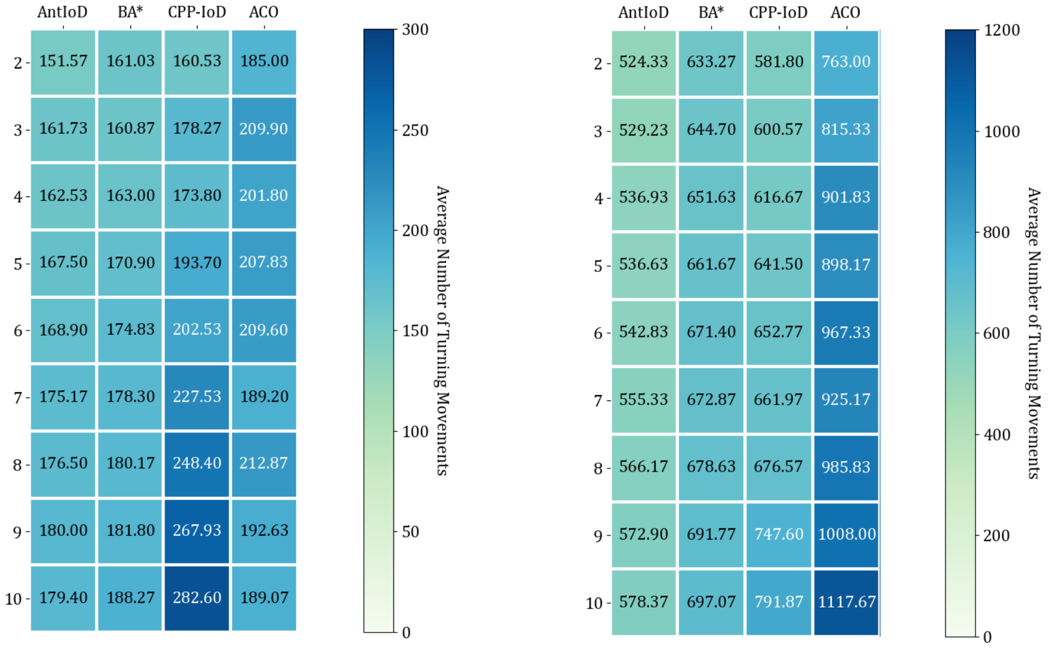
Figure 4.17c shows the average number of turns performed by all drones in a mission in Ipanema. From the analysis, it was observed that the optimization of turn costs is not highly significant. This can be attributed to the fact that the drones must perform numerous turning movements in the Ipanema scenario to accomplish the coverage task. Additionally, since the edges in the scenario are directed and limited by airways, they limit the drone's movement options, thereby leading to an increase in the number of turns executed.

The analysis of the energy costs related to turning movements in the Riverside scenario revealed a higher expenditure than in the Ipanema scenario, as depicted in Figure 4.17b. The ACO algorithm performed poorly in all cases, selecting edges to be covered randomly without considering energy consumption. The AntIoD algorithm demonstrated its ability to minimize energy expenditure in turning movements, particularly in scenarios unfavorable to the BA* heuristic, such as the Riverside scenario. The CPP-IoD algo-

Figure 4.17: Turning movement results for both Ipanema and Riverside scenarios



(a) Average energy consumption in turning movements (kJ) × number of drones in Ipanema
(b) Average energy consumption in turning movements (kJ) × number of drones in Riverside



	AntIoD	BA*	CPP-IoD	ACO
2	151.57	161.03	160.53	185.00
3	161.73	160.87	178.27	209.90
4	162.53	163.00	173.80	201.80
5	167.50	170.90	193.70	207.83
6	168.90	174.83	202.53	209.60
7	175.17	178.30	227.53	189.20
8	176.50	180.17	248.40	212.87
9	180.00	181.80	267.93	192.63
10	179.40	188.27	282.60	189.07

	AntIoD	BA*	CPP-IoD	ACO
2	524.33	633.27	581.80	763.00
3	529.23	644.70	600.57	815.33
4	536.93	651.63	616.67	901.83
5	536.63	661.67	641.50	898.17
6	542.83	671.40	652.77	967.33
7	555.33	672.87	661.97	925.17
8	566.17	678.63	676.57	985.83
9	572.90	691.77	747.60	1008.00
10	578.37	697.07	791.87	1117.67

(c) Average number of turning movements × number of drones in Ipanema
(d) Average number of turning movements × number of drones in Riverside

Source: Elaborated by the author

rithm showed a decline in performance as the number of drones increased, surpassing the performance of the BA* algorithm.

The Riverside scenario features modeled corners that allow for a representation of a two-way airway, resulting in a greater number of turns and more significant energy expenditure when compared to the Ipanema scenario. Figure 4.17d illustrates the average number of turns in each simulated scenario for Riverside. Notably, the difference in the number of turns between AntIoD and the other algorithms is more prominent in this scenario. However, it is essential to note that the number of turns alone does not accurately represent the energy expenditure, as each angle has a different associated energy cost.

Table 4.14 summarizes the improvement results of the ACO algorithm compared to other algorithms, considering the number of turns. A significant improvement of 40.54% was observed for the Riverside scenario compared to the ACO algorithm. On the other hand, a smaller improvement of 2.27% was observed for the BA* algorithm in the Ipanema scenario. However, all the algorithms have resulted in an overall improvement in the energy consumption for the given mission.

Table 4.14: Reduction of energy consumption in turning movements

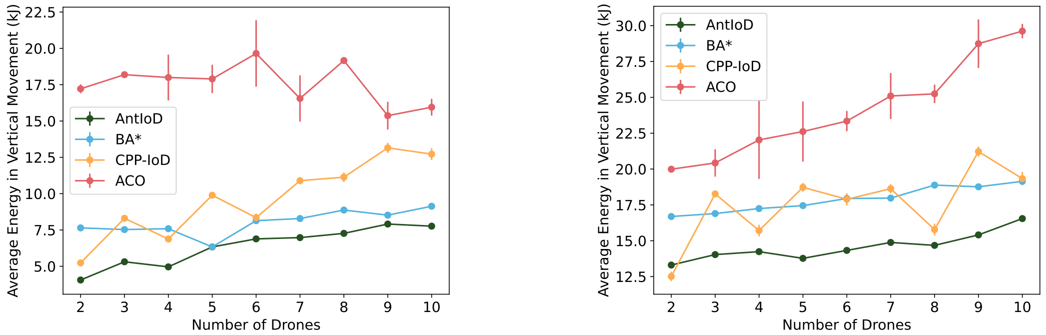
# Drones	**Ipanema (%)**			**Riverside (%)**		
	BA*	CPP-IoD	ACO	BA*	CPP-IoD	ACO
2	5.88	5.59	18.07	17.20	9.88	31.28
3	-	9.27	22.95	17.91	11.88	35.09
4	0.29	6.48	19.46	17.60	12.93	40.46
5	1.99	13.53	19.41	18.90	16.35	40.25
6	3.39	16.61	19.42	19.15	16.84	43.88
7	1.76	23.01	7.42	17.47	16.11	39.97
8	2.04	28.95	17.08	16.57	16.32	42.57
9	0.99	32.82	6.56	17.18	23.37	43.16
10	4.71	36.52	5.11	17.03	26.96	48.25
Average	**2.27**	**19.19**	**15.05**	**17.66**	**16.73**	**40.54**

► **Vertical Movement Energy Consumption**. We examine the energy costs of vertical movement, representing the upward and downward flights. Changing altitudes is a significant expense in drone missions. The vertical direction is calculated using Equations 4.7 and 4.8. Figure 4.18a shows the average energy expended for vertical movements by all drones in a mission, with varying numbers of drones used in the Ipanema scenario. We observed that CPP has a higher average energy consumption for vertical movements, which increases more rapidly as the number of drones increases compared to the other algorithms.

ACO has the worst results and a considerable confidence interval. This is because when a drone is in a node without uncovered edges, ACO randomly selects a new edge to be covered, and the results may vary considerably. Consequently, ACO requires much more vertical movement because it takes time to locate uncovered edges, causing the drone to fly more than necessary. On the other hand, AntIoD has the best performance in vertical movements as it searches for solutions in which drones do not need to change altitude unnecessarily.

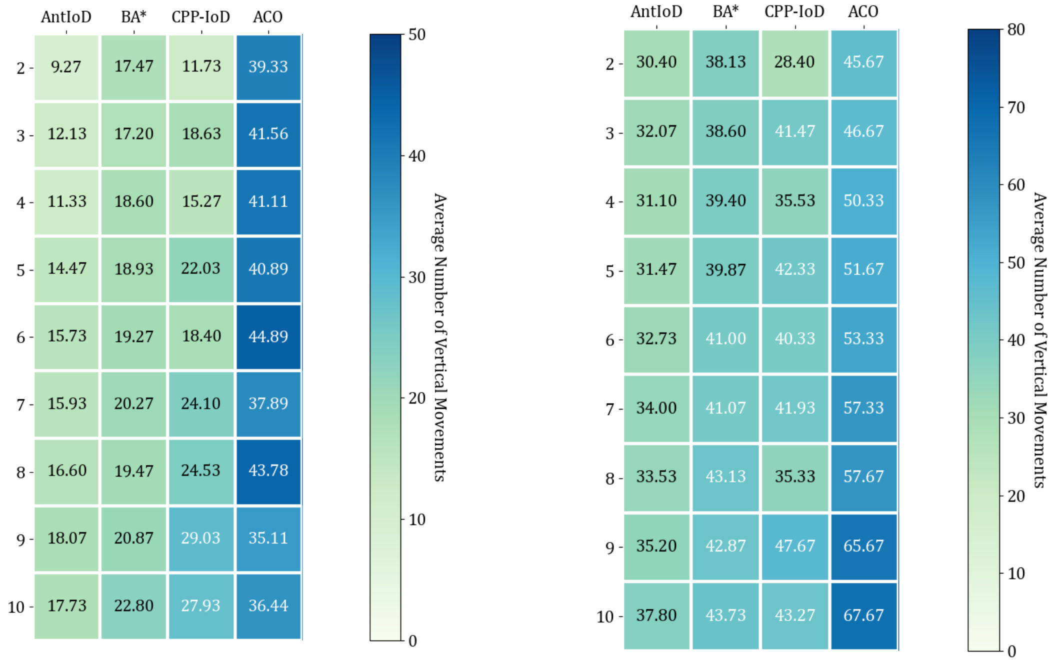
A similar result runs into the Rivesrside scenario. Figure 4.18b shows the average energy spent for vertical movements performed by all drones in a mission, varying the number of drones used. Once again, ACO had the worst performance, while AntIoD obtained the lowest energy average for vertical movement compared to the other algorithms. Comparing the results with those obtained in the Ipanema scenario, it is observed that the CPP-IoD algorithm had a more unstable effect in this scenario. However, the number of

Figure 4.18: Vertical movement results for both Ipanema and Riverside scenarios



(a) Average energy consumption in vertical movements (kJ) × number of drones in Ipanema

(b) Average energy consumption in vertical movements (kJ) × number of drones in Riverside



(c) Average number of vertical movements × number of drones in Ipanema

(d) Average number of vertical movements × number of drones in Riverside

Source: Elaborated by the author

climbs and descents may not directly represent energy expenditure, as the energy expenditure depends on the distance traveled by the drone in the vertical direction. Therefore, short vertical stretches may be more advantageous than following a long path at the same height. Nevertheless, the energy expenditure results reflect the number of ascents and descents (Figures 4.18c and 4.18d). Table 4.15 illustrates the improvement of AntIoD compared to other algorithms regarding the number of vertical movements. The best improvement was observed in the Ipanema scenario, with 63.23% compared to the ACO algorithm. AntIoD did not outperform the CPP algorithm for 2 drones in the Riverside scenario. The algorithm has generally improved by at least 15.04% in the number of vertical movements. Overall, the AntIoD algorithm's performance was noteworthy in reducing the number of vertical movements.

Table 4.15: Reduction of energy consumption in vertical movements

# Drones	**Ipanema (%)**			**Riverside (%)**		
	BA*	CPP-IoD	ACO	BA*	CPP-IoD	ACO
2	46.95	21.02	76.44	20.28	-	33.43
3	29.46	34.88	70.80	16.93	22.67	31.29
4	39.07	25.76	72.43	21.07	12.48	38.22
5	23.59	34.34	64.62	21.07	25.67	39.10
6	18.34	14.49	64.95	20.16	18.84	38.62
7	21.38	33.89	57.95	17.21	18.92	40.70
8	14.73	32.34	62.08	22.26	05.09	41.85
9	13.42	37.77	48.54	17.88	26.15	46.40
10	22.22	36.52	51.34	13.57	12.63	44.14
Average	**25.46**	**30.11**	**63.23**	**18.93**	**15.04**	**39.30**

4.4 Final Remarks

In this section, we present a way of organizing the airspace using airways. We propose that the airways be overloaded on the land routes. In addition, we classify between public and private airways. The public ones are for all drones, and the private ones are for the exclusive use of the government or companies. Furthermore, we consider that the airways can be overlapped and that drones can change altitudes between them. This organization is considered in all protocols and methods developed in this thesis project.

We presented a comprehensive guideline for designing drone delivery decision systems. Specifically, our contribution lies in providing a systematic approach that facilitates the development of drone delivery decision systems. The proposed method considers why, where, and when to use drone delivery. Our approach covers the definition of the Internet of Drones and is validated using real-world data. Additionally, we design an algorithm for distributing points that can serve as depots or droneports. We applied our method to a case study, and our results show that employing our guideline in decision systems decreases the delivery time by 49% compared to real delivery time.

We also devised a method for building coverage path plans for multiple drones in the context of IoD. The process has two steps: path planning improvement and balancing the paths between the drones. Our results showed a decrease in the distance traveled by the drones to cover the evaluated area. Specifically, there was an improvement of approximately 9.59% in the cost of results when using the loop deletion heuristic and

12.46% when using the loop deletion heuristic and mutation algorithm. For balancing, the fewer drones used, the better the results obtained.

Chapter 5

Routing Protocols for Internet of Drones

In this Chapter, we discuss two protocols for airway-based IoD. In Section 5.1, we propose a Geocast Routing Protocol for the Internet of Drones (IoDGR). Subsection 5.1.1 presents the preliminary concepts for IoDGR. Subsection 5.1.2 discusses essential aspects to consider when developing geocast protocols for IoD. Subsection 5.1.3 presents the related work. Subsection 5.1.4 details the system model of IoDGR. Subsection 5.1.5 describes the IoDGR protocol. Subsection 5.1.6 details the simulation modeling, and Subsection 5.1.7 discusses the results. Finally, Subsection 5.1.8 presents the final discussion about the IoDGR.

In the sequence, we introduce the Position-based Routing Protocol for Software-Defined IoD (PSDIoD) (Section 5.2). This protocol seeks to consider an IoD scenario with Software Defined Network (SDN). Subsection 5.2.2 presents the related work. In Subsection 5.2.3, we introduce the proposed protocol. In Subsection 5.2.4, we specify the performance setup and we detail the evaluation results. In Subsection 5.2.5, we have a final discussion about the PSDIoD protocol. Finally, we have the final remarks in Section 5.3.

5.1 IoDGR

In IoD, there are scenarios that will need to block different airways, such as in firefighting, places with risk of explosion, safety during major events, to name a few. In this case, we will need to use a routing protocol that transmits a message warning all drones in a certain region informing them of the blocked airways. A suitable protocol for this situation is a geocast protocol, i.e., a protocol responsible for delivering messages to a set of nodes identified by their geographical location [237]. In this Section, we introduced the Geocast Routing Protocol for the Internet of Drones (IoDGR), which is the first one to consider a geocast approach for IoD.

5.1.1 Preliminaries

In an IoD scenery, ZSPs are responsible for controlling the access to the airspace and path planning [5]. The airspace is divided into zones, and each one can have one or more ZSPs. The airways can exist at the exact location at different altitudes (see Figure 5.1). The ZSP might eventually deal with some airspace constraints when creating a path planning. The restrictions have three levels: mandatory, schedulable, and unpredictable. The constraint degree is not related to the reason for blocking airways but to the place and time of airway blockage. In the following, each degree is detailed.

- **Mandatory constraints**: it is when we have deactivated airways or private airways. For example, the government can own private airways, and thus, ordinary drones cannot be used for safety reasons. ZSPs know this restriction and already consider it for path planning. Also, this type of restriction is unlikely to change over time;

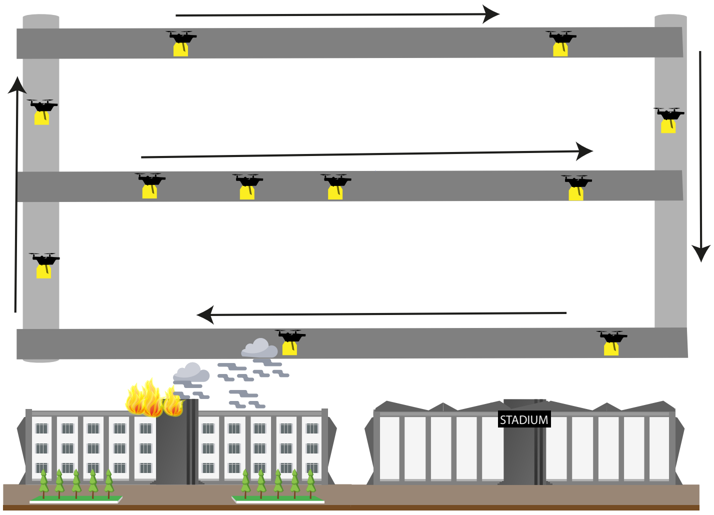
- **Schedulable constraints**: it is when we have, for some reason, airways blocked frequently. For instance, an airway over a stadium can be frequently blocked during events and matches. However, this type of blocking has a well-defined place and time for occurrence. Similarly to what happens in mandatory constraints, ZSPs also can easily prepare for schedulable restrictions;

- **Unpredictable constraints**: it is when something unusual happens (e.g., the occurrence of a fire, as depicted in Figure 5.1). This restriction can occur anywhere, at any time, and for an indeterminate time. However, in the fire example, airways around the event are expected to be blocked. ZSPs can not prepare in advance for this type of restriction. Thus, drones around that region can have their routes changed for safety.

A geocast routing protocol transmits the message to a group of nodes considering their geographical location [237]. The IoDGR protocol sends a warning message in cases of unpredictable constraints. In this case, an alert message must only be delivered in a region where unexpected restrictions will be applied, so a geocast protocol is suitable.

Protocols designed for VANETs and traditional UAV networks do not meet all requirements for IoD. Table 5.1 presents some protocol requirements for VANETs, traditional UAV networks, and IoD. An important point to be discussed is network manage-

Figure 5.1: Illustration of a risk scenery with multiple and overlapping airways at different altitudes in the IoD



Source: Elaborated by the author

ment. In IoD, ZSPs are responsible for carrying out the control and coordination of the drones. Additionally, the ZSP needs to organize and manage the drone's access to the airspace. Specifically, IoD is more like a VANET, which has a Road Side Unit to control the traffic of vehicles. On the other hand, a traditional UAV network does not include drone control in a zone.

The speed of nodes, mobility, and use of defined roads or airways have crucial differences when we compare UAVs to VANETs. In VANETs, the nodes can have high speed, but in IoD and traditional UAV networks, their speed is usually higher in urban scenarios. Mobility in VANETs happens in 2D and in UAVs in 3D for both traditional UAV networks and IoD.

Note that airways make IoD closer to VANETs. However, in IoD, we have the presence of airways at different altitudes and even overlapping. Therefore, IoD has some requirements similar to VANETs and traditional UAVs but also have some particularities (e.g., drones move fast over the airspace limited by the airways [48], drones are regulated and managed by ZSP, and the structure of the airways influences the network topology).

ZSPs also influence the power supply. In the IoD architecture [5], ZSPs are expected to manage the energy of each drone in the network. For instance, ZSPs can schedule the drones' recharge stops, given their battery's short lifetime.

Table 5.1: Some protocol requirements for VANETs, traditional UAV networks and IoD

Requirement	**VANET**	**Traditional UAV**	**IoD**
Management	A Road Side Unit (RSU) coordinates vehicles in a given region	Drones in a given region do not coordinate access to air space. Collaborations between multiple drones and multiple swarms can occur. However, each drone or swarm is generally controlled individually	All drones in a given region coordinate access to the air space. ZSPs carry out this control and coordination
Speed of nodes	High	High, usually more than VANETs in urban scenarios	High, usually more than VANETs in urban scenarios
Mobility	2D mobility	3D mobility	3D mobility and use of airways at different altitudes (see Figure 5.1)
Use of defined roads or airways	Use of roads is necessary	Use of airways is not necessary	Use of airways is necessary
Power Supply	Battery; lifetime usually high	Battery; lifetime usually low	Battery; lifetime usually low. However, the ZSP is responsible for energy management.

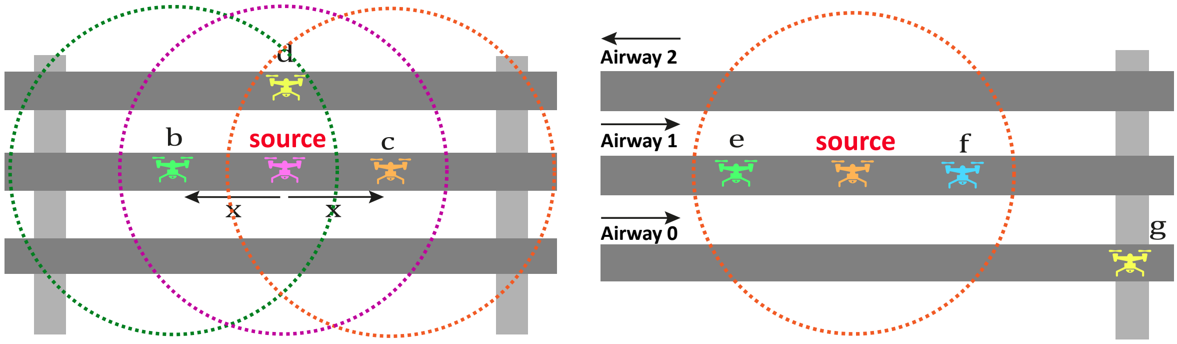
5.1.2 Geocast Routing Protocol Design for IoD

As mentioned, the IoD scenario has unique characteristics when designing geocast protocols. It is a fact that 3D mobility and high node speed impact the use of existing geocast protocols (i.e., VANETs and traditional UAV protocols) in IoD [7]. However, the use of airways at different altitudes is an issue that influences the use of geocast protocols for UAV networks developed until then, mainly in sparse scenarios. Figure 5.2 shows some examples of airway issues.

5.1.2.1 Collision Problem

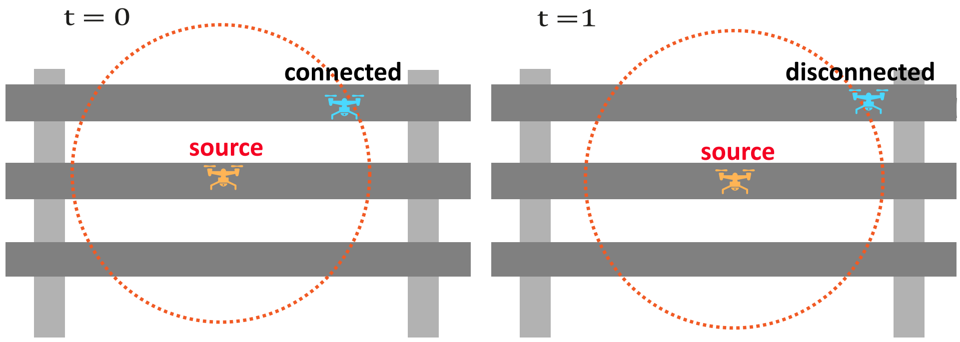
In Figure 5.2a, the *source* drone wants to transmit a message. Drones b and c are at the same distance x from the *source* drone. Considering transmission based on distance, drones b and c will schedule a message at a similar time t. If drones b and c

Figure 5.2: Examples of design issues of IoD routing protocols



(a) Collision problem

(b) Direction problem



(c) Dynamic topology change problem

Source: Elaborated by the author

start transmitting an alert message at time *t*, a collision can happen on drone *d*. That collision scenario only exists due to airways' existence at different altitudes. Using airways at different elevations can lead to other collision scenarios.

5.1.2.2 Direction Problem

Figure 5.2b illustrates the message propagation direction problem. In a VANET scenario, in some cases, routing protocols assume that a message should only be propagated to destinations heading toward the vehicle that wants to transmit the message. Thus, these protocols consider a single-direction message propagation. For instance, when an accident occurs, it is assumed that only vehicles heading toward the accident should receive an alert message, as the road will be blocked. In an IoD scenario, there is no guarantee that single-direction propagation can be effective. This is mainly a problem in sparse scenarios. As shown in Figure 5.2b, if the *source* drone transmits an alert message, drones e and f can be chosen to retransmit the message. In this case, the drone e will retransmit the message due to the single-direction approach and the airway 1 direction. Therefore, drone g may never receive the alert.

5.1.2.3 Dynamic Topology Change Problem

Another critical issue is the high mobility of nodes. As illustrated in Figure 5.2c, a possible scenario is a node at the boundary of the *source* node transmission range. In this example, the two nodes are going in the same direction and at the same speed. At first, it can be expected they remain connected at moments $t = 0$ and $t = 1$. However, a node can go up and down within the airway boundaries. Thus, if the node is at the transmission range boundary, it can disconnect and connect frequently.

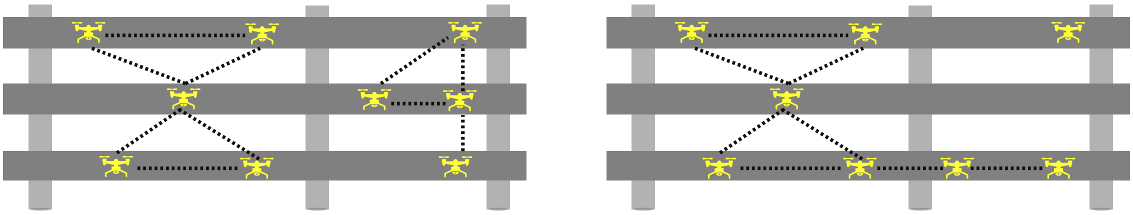
5.1.2.4 Network Fragmentation Problem

The more sparse a network is, the greater the chances of network fragmentation. This can happen in specific locations and times. As Figure 5.3 shows, we can have two main types of fragmentation. The first is when two groups of drones cannot communicate, as shown in Figure 5.3a. The second is when a single drone is isolated from the other drones in the network (Figure 5.3b). These fragmentations compromise the message diffusion to other nodes, as the network is not fully connected. In IoD, due to the speed of the nodes and high dynamic topology, a node can become isolated and then connected frequently. In a real scenario, it is reasonable to assume that there are airways with denser traffic while others have sparse traffic. Thus, it is essential to use strategies for isolated nodes to receive alert messages that compromise network security and users.

5.1.3 Related Studies

Currently, geocast routing protocols for drones are designed considering either VANET requirements or Traditional UAV Networks, including FANETs (Flying Ad-hoc

Figure 5.3: Example of network fragmentation



(a) Group fragmentation (b) Isolated drone

Source: Elaborated by the author

Networks), self-organizing wireless networks for flying nodes [238]. As mentioned, IoD has similarities with some requirements of both networks (VANET and Traditional UAV Networks) but also has particular needs. Eventually, these protocols can be adapted for IoD and further investigated.

VANET geocast routing protocols are more consolidated in the academic community than their counterpart in traditional UAV networks. A VANET presents intrinsic characteristics more suitable for geocast protocols, such as continuous traffic and the necessity to warn vehicles in a specific region due to traffic congestion and/or accidents. On the other hand, the typical limited size of current UAV networks does not seem to demand geocast protocols yet. However, IoD brings a new perspective to UAV networks more similar to VANETs. This perspective has intrinsic characteristics where geocast protocols are suitable due to the continuous UAV traffic in the urban environment. Table 5.2 compares the IoDGR with other geocast routing protocols.

As observed in Table 5.2, to the best of our knowledge, IoDGR is the first routing geocast protocol for IoD that considers network fragmentation, airways, and shortcut concepts simultaneously. The following sections will discuss geocast routing protocols for VANETs and traditional UAV networks and the use of these protocols for IoD.

5.1.3.1 Geocast Routing Protocols for VANETs

Geocast routing protocols for VANETs aim to transmit a message to vehicles for safety. In this aspect, the objective is similar to the expected behavior of geocast IoD protocols. However, VANET protocols were designed for a 2D environment. A geocast routing protocol for a VANET works as follows: when a risk scenery is detected, the source vehicle broadcasts an alert message to all vehicles inside a predetermined area. Next, metrics or strategies choose the vehicles that will disseminate the message. One of the most commonly used metrics is the farthest nodes.

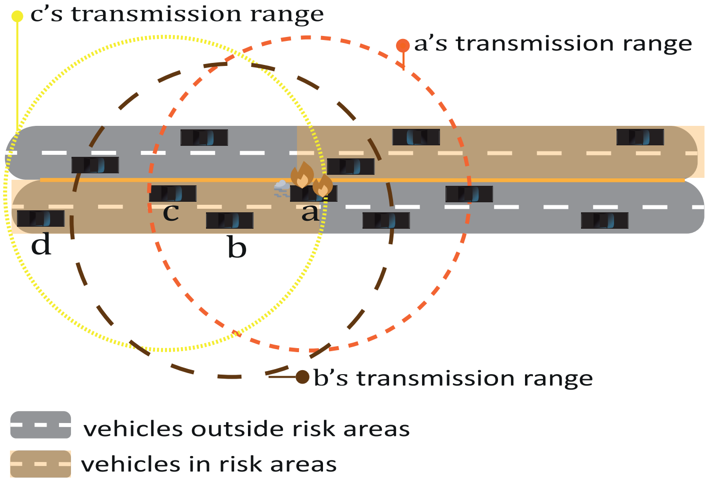
Table 5.2: Related studies with geocast routing protocols for VANETs, FANETs, traditional UAV and IoD

Protocol	Network	Scenario	Simulation	NF	A	S
IVG [239]	VANET	Highway	GloMoSim	✓		
DRG [240]	VANET	Urban	SWANS	✓		
ROVER [241]	VANET	Urban	SWANS			
DG-Castor [242]	VANET	Highway	GloMoSim			
DTSG [243]	VANET	Highway	.NET			
Geo-SDVN [244]	VANET	Urban	OMNeT++			
Mobicast [245]	VANET	Urban	NCTUns 4.0	✓		
GeoTemporal-cast [246]	VANET	Urban	NS-3			
GeoAKOM [247]	VANET	Urban	ONE			
Geo-SID [248]	VANET	Urban	NS-3	✓		
RRPR [249]	FANET	Urban	Ad hoc C++			
GeoUAV [58]	UAV	Urban	NS-3			
IoDAGR	IoD	Urban	OMNeT++		✓	
IoDGR	**IoD**	**Urban**	**OMNeT++**	✓	✓	✓

NF: Network Fragmentation;
A: Airways;
S: Shortcut;

Some protocols, such as the Inter-Vehicular Geocast protocol (IVG) [239] and the Distributed Robust Geocast (DRG) [240], consider the farthest nodes to choose the vehicles that will transmit the message. The IVG selection illustrated in Figure 5.4 expresses the reason for using the strategy of the farthest node. Suppose nodes b and c receive an alert message from node a. In this case, node c is more distant and selected as a relay. This approach allows the message to reach node d eventually.

Figure 5.4: IVG relay selection



Source: Figure Based on [237, 239]

Another concept used for some protocols in VANETs is the zone of forwarding (ZOF) and the zone of relevance (ZOR). These terms are used in DRG [240] and Reliable Geographical Multicast Routing (ROVER) [241] protocols. All nodes in the ZOF area must receive the message and are possible relay nodes. All nodes in the ZOR must receive the message, but they are not potential relay nodes.

Also, as shown in Table 5.2, various VANET protocols were developed for highway scenarios (one-dimensional topology). In this case, the protocols take advantage of the direction of the road, such as IVG, Direction-based GeoCast Routing Protocol (DG-Castor) [242], and Dynamic Time-Stable Geocast Routing (DTSG) [243]. For instance, given the one-dimensional topology, the DTSG protocol uses vehicles moving in the opposite direction to transmit the alert message. On the other hand, some protocols consider the two-dimensional topology (urban scenario), such as DRG, ROVER, and Geocast Protocol for Software Defined Vehicular Networks (Geo-SDVN) [244].

Geocast routing protocols for VANETs also need to deal with network fragmentation. DRG [240], for instance, has a mechanism to overcome temporary network fragmentation. Specifically, periodic retransmissions are used until the message is received. Mobicast Routing [245] uses a zone of approaching (ZOA) to estimate a dynamically flexible ZOF to disseminate the alert message and, thus, overcome network fragmentation.

Recent studies related to geocast routing protocols have considered different strategies. For instance, Geo-SDVN [244] uses (Software Defined Vehicular Network (SDVN), GeoTemporal-cast [246] takes advantage of the temporal scope of data, Smart Geocasting Protocol for Vehicular Networks (GeoAKOM) [247] uses complementary information extracted from GPS data, and Geocast-based Safety Information Dissemination (Geo-SID) [248] disseminates safety information dissemination in urban areas without using a ZOF.

Although those protocols can be adapted for IoD, almost all have intrinsic characteristics precisely designed for VANET. The most distinctive aspect is related to the 1D/2D environment (e.g., highways). For instance, DTSG considers the structure of a traditional highway to disseminate a message to vehicles moving in the opposite direction. This scenario is not always common in IoD. Also, as mentioned in Section 5.1.2, the IoD environment can have collision scenarios that did not occur in 2D networks due to the positions of airways. Given those and other IoD requirements, such as the high speed of the nodes, airways at different altitudes (even overlapping), and frequent network fragmentation, we need to devise new IoD protocols.

5.1.3.2 Geocast Routing Protocols for UAV Networks

As mentioned, the current UAV networks are not suitable for geocast protocols due to the typical limited size of the networks. Nonetheless, the IoD is expected to be soon used similarly to VANETs. The architecture of IoD will provide a scenario where will be necessary geocast protocols.

Despite this, FANETs and traditional UAV network protocols frequently use UAVs' geolocation. Commonly, each node knows its position using GPS [106, 238, 250]. Also, the destination position can be obtained through a Reactive Location Service (RLS), Grid Location Service (GLS), or Hierarchical Location Service (HLS) [106]. The UAV positions, in certain circumstances, are broadcast to avoid and prevent node collisions and manage the network. The localization can also be utilized for geolocation-based routing protocols [251].

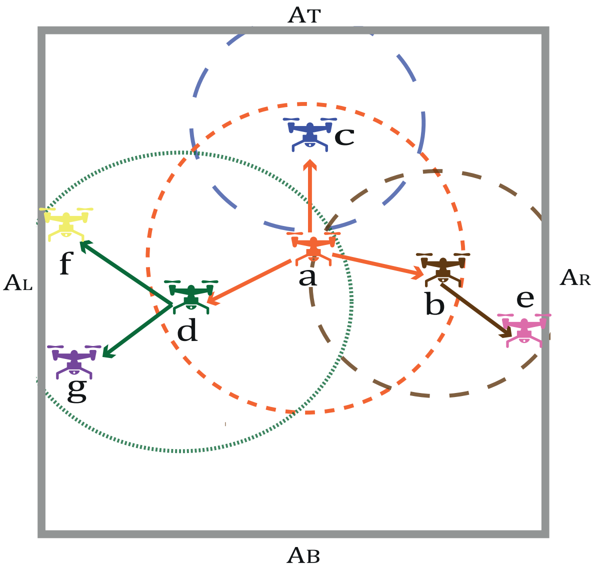
In geocast protocols, using them with characteristics similar to VANET in current drone networks is not common. The main reason is that swarm drone applications do not need this resource. Drone missions generally have relatively low numbers of drones (compared to VANET traffic) and are very close to each other. However, it is possible to find geocast protocols for UAVs in the literature.

One example is GeoUAV [58], a geocast protocol for fleet UAVs. This protocol considers the forwarding zone a square (see Figure 5.5), and each UAV determines its position. Figure 5.5 illustrates the packet dissemination, where UAV a is the source. Briefly, packet dissemination in GeoUAV works as follows: initially, UAVs b, c, and d receive the packet from the source a. These drones are closer to edges A_R, A_T, and A_L, respectively. Next, they are chosen as relays and transmit the packet. In the following step, for example, node b calculates the distances $D(b, A_R)$, $D(a, A_R)$, $D(b, A_B)$, and $D(a, A_B)$. In the example of Figure 5.5, $D(b, A_R)$ is smaller than $D(a, A_R)$, and $D(b, A_B)$ is smaller than $D(a, A_B)$. Thus, the closest edge of minimum distance is A_R since $D(b, A_R) < D(b, A_B)$.

In the sequence, UAV b waits for a time t. After that time, if the packet is still relevant (received only once), it discovers another drone near its axis and then transmits the packet to it (in this example, drone e). Then, drone e performs the same process. Although this protocol was designed for UAVs, the GeoUAV protocol is not suitable for IoD. Specifically, it does not consider a topology where it is possible to have multiple directions (due to overlapping airways).

Another case is the Robust and Reliable Predictive Routing (RRPR) [249] protocol. This is a hybrid protocol that considers unicast and geocast router strategies. RRPR performs a location prediction of an intermediate node using 3D estimation and uses directional transmission toward the predicted location. However, the two mentioned

Figure 5.5: GeoUAV packet dissemination



Source: Figure Based on [58]

protocols do not consider the characteristic of an Internet of Drones with a constant flow and well-defined airways, resembling VANETs. Hence, an IoD protocol must consider both UAV network features, such as speed, frequent disconnection of nodes, and the 3D environment, as well as issues related to IoD, such as airways and energy. To our knowledge, only our previous work IoDAGR considers some IoD characteristics, such as airspace organization using airways. Nevertheless, IoDGR advances IoDAGR considers not only airways and the speed of the nodes but new characteristics such as network fragmentation and reaching isolated drones.

5.1.4 System Model

The system model in this work comprises three parts: infrastructure, environment, and mobility definition. The infrastructure is related to the physical part, such as drones and ZSPs. The environment is about how the airways are established. And mobility is how drones move in the environment. The infrastructure includes two sets: drones (D) and ZSPs (Z).

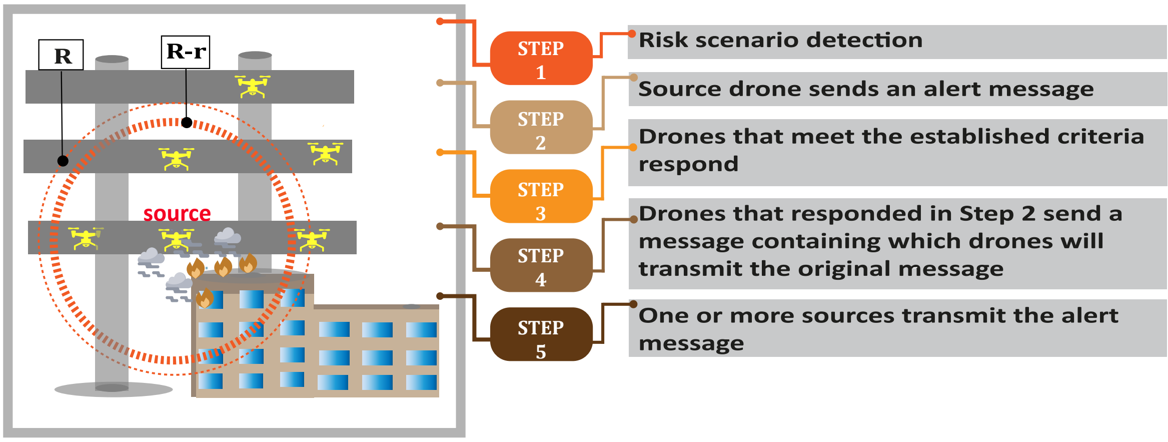
In this work, the environment comprises airways and intersections, which establish a directed graph $G = (V, E)$. A vertex v is a $3D$ point in the space represented by the triple $\langle x, y, z \rangle$. Each intersection is a vertex $v \in V$, and an airway is an edge $e \in E$. Also, an edge e is triple $\langle v_s, v_d, \mathit{altitude} \rangle$, where v_s is the source point, and v_d is the destination point.

The mobility system is about how the infrastructure elements move or position in the environment. Each ZSP $z \in Z$ has a stationary position represented by the triple $\langle x, y, 0 \rangle$. Each drone has a predefined path $p \in P$. The path is a set of vertices $v \in V$. The first v in the path is the source point, and the last v is the destination point. To reach the destination point, the drone has to pass through every intermediary vertex v in its path planning using the edges $e \in E$.

5.1.5 IoDGR Protocol

This section describes IoDGR – a geocast routing protocol for the Internet of Drones that considers the drone's future position when choosing a candidate drone to forward the message. Also, we detail the IoDGR-S and IoDGR-SC (both apply the shortcut concept). In the IoDGR-S, ZSPs are not connected; in the IoDGR-SC, ZSPs are connected. Figure 5.6 illustrates the IoDGR overview. IoDGR considers the use of airways, the dynamic topology, and the high speed of nodes. Besides those aspects, the protocol also includes the future position of drones, and the ZSPs, which allow a higher coverage to reach isolated drones. In the proposed protocols, we consider the following definitions and assumptions: each drone has an identification number and is equipped with GPS; the ZOF and the ZOR cover 3D areas.

Figure 5.6: Overview of the IoDGR protocol



Risk scenario detection

Source drone sends an alert message

Drones that meet the established criteria respond

Drones that responded in Step 2 send a message containing which drones will transmit the original message

One or more sources transmit the alert message

Source: Elaborated by the author

The first step of IoDGR occurs when a drone detects a situation that needs an alert message transmission (Step 1 in Figure 5.6). After ZOF and ZOR are specified, the coverage of the zones considers the source drone position to be determined. The size of each zone is defined in advance and is a default value for each emergency. Both zones consider a 3D area.

After a risk scenario detection, the source drone sends an alert message (Step 2 in Figure 5.6). All drones inside the source transmission range (R) can receive the message. However, due to the dynamic topology change in IoD, just the drones in the $R-r$ distance will respond to the source drone message. Variable r is a discount value to decrease the chances of disconnection. Specifically, this variable is determined based on the drone's maximum speed.

The criteria for drones to respond to an alert message are: (i) the drone is in the ZOF area; (ii) the drone is in the source drone transmission range area $-$ r; (iii) it is the first time that the drone is receiving the message after a time t. These criteria are considered in Step 3 of Figure 5.6. Next, the source drone sends a message containing which drones will transmit the original message (Figure 5.6, Step 4).

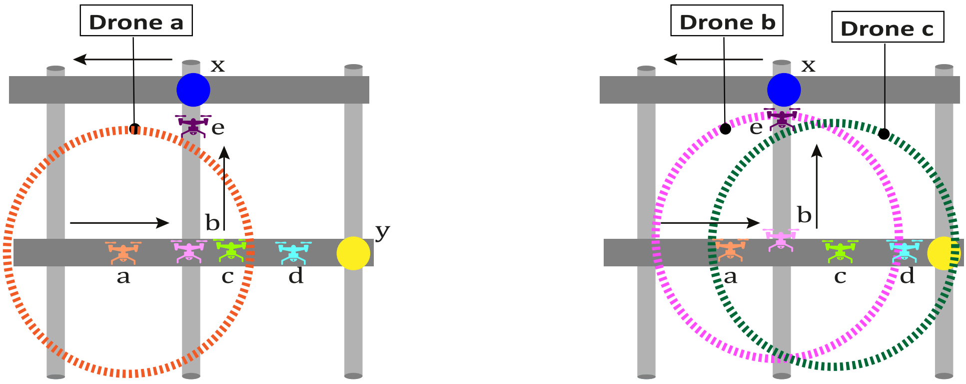
Algorithm 7: Nodes choice algorithm

```
Data: nodes_answer: List of nodes that answered to source drone
Result: nodes_retransmit: List of nodes that will retransmit the message
for $a \in altitude$ do
    for $n \in nodes\_answer$ do
        if $n.type = drone$ then
            if $n.altitude = a$ then
                $dronesByAltitude[a].append(n)$;
        else
            zsps.append($n$);
if $zsps.size() > 0$ then
    for $z \in zsps$ do
        nodes_retransmit.append($z$);
else
    for $drones \in dronesByAltitude$ do
        if ( $drones.size() = 1$) then
            $drones\_retransmit.append(drones[0])$;
        else
            for $d \in drones$ do
                $idAirway \leftarrow d.airway$;
                $droneByAirway[idAirway] \leftarrow d$;
        for $drones \in droneByAirway$ do
            nodes_retransmit.append(getDistance($drones$));
return $nodes\_retransmit$
```

IoDGR makes a choice based on the drone's position, considering the airway, the direction, and the distance of the source drone. Also, the algorithm needs to consider the future drone's position. When the shortcut concept is used (IoDGR-S and IoDGR-SC), the method analyzes whether a ZSP is among the choices. Algorithm 7 shows how the new selection is made. The first step of the algorithm separates the drones and ZSPs (Line 3). Also, the drones are grouped by altitude (Line 5).

If the shortcut concept is not being applied (IoDGR), only the part corresponding to the else statement of Line 12 is used. The whole code is necessary if the shortcut approach is used (IoDGR-S, IoDGR-SC). If a ZSP responds to the source drone, it is chosen to retransmit the message (Line 8). Only the ZSP retransmits because its range is greater than the communication radius of any other drone that has responded. If any ZSP answers, drones of each altitude are grouped by airways (Lines 16–18). As mentioned in Section 5.1.4, an airway is an edge in a graph. Finally, for each airway, the drone farthest from the source drone is chosen (Lines 19 and 20). Figure 5.7 illustrates the choice.

Figure 5.7: IoDGR example



(a) $t = 0$ (b) $t = 1$

Source: Elaborated by the author

As IoDGR focuses on sparse networks, we must choose more drones to carry out the relay and increase the coverage rate, corresponding to the percentage of ZOR drones receiving the alert message. Thus, in the new method, we look at the next point in the path planning of each drone. Figure 5.7a shows an example where drone c has vertex y as its next point, and drone b has vertex x as its next point. Figure 5.7b illustrates that the two drones are chosen because they have different directions. For each way, they are the farthest drones from the source drone. In this case, the message reaches the upper airway drones.

5.1.6 Simulation Modeling

To simulate the proposed protocols, we used the IoDSIM[1] simulation tool. IoDSIM is a simulator for IoD that can define airways in distinct altitudes. Table 5.3 shows the value of the configuration parameters used in the simulator. The three scenarios are distinguished only by the number and position of the ZSPs, which are only available for shortcuts in the IoDGR-S and IoDGR-SC scenarios. We performed 30 simulations in an area of 1900 m × 500 m × two altitudes of airways. Also, the results correspond with an average value of 30 runs with a 95% confidence interval. The first altitude was 30 m, and the second was 60 m. The time of the simulation was 300 s.

Table 5.3: Simulation configurations

Parameter	**IoDAGR**	**IoDGR**	**IoDGR-S**	**IoDGR-SC**
Mobility model	Gauss-Markov			
Number of drones	{100, 200, 300, 400} & {100, 120, 140, 160, 180, 200}			
Simulated area	1900 m×500 m×2A			
Velocity	10–15 m/s			
DTR	120 m			
Simulation time	300 s			
Simulation runs	30			
ZOF	1900 m×500 m×2A			
ZOR	1900 m×500 m×2A			
Number of ZSPs	1	1	2	2
Number of ZSPs for shortcuts	0	0	2	2
ZTR	N/A	N/A	960 m	960 m
ZSP connected?	N/A	N/A	No	Yes

2A: two altitudes of airways;
DTR: drone transmission range;
ZTR: ZSP transmission rannge;

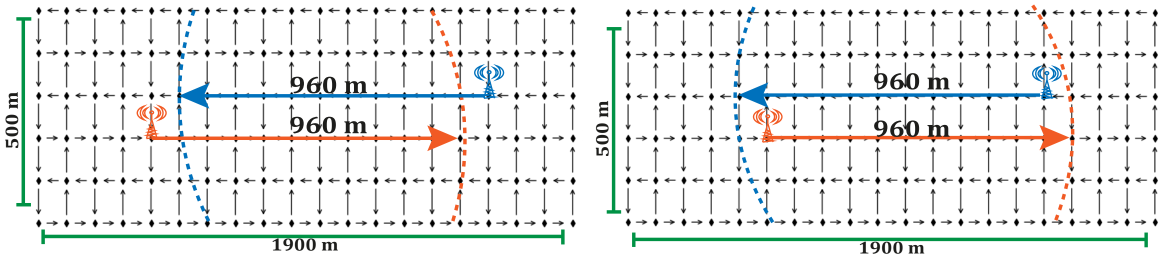
The Gauss-Markov mobility model was used to simulate the drone flight. The Gauss-Markov mobile model has been employed to generate dynamic trajectories [99]. It has been utilized in various studies in the literature [48, 58, 252] to simulate drone flight due to its ability to represent a non-linear trajectory. In the simulations, the drone speed varied between 10 m/s and 15 m/s. In addition, the number of drones was 100, 200, 300, and 400 for the IoDGR variants, and 100, 120, 140, 160, 180, and 200 when evaluating the proposed protocols and the related work. Except for IoDAGR, the number of ZSPs is two. In addition, in IoDGR-SC, the ZSPs can communicate with each other, whereas in IoDGR-S, they are isolated. Figure 5.8 shows both scenarios used in a two-

---

[1] https://iodsim.manna.team/

dimensional way. Each arrow represents the airway direction, and each dot represents an airway crossing, forming a graph. The ZSPs were positioned on the ground. Although their position is different, their transmission range is the same.

Figure 5.8: Positions of ZSPs in each scenario with shortcut topology



(a) IoDGR-S (b) IoDGR-SC

Source: Elaborated by the author

5.1.7 Performance Evaluation

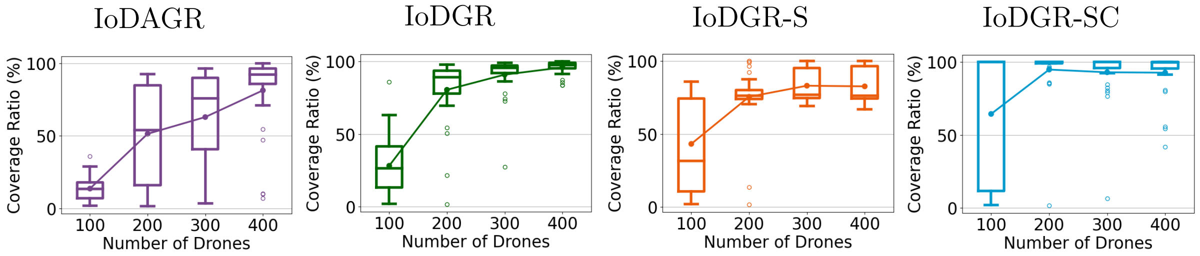
This section presents and discusses the results. Firstly, we detail the results considering our seminal work (IoDAGR) and IoDGR and its variants. Afterward, we compare two variants of IoDGR with the GeoUAV protocol since it is a geocast protocol for UAVs. We use the following four metrics to evaluate the protocols:

- **Coverage Ratio**: all drones in the ZOR need to receive the geocast message. The coverage rate represents the percentage of drones in the zone that receive the alert message;
- **End-to-end Delay**: represents the average time the message takes from the source drone to reach each of the drones;
- **Average number of hops**: represents the average number of hops used to deliver the message from the source drone to reach each of the drones;
- **Average radio power consumption per delivery**: the average consumption spent by the radio to transmit and receive messages per delivery.

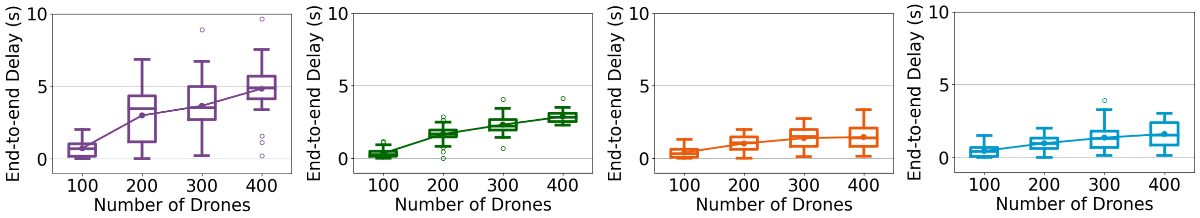
5.1.7.1 IoDGR

Figure 5.9 compares our seminal work (IoDAGR) with IoDGR and its variants. The first one uses the future position of drones (IoDGR), the second one uses the shortcut concept for ZSPs not connected (IoDGR-S), and the third one applies a shortcut approach in ZSPs connected scenario (IoDGR-SC). We detail the results for each evaluated metric in the following.

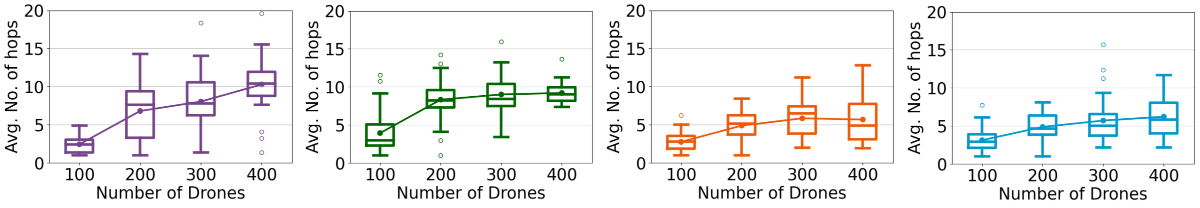
Figure 5.9: Comparison of results between IoDAGR, IoDGR, IoDGR-S and IoDGR-SC



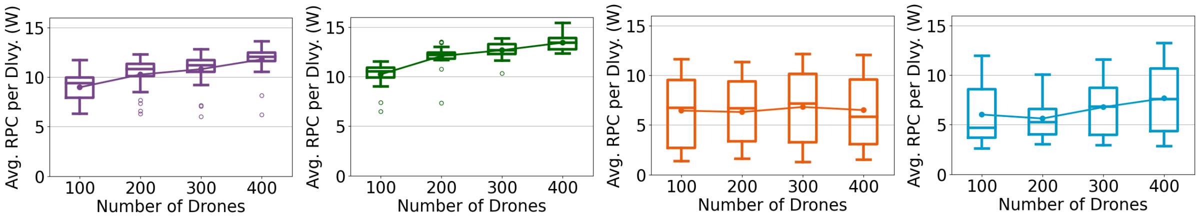
(a) Coverage Ratio (%)



(b) End-to-end delay (s)



(c) Average Number of Hops



(d) Average Radio Power Consumption (RPC) per Delivery (W)

Source: Elaborated by the author

- **Coverage Analysis**. Figure 5.9a presents the coverage ratio results for each case. IoDGR shows a significant difference in performance compared with IoDGR. In our protocol, when using 100 drones, the network is sparse and has a low coverage ratio. For 200 drones, the network can already be considered dense, so the coverage rate is higher. Choosing a suitable drone to retransmit the message is essential for broader coverage. Moreover, a possibility to increase the delivery is to consider the drone's future positions. This approach allows a better choice because it avoids two drones going the same path. Thus, IoDGR presented better results for the coverage rate.

For 100 drones, the coverage ratio increases in both scenarios where we considered the shortcut topology compared to IoDAGR and IoDGR. However, the box plot reveals a considerable data variation. This is because, with 100 drones, the network is sparse, and the message only reaches the ZSP in some scenarios. Thus, in cases where the message arrives at a ZSP, it is transmitted to many drones. In cases where it does not reach, the coverage rate is low.

For 200 drones, IoDGR-S presents better results than IoDAGR. However, the variability of the outcomes increases compared to IoDGR. This occurs because after a ZSP transmits a message, it is no longer forwarded. When a drone receives a message from a ZSP, the drone's reply message does not arrive at the ZSP, and the retransmission stops. In this scenario, the ZSPs do not reach each other, but only in cases where both ZSPs are connected independently will they have the chance to communicate with all drones. For 400 drones, the network is dense, and without the shortcut topology, the coverage rate is high. However, when using ZSPs to transmit the message, the average coverage ratio is lower as the ZSP ends the cycle of retransmissions.

Considering the scenario in which the ZSPs are connected (IoDGR-SC), the coverage ratio increases compared to IoDGR-S due to ZSP communication. For 100 drones, the result variation increases even more since the coverage is high when a ZSP receives the message, and it is low when ZSP is not reached. For 200 drones, we have the better result in all experiments. However, as the network becomes denser, isolated message collision cases can occur, causing the ZSP not to receive the message, preventing a high coverage rate.

► **End-to-end Delay and Hops Analysis**. Figure 5.9b shows the results of the end-to-end delay metric. Delay was calculated based on the delivered packets. Thus, in IoDAGR for 100 drones, few packets are delivered, and the delay is low, as the drones that received the message were close to the source drone. Figure 10c confirms this, presenting the average number of hops used in deliveries. The delay accompanies the increase in the coverage rate. The further the drone, the more hops are needed to reach it, and consequently, the longer it takes to increase the delay

With future drone positions, the delay presented a better result than IoDAGR. Figure 5.9c shows fewer hops used for message transmission. In scenarios with the shortcut topology, the end-to-end delay decreases in all cases compared to the IoDAGR protocol and with IoDGR. As the network gets denser, the message's delivery time tends to stabilize because when it reaches a ZSP, it is transmitted to more drones. Due to the time required for one ZSP to send the message to the other, the scenario with connected ZSPs has a slightly longer average delay than the scenario with no connection between ZSPs.

As expected, the average hops for message delivery reflect in the end-to-end delay scenarios that do not use the shortcut technique, requiring more hops to get the message across the ZOR. Despite the average being slightly higher for the scenario with connected ZSPs, there is no significant difference between the number of hops compared to IoDGR-S. Thus, more drones are reached with less delay and fewer hops.

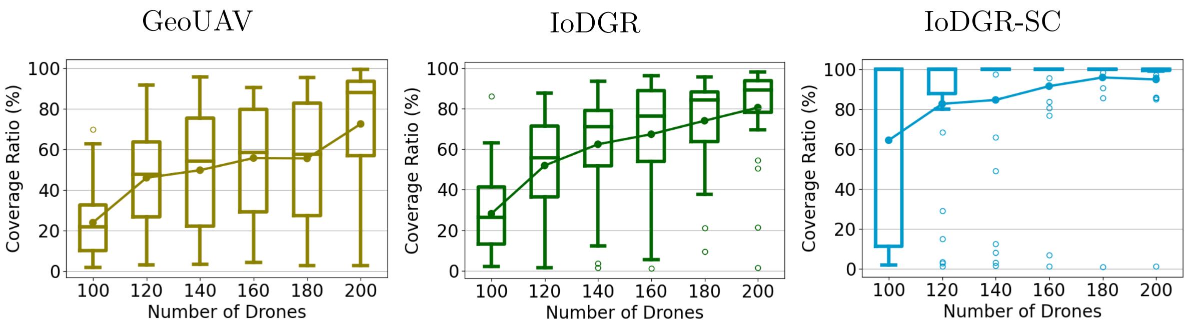
► **The Impact of Shortcuts in Energy**. A critical factor influencing the dynamics of IoD is the restricted energy of drones. Currently, drones have limited battery power. Thus, it is essential to consider this point in the design of IoD solutions. The most significant impact of using the shortcut approach is on the average consumption of the radio transmitter. When comparing IoDGR with IoDAGR, we note that the costs are similar. Thus, IoDGR has a better coverage rate and a lower delay for the same power consumption. However, we observe that the shortcut strategy reduces the average consumption. Comparing the two scenarios with the shortcut technique, IoDGR-SC has slightly higher radio power consumption for denser scenarios. Specifically, for some drones, both ZSPs reach them, so the drones in the intersecting area hear the message twice.

5.1.7.2 Comparing IoDGR with GeoUAV

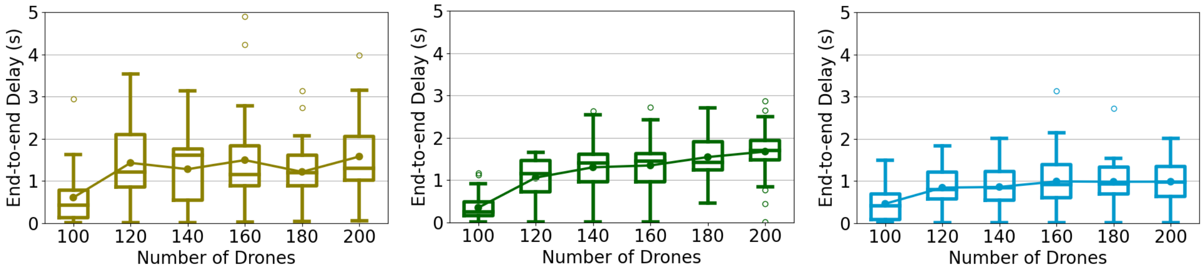
In this section, we compare IoDGR with the GeoUAV protocol. GeoUAV works in a scenario similar to IoDGR. Its objective is to propagate a message within a ZOR. In the previous section, we observed that for 200 drones, the network has characteristics of a dense network. As IoDGR performs better in sparse network situations due to the features used in its design, in this set of experiments, we used 100, 120, 140, 180, and 200 to compare IoDGR with GeoUAV.

Figure 5.10 compares the results obtained for the coverage rate, end-to-end delay, average number of hops, and average radio power consumption per delivery metrics. Figure 5.10a shows the results of the coverage rate metric. IoDGR-SC and IoDGR achieve better performance than GeoUAV for all scenarios. As the ZSPs are connected, it is possible to reach many drones. For the most sparse case (100 drones), IoDGR and IoDGR-SC increased by 4% and 40%, respectively, compared to GeoUAV. On the other hand, for the densest scenario (200 drones), IoDGR and IoDGR-SC increased by 8% and 22%, respectively, compared to GeoUAV.

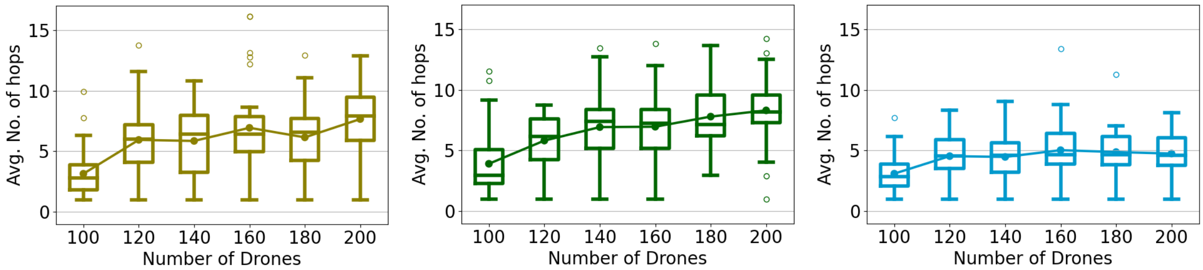
Figure 5.10: Comparison of results between GeoUAV, IoDGR and IoDGR-SC



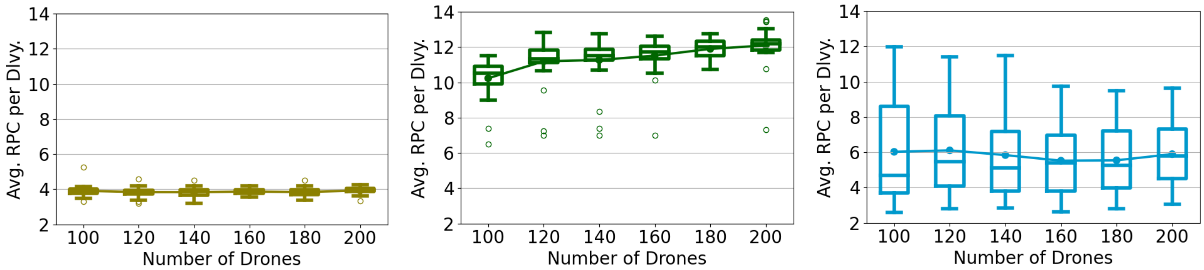
(a) Coverage Ratio (%)



(b) End-to-end delay (s)



(c) Average Number of Hops



(d) Average Radio Power Consumption (RPC) per Delivery (W)

Source: Elaborated by the author

Figure 5.10b presents the results of the end-to-end delay. In general, the average delay is similar for IoDGR and GeoUAV. Figure 5.10c shows that the number of hops matches the delay results. IoDGR-SC has a better performance for this metric. Moreover, using ZSPs decreases the time needed to reach drones further from the message origin point. In addition, the ZSP can contact many drones simultaneously, so we also see the number of hops decrease compared to the other protocols.

Figure 5.10d presents the results of the average radio power consumption per delivery. The GeoUAV protocol has a similar consumption per delivery in all analyzed scenarios. IoDGR has a higher consumption since more messages are processed (transmitted and received), making delivery possible in more sparse environments. IoDGR-SC still has a high consumption compared to GeoUAV, but the delivery is much better for sparse scenarios. Although the energy factor is a critical issue for drones, it is more important to prioritize high packet delivery over energy in emergencies.

5.1.8 Discussion

It is expected that IoD promotes constant drone traffic in airspace, enabling a dense and sparse network over time. This protocol focuses on strategies that take advantage of airways, path planning, and shortcut concepts to improve the dissemination of emergency messages when the network is sparse. The drone's energy capacity needs to be considered when planning a drone mission. Thus, in addition to the energy used to fulfill its objective, the drone must consider an extra margin of safety. Therefore, in emergencies external to the drone, it is more interesting to maintain the network security using more energy. An IoD network is highly dynamic, requiring several geocast protocol solutions. Thus, this work focuses on sparse cases with promising results to leverage state-of-the-art.

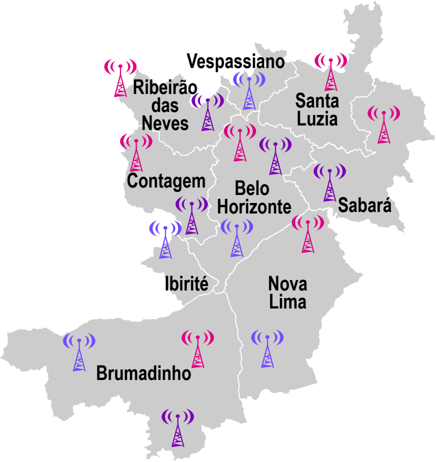
5.2 PSDIoD

Controlling a drone network is a complex process that involves both airspace control and communication-related issues. Drones of different models from several companies are expected to provide many services and share the same airspace. Thus, integrated airspace control is necessary for better traffic control, route planning, and energy savings, providing greater network security. One way to promote centralized control of the network is to use Software Defined Network (SDN). In this Section, we introduced the Position-based Routing Protocol for Software-Defined IoD (PSDIoD), which is the first one to consider an IoD scenario with SDN.

5.2.1 Software-Defined Internet of Drones

In the original IoD definition [5], each zone can have one or more ZSPs, and an organization that offers ZSPs is named Internet of Drone Service Provider (IoDSP). Figure 5.11 shows a possible form of zones organization. Specifically, in this case, each zone is a city. ZSPs is responsible for several services such as path planning, airspace control, collision avoidance, weather condition information, refuel, to name a few [5]. In zones composed of more than one ZSP, it is necessary the existence of coordination between ZSPs to control the airspace.

Figure 5.11: Belo Horizonte and near cities. Each city represents a zone severed by the ZSPs. Each color represents a different IoDSPs.

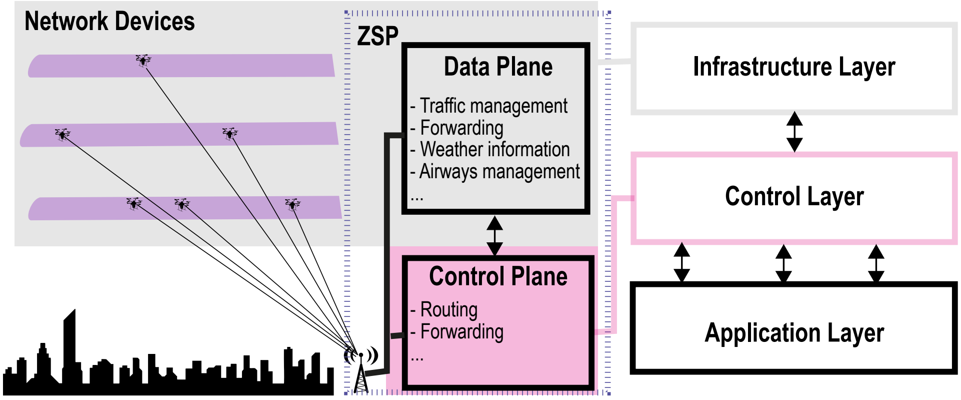


Source: Figure based on Gharibi et al. [5]

Figure 5.12 illustrates the Software-Defined Internet of Drones (SDIoD) architecture. ZSPs also need to work together as a SDIoD. Therefore, in each zone, the group of ZSPs is the control layer in the SDIoD. The infrastructure layer is composed of drones that are in the airspace of the zone. Considering that drones can fly between the zones, the infrastructure layer in SDIoD is dynamic and can change frequently.

On the Internet of Drones, each zone has different characteristics. For example, Belo Horizonte is a large state capital in Brazil and, thus, will possibly have more drones when compared with Brumadinho, a small city (in terms of population). Each zone is treated individually, and consequently, the inside management is independent of the other zones. Thus, each zone has one SDIoD control.

Figure 5.12: Software-Defined Internet of Drones Architecture



Source: Elaborated by the author

5.2.2 Related Studies

A traditional SDN increases programmability for application development and separates the control plane from the data plane [253]. A typical SDN architecture is divided into three layers: application, control, and infrastructure [254]. Specifically, this segmentation has several advantages that lead to a better user experience, such as flexibility, controllability, and ease of network management and maintenance [255].

SDNs have been used to manage VANETs [256, 257] and allow several applications like intersection collision avoidance and forward cooperative crash alert [257]. Some applications are also suitable for UAV networks. For example, improving security is a fundamental issue in both networks. Recently, many drone applications started to look for solutions that use a swarm of drones and are no longer a single drone. To leverage a fleet of UAVs, we need an architecture that provides the necessary security for drones to operate. However, just a few proposals consider using SDNs in the context of UAVs [258, 259, 260].

[258] present an SDN architecture to improve security in a swarm of drones. The SDN architecture used in this work is very similar to the typical architecture, and both approaches used the Ad-hoc On-demand Distance Vector Routing (AODV) [261] protocol for packet routing. SDN features can be helpful in specific scenarios and applications, such as disaster scenarios, surveillance monitoring, urban sensing, and search and rescue [259]. Specifically, [259] introduce the (Software-Defined UAV Network (SD-UAVNet) with the goal of live video dissemination in disaster recovery scenarios. In this case, the SD-UAVNet controller determines the locations of relay nodes for effective video dissemination. The relay nodes' choice is based on the drone location, residual energy, and drone movements to avoid a collision.

Collision avoidance is also necessary for drones collaboration networks. [260]

present an Software-Defined Drone Network (SDDN) to on-road traffic monitoring and management to improve the Internet of Vehicles (IoV). They use drones to leverage on-road monitoring systems due to their mobility and aerial view. To enable on-road drone monitoring, a set of drones must work together, and that study focus on SDDN-based collision avoidance.

Table 5.4 summarizes some of the studies discussed above that are related to this research. None of them consider an IoD scenario, in which drones have well-defined airways and their control is divided into zones. Each zone is controlled by one or more ZSPs [5]. In addition to allowing SDN traditional functionality, such as routing control, a software-defined network in this context needs to be part of the ZSPs. To the best of our knowledge, PSDIoD is the first protocol to consider an IoD scenario with SDN.

Table 5.4: Related studies with drones and Software-Defined Networks

Year	**Ref.**	**ZSP**	**Energy-awareness**	**Zone-based controller**	**Main Goal**
2019	[258]				Improve security in a swarm of drones
2019	SD-UAVNet [259]		✓		Video dissemination in disaster recovery scenario
2021	SDDN [260]		✓		Drone collision avoidance for on-road traffic monitoring
2021	**SDIoD**	✔	✔	✔	**IoD management**

5.2.3 Protocol Description

The PSDIoD is a reactive protocol that aims to transmit messages between two nodes in a drone network. The protocol takes advantage of the fact that drones frequently report their position to the ZSP to establish packet routes between two network nodes. It is quite reasonable to assume that IoD will have different types of drones, with distinct types of hardware, sharing the same airspace. We also assume that if one ZSP fails another one takes its place. In this work, we designed two versions of the protocol. The first, called PSDIoD-SP (Shortest Path), considers the shortest path in the airways to choose the routes. The second, called PSDIoD-EA (Energy-Aware), considers the estimated energy in a given path to choose the routes. The following assumptions are considered in both versions:

- Each drone has a GPS;
- Each drone informs the ZSP of its position;

- The network contains drones with five different types of transceivers. Each transceiver model has a different power consumption;
- Each zone is treated individually and has its SDIoD;
- Drones always transmit messages for drones in the same zone.

PSDIoD-SP works as follows. Whenever a drone wants to transmit a message, it sends it to the ZSP, which has access to the SDIoD of the corresponding zone. When a request arrives, a graph is created using the drone's current positions. The graph considers the range of each drone to establish an edge between two drones. As the IoD is a dynamic network in which nodes move quickly, the graph construction does not consider the drone's Communication Range (CO) but a smaller range.

The purpose of considering a range smaller than the real one is to maintain the connectivity between the drones for a long time. We call the range used the Connection Range (CN). Therefore, the connection range is calculated according to Equation 5.1, where N can be determined using the average speed of the drones and the average end-to-end delay. Thus, it is possible to discover which connection range needs to be considered and, consequently, the value of N. The shortest path is obtained using the Dijkstra algorithm, which returns just one path used to transmit the packet (Lines 1 and 3, Algorithm 8).

$$CN = CO - N. \tag{5.1}$$

Energy is a crucial factor in an IoD. Drones are expected to have a flight range better suited for urban applications such as drone delivery shortly. However, even with a capacity higher than the current average, energy will always be a relevant factor to consider. Thus, the second version of the PSDIoD protocol, namely PSDIoD-EA, considers the energy in a given path to choose the routes. Algorithm 8 presents the pseudocode of the best path return. Besides the shortest path, a breadth-first search is added to return more path possibilities (Line 5). Furthermore, the paths are analyzed, and the one with the lowest energy estimate is chosen (Line 7).

As Algorithm 8 obtains the shortest path, it is possible to have a cutoff margin so that Algorithm 9 does not generate paths with a better energy cost than the shortest path. Algorithm 9 uses a Breadth-First Search (Line 5) to traverse the graph and generate other path possibilities with the same size (in terms of the number of hops).

In addition to not analyzing paths longer than the shortest path (Line 12), Algorithm 9 does not consider paths that the energy cost estimate is greater than the shortest path energy cost (Line 18). Due to the network density, other stopping criteria can be added if Algorithm 9 takes a long time. For instance, the maximum number of times to search for a path and change the variable *continue* to be false.

Algorithm 8: Return best path from source to destination

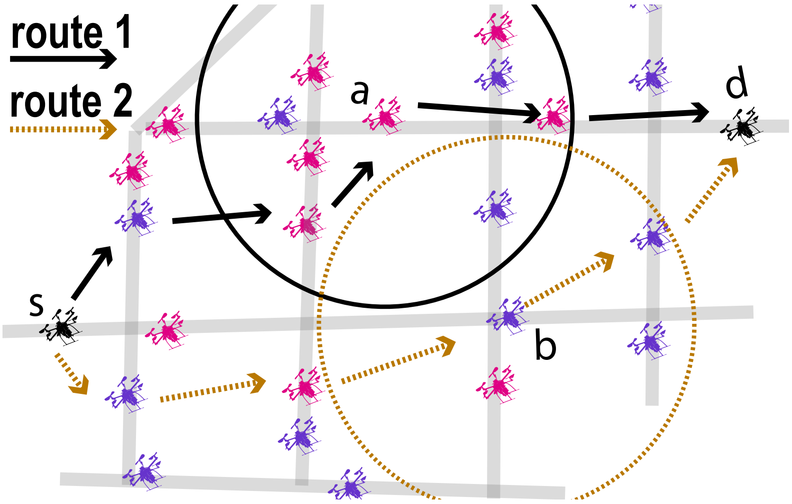
```
Data: source, destination
Result: path
$G \leftarrow$ createGraph() ;
$paths \leftarrow \emptyset$ ;
$shortestPath \leftarrow$ returnShortestPath(G, source, destination) ;
$paths \leftarrow paths \cup \{shortestPath\}$ ;
$pathsPossibilities \leftarrow$ returnPaths(G, source, destination, size($shortestPath$)) ;  // See Algorithm 9
$paths \leftarrow paths \cup \{pathsPossibilities\}$ ;
$path \leftarrow$ bestPath($paths$) ;
return $path$
```

Algorithm 9: Return some possible paths from source to destination

```
Data: matrix G, source, destination, shortestPath
Result: paths
$paths \leftarrow \emptyset$ ;
$Q \leftarrow 0$;
$p \leftarrow []$;
add(p, source);
enqueue($Q$, $p$);
$continue \leftarrow$ true;
while $Q \neq 0 \wedge continue$ do
    dequeue($Q$, $p$);
    $last \leftarrow p[\text{size}(p) - 1]$;
    if $last = destination$ then
        $paths \leftarrow paths \cup \{p\}$;
    if $size(p) < size(shortestPath)$ then
        for $i \leftarrow 0; i < size(G[last]); i{+}{+}$ do
            if $isNotVisited(G[last][i], p)$ then
                $newPath \leftarrow []$;
                add($newPath$, $p$);
                add($newPath$, $G[last][i]$);
                if $newPath.cost < shortestPath.cost$ then
                    enqueue($newPath$);
return $paths$
```

After generating the paths, Algorithm 8 finds the best path in terms of energy cost (Line 7). For this, a forecast of the transmission power cost is calculated. Figure 5.13 shows an example of how PSDIoD-EA works. In this scenario, purple drones have a more efficient power consumption model than pink drones. Routes 1 and 2 have five hops. However, route 1 has a single drone with a more power-efficient consumption, while route 2 has three drones.

Figure 5.13: Example of path selection



Source: Elaborated by the author

Also, the power spent by neighbors for listening to the transmission is considered in the forecast calculation. For example, drone ***a*** has seven drones in its range, and all of them will listen to the transmission, leading to power consumption. While drone ***b*** has four drones in its range, most of them are low-power. When choosing a route, PSDIoD-EA considers the energy expenditure of the drones and their neighbors along the path. In this example, route 1 will consume more energy because it has more neighbors and more drones with a higher energy cost. Even if routes 1 and 2 had the same types of drones (for example, all purple), route 2 would still be more efficient because it has fewer neighbors.

5.2.4 Performance Evaluation

To create the necessary IoD infrastructure for evaluating this study, we use IoD-SIM. The mobility module of this simulator allows creating airways for drones to fly. Furthermore, it is possible to specify airways at different altitudes. In the simulation, the ZSP has the role of being the SDIoD and also of controlling the traffic of drones (see Figure 5.12). Table 5.5 shows the configuration parameters used in the simulator.

Table 5.5: PSDIoD Simulation configurations

Parameter	**Configuration**
Mobility model	Gauss-Markov
Number of Drones	300
Simulated area	1000 m×1000 m×2 airways
Speed	10 m/s
Communication range	200 m
Connection range	170 m
Simulation time	300 s
MAC layer protocol	IEEE 802.11n
Simulation runs	30
Number of types of Drones	5

2A: 2 altitude for airways;

Both versions of PSDIoD were compared with Ad-hoc On-demand Distance Vector Routing (AODV) [261] since it is a well-established reactive protocol. For each protocol, 30 simulations were performed. Each drone has a unique ID in the range 0–299. In the simulation, all drones start on the ground. Due to the initialization of the drones, the first transmission occurs at $t = 12$. The sequence of transmissions follows the order of the IDs. In each, simulation a message is transmitted every two seconds by a different drone. Therefore, at $t = 12$, drone 0 transmits, at $t = 14$ drone 1, etc. Drones can suffer external actions that influence the flight, such as the wind. Thus, we use the Gauss-Markov mobility model to simulate the drone's trajectory.

Moreover, each message has the destination chosen based on the ID. Thus, drone 0 transmits to drone 1, drone 1 to drone 2, etc. If the destination is less than five hops away, it becomes the drone of $ID = ID + 1$. This process continues until a destination that is five or more hops away is found, or the highest destination is chosen. As Table 5.5 shows, the connection range is 170 m, where N was considered to be 75% of CN (calculated by Equation 5.1). Also, we have five types of drones that differ by their transceiver type. This means that we have five power consumption types. The following transceivers models were used: CC3220, CC3235S, 88W8977, ESP32-DOWDQ6, ATWINC3400A. The consumption of each one was based on the specific datasheet of each model.

Figure 5.14a shows the PDR along the time. Both PSDIoD versions have a similar delivery rate. Table 5.6 presents the final average of the PDR over the total simulation time for each protocol. It is possible to observe that the AODV protocol has a lower delivery rate than PSDIoD protocol versions. One reason is that AODV does not use the position of each drone to route message packets. Another influencing factor is the speed of nodes on the network, so the routes between drones are highly dynamic.

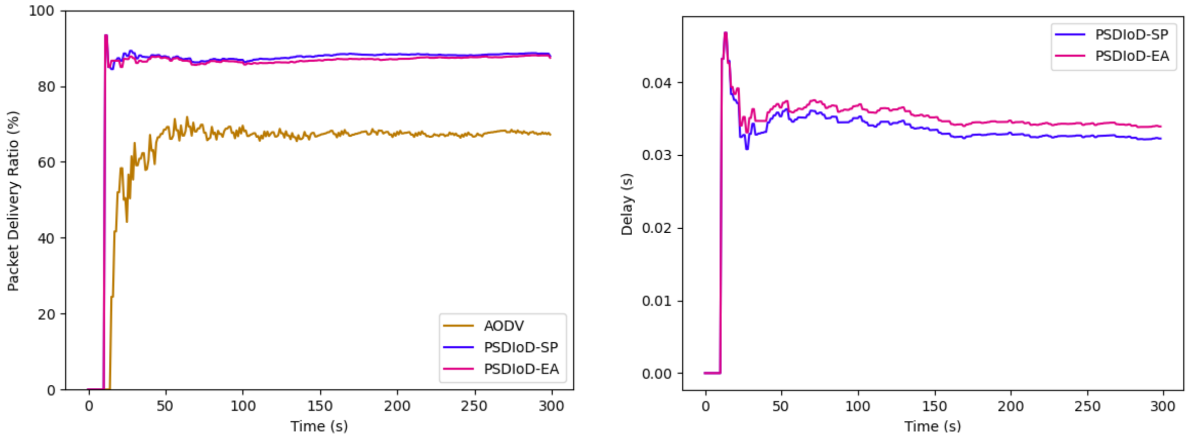
Figure 5.14b shows the end-to-end delay. As shown in Table 5.6, the AODV protocol has a longer delay than the PSDIoD protocol versions. Thus, for comparison purposes,

Table 5.6: PSDIoD Summary of results

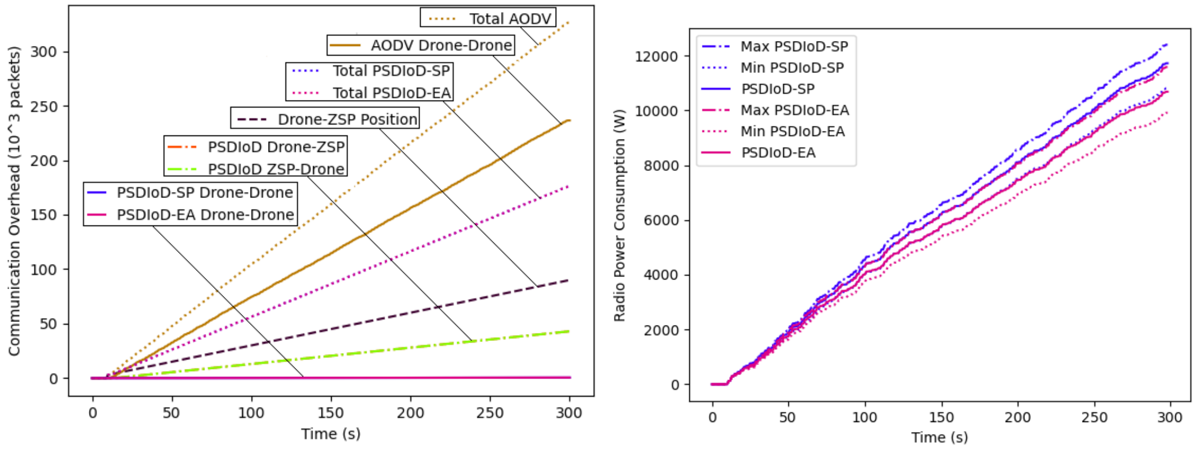
Metric	AODV	PSDIoD-SP	PSDIoD-EA
PDR	67.1954 ± 0.1056	87.9080 ± 0.1777	87.4482 ± 0.1777
Delay	5.7583 ± 0.0040	0.3390 ± 0.0003	0.322 ± 0.0002
TO	326408.1 ± 62.46	176329.6 ± 0.48	176356.3 ± 0.48
RPC	323084.7 ± 60.51	11722.4 ± 3.92	10682.4 ± 3.68

PDR: Packet delivery ratio; TO: Total Overhead; RPC: Radio Power Consumption

Figure 5.14: PSDIoD evaluation results



(a) PDR (b) End-to-end Delay



(c) Communication Overhead (d) RPC

Source: Elaborated by the author

Figure 5.14b does not show the AODV protocol. Due to the network dynamics, the AODV protocol always needs to update the routing table, delaying the message delivery. As seen in Figure 5.14b, the PSDIoD-EA protocol has a little longer delay than the PSDIoD-SP protocol. This fact is due to the generation of more path possibilities performed by Algorithm 9.

Figure 5.14c shows the communication overhead in terms of the number of generated packets. The AODV protocol performs the entire route discovery process using drone-to-drone communication. PSDIoD, on the other hand, only exchanges the necessary messages between drones. Thus, the number of messages generated between drones in PSDIoD is low compared to AODV. Despite this, PSDIoD exchanges messages with the ZSP for route discovery. Thus, the AODV protocol produces more messages than the PSDIoD protocols. PSDIoD protocol becomes even more efficient considering that drones often need to report their positions to the control base regardless of the protocol used. Therefore, PSDIoD takes advantage of location information to produce more efficient routing.

Figure 5.14d illustrates the radio power consumption (RPC) for both versions of PSDIoD. Table 2 shows the average total radio consumption. Due to AODV route discovery, radio power consumption is significantly higher than PSDIoD. Thus, with 95% confidence, it is possible to state that the protocols are different and that PSDIoD consumes less power. It is important to note that the drone mobility power consumption was not computed as the drones travel the same path. Therefore, the power consumption is the same across all protocols.

When comparing the PSDIoD-EA protocol with PSDIoD-SP, the former has a lower power consumption. According to Table 5.6, at the end of the simulations, the average of PSDIoD-EA was approximately 9% lower than that of PSDIoD-SP. Figure 5.14d also shows the expected maximum and minimum for the two versions. The maximum was calculated by simulating that all drones on the selected route had the highest radio power consumption cost. The same was done to calculate the minimum, but all drones had the lowest radio power consumption cost. However, the cost of neighbors was kept the same. For example, given Figure 5.13, to calculate the minimum of route 2, it was considered that all drones were purple, and to calculate the maximum, all drones were pink, but the neighbors were kept the same. Thus, it is possible to see that even though PSDIoD-SP selects a route only with the best drones, its cost will still be similar to PSDIoD-EA.

5.2.5 Discussion

The Internet of Drones presents a scenario where we employ SDN to enhance network programmability and expedite changes due to different factors (e.g., dense or sparse IoD). We can expect an IoD where drones will have to inform their position to a base station as proposed by the FAA. This approach can have a high energy cost. Energy expenditure is an important factor due to the power limitation of drones. In addition, a technology that drones can use to send their position to the base station also has a high cost to deploy. However, considering that the structure already exists, constantly knowing the drone's position is a piece of valuable information to build efficient message routing.

5.3 Final Remarks

In this Chapter, we presented two routing protocols for IoD. The first one is a geocast protocol, called IoDGR. The results show that the protocol has good performance in terms of coverage and end-to-end delay compared to IoDAGR and the GeoUAV protocol in sparse cases. The results were even better using the shortcut concept. Despite the need for more radio power consumption to employ the shortcut principle, it is essential to emphasize that this factor is justified due to the sparse network and the emergency scenario.

This second protocol proposed was PSDIoD, a Position-based Routing Protocol for Software-Defined Internet of Drones. Both versions (shortest path and energy-aware) take advantage that the drone often informs its position to the ZSP. PSDIoD-EA considers energy to choose the best path between two drones. PSDIoD outperforms AODV by maximizing the packet delivery ratio and reducing the delay, radio power consumption, and communication overhead. With this proposal, PSDIoD advances the state of the art of routing protocols for IoD and SDIoD.

Chapter 6

Routing Protocols for joint Other Networks and IoD

In this Chapter, we delve into the collaboration between IoD and other networks. First, in Section 6.1, we present the chapter introduction. Subsequently, in Section 6.2.5, we introduce the seminal protocol known as Store-carry-forward Protocol to Joint Bus Networks and IoD (IoDSCF), which investigates the synergy between drones and Bus Networks. In Section 6.2.6, we introduce the Delay-tolerant Internet of Drones Protocol in a Multi-vehicle Scenario (IoDMix) protocol, an enhancement of IoDSCF. With IoDMix, we begin to explore the collaboration between drones and various other terrestrial networks, including public transport networks, bicycles, and cars. For each of these protocols, we provide a scenario definition, a protocol description, and an analysis of performance results.

In the sequence, we introduce the Altitude-based Routing Protocol for Hybrid Aerial-terrestrial Networks (ARAT) (Section 6.3). This protocol seeks to integrate aerial and terrestrial networks. In Section 6.3.1, we present the ARAT preliminary concepts. In Section 6.3.2, we introduce the proposed protocol. In Section 6.3.3, we specify the performance setup. In Section 6.3.4, we detail the evaluation results. In Section 6.3.5, we have a final discussion about the ARAT protocol. Finally, we have the final remarks in Section 6.4.

6.1 Introduction

IoD is a distributed system with a huge potential to make part of the IoT environment in smart cities [124]. Specifically, applications and services can be created and/or improved using the unique characteristics of drones, such as high speed and 3D aerial vision. In addition, IoD can collaborate and interconnect other networks such as vehicular networks, cellular networks, among others. Collaboration between networks makes it possible to build cooperative protocols that improve communication, energy consumption

[112] and the networks' security.

A collaboration between IoD and other networks can happen in three ways. The first is between IoD and other networks such as Internet of Flying Cars (IoFC) and satellites. The second is between IoD and terrestrial networks like Bus Networks, VANETs, and Public Transportation Network (PTN), to name a few. The Third is the collaboration between several aerial networks and terrestrial networks, with IoD being one of these networks. The application is expected to often make use of networks. Each of these collaborations may have different requirements such as mobility, nodes, network topology, applications, hardware requirements and energy capacity. Thus, protocols must be developed for different protocols developed. In this Chapter, we explore a collaboration between IoD and Bus Networks and a collaboration between IoD, terrestrial networks, and other aerial networks.

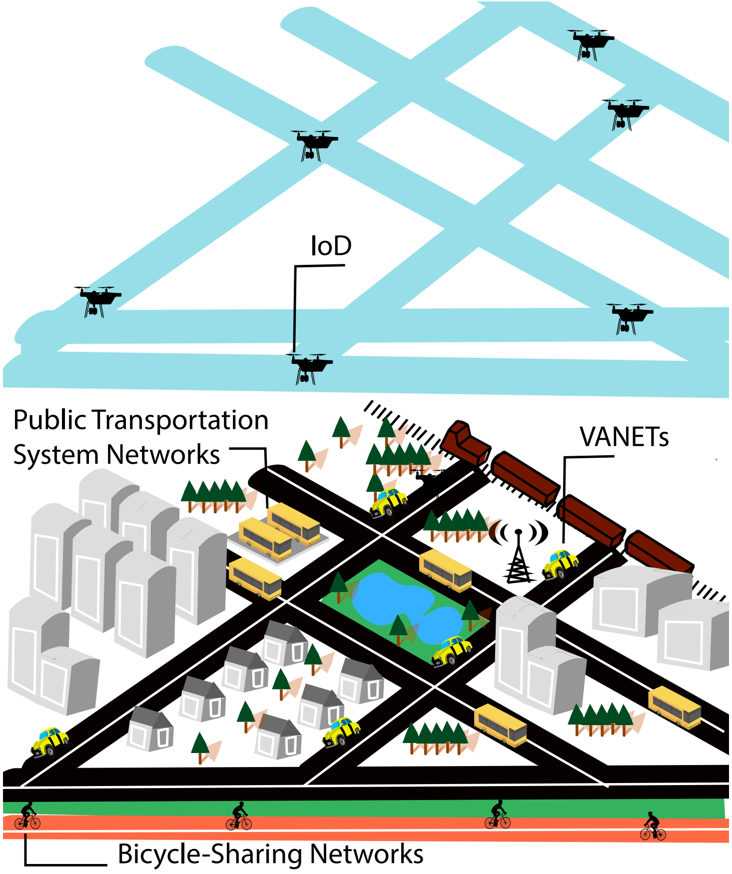
6.2 Join IoD and Terrestrial Networks

In this section, we will discuss the IoDMix protocol and its seminal IoDSCF protocol, which connects IoD and terrestrial networks. To begin, in Section 6.2.1, we will present the preliminary aspects of this work. Following that, in Section 6.2.2, we will examine related works. Moving on to Section 6.2.5, we will introduce the IoDSCF protocol, including the scenario definition, protocol description, and its evaluation and results. Finally, in Section 6.2.6, we will introduce the IoDMix protocol, along with the scenario definition, protocol description, and its evaluation and results.

6.2.1 Preliminaries

Urban centers have different networks on the ground and in the air space. For example, as illustrated in Figure 6.1, we may have VANETs, PTNs, Bicycle Sharing Networks (BSN)s, and IoD. In the case of IoD, this study assumes that the virtual airways in the airspace are parallel to landways. In addition, the airways may overlap between them. Land roads can be pathways for cars, bicycles, trams, and other means of transport. Each of these networks works independently. However, there can be a collaboration between networks. For instance, a VANET can notify a PTN of an accident or vice versa. There are two types of cooperation: separated networks that collaborate between them and networks whose only role is to assist another network. We classify these two networks as dependent networks and independent networks, respectively.

Figure 6.1: Urban center with multiple networks



Source: Elaborated by the author

- Dependent networks: these are networks that exclusively assist another network in some activity related to mobility, communication, or the provision of data for some application. For example, Samir et al. [262] present a swarm of UAVs whose objective is to provide coverage to vehicles entering a road that is not covered by other infrastructure. Consequently, this network operates only to assist the vehicular network;

- Independent networks: these are networks that work independently. However, collaboration is suitable for collecting data, moving around, or making communication more efficient. One example is a drone delivery network assisted by a truck network to carry out long deliveries [79]. The drone network works and exists independently of the truck network. The truck network also exists independently of the drone network, but collaboration among them can bring advantages for both.

Collaboration between drones and other networks can occur for several reasons, including ground transportation to increase drone travel time and save battery power

[112, 263, 264, 265, 266, 267, 268]. This type of partnership can be classified as an independent network, as networks exist independently of collaboration. Another reason for network and drone collaboration in the literature is using drones as relays to fill the gaps in sparse network communication [157, 269, 270]. In this case, we consider this partnership in the dependent network category due to using drones exclusively to support some other network (e.g., VANET and PTNs). IoDSCF and IoDMix focus on a partnership between independent networks, exploring a possible collaboration between the networks without interfering with node mobility.

6.2.2 Related Work and Motivation

The academic community has been investigating different proposals related to IoD and collaboration with IoT [113, 271, 272, 273], VANET [152, 158, 274, 275], and PTNs [112, 263, 264, 265, 266, 267], to name a few. Regarding studies related to communication protocols between IoD and another network, most studies in the literature investigate the scenario of UAV-aided VANETs. Consequently, in this context, the drone's mission is only to assist VANET networks, thus being a dependent network. For example, Samir et al. [262] use a swarm of drones to provide coverage to vehicles entering a road that is not covered by other infrastructure. An interesting factor is that the drones move according to the need of the network being assisted.

In IoDSCF and IoDMix, we focus on investigating the collaboration between IoD and other networks in the urban space without changing the mobility of the nodes. PTNs have fixed paths that can not be altered to facilitate cooperation between them and other networks. VANETs and BSNs have a path that depends on the user's destination and is also not liable to have their paths changed. For this reason, drones are suitable for filling communication gaps in terrestrial networks. However, this work considers a futuristic scenario in which the IoD will allow multiple drones performing different applications to share the airspace.

For example, drone delivery is a top-rated application and has attracted the attention of many companies [39]. Drone delivery will form a continuous flow of drones in the sky. Hence, it will not be possible to change the path of these drones to make collaboration between IoD and other networks possible. Thus, we seek to investigate a scenario in which collaboration occurs without interfering with the mobility of nodes. Another crucial point is that the density of the IoD is highly dynamic and depends on the spatiotemporal characteristics of the network. Therefore, it is reasonable to assume that the IoD will have sparse scenarios in which continuous communication between nodes will be impossible. In this case, we have a Delay-Tolerant Internet of Drones (DTIoD). Thus, IoD must be delay-tolerant to keep communication between its nodes.

Table 6.1 presents proposals related to routing protocols for drones to collaborate with other networks in a delay-tolerant scenario. Most proposals [153, 157, 269, 270] do not use IoD concepts, i.e., drones have their mobility affected to assist other networks.

Table 6.1: Related studies to routing protocols, IoD and collaboration with other networks

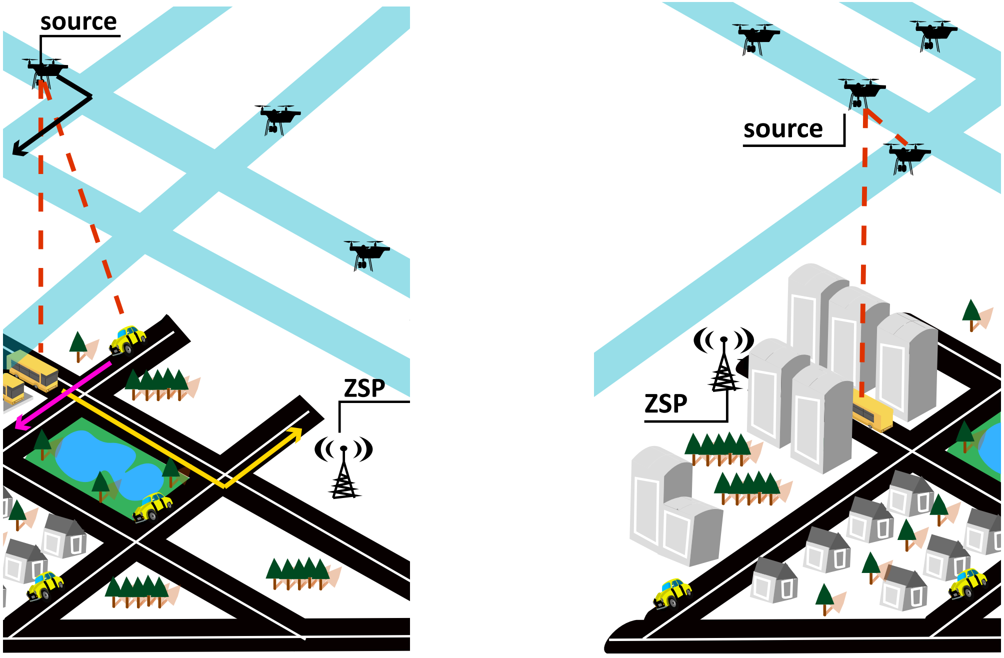
Year	**Ref.**	**IoD**	**UAV**	**VANET**	**PTN**	**Bicycle**	**Highlights**
2017	[153]		✓	✓			A reactive routing protocol for transmitting packets in the sky
2020	[157]		✓	✓			A reactive protocol for UAV-aided VANETs
2021	[269]		✓	✓			A routing protocol for UAV-Assisted Vehicular Delay-Tolerant Networks
2022	[270]		✓	✓			A bio-inspired UAV-aided vehicular delay-tolerant network routing protocol for urban VANET environments
2022	IoDSCF	✓			✓		A store-carry-forward protocol for joint bus networks and IoD
2023	**IoDMix**	✓		✓	✓	✓	**A store-carry-forward protocol for joint PTNs, VANET, BSNs, and IoD to fill the gaps in IoD communications**

6.2.3 Design Issues of Routing Protocols for Delay-tolerant IoD in a Multi-vehicle Scenario

This section introduces crucial issues that must be considered in the IoD protocol development. Specifically, we address the context of multi-vehicle and delay-tolerant scenarios. Figure 6.2 illustrates possible collaboration scenarios of IoD and other networks to be discussed below.

- **Path Planning Predictability:** A characteristic of PTNs is high network predictability. Buses have fixed routes and schedules, allowing discover the path the vehicle will follow. The same occurs for trains, as the path is easily predictable. Therefore, often choosing the node closest to the message destination may not be the best choice. Figure 6.2a illustrates an example where the source drone wants to send a message to the ZSP. The source drone has two neighbors within range, a bus

Figure 6.2: Illustration of possible collaboration scenarios of IoD and other networks



(a) Path planning predictability

(b) Urban canyon

Source: Elaborated by the author

and a car. The bus has its path known and can inform the drone if it can deliver the message and even inform its expected delivery time. As for the car, it does not have its known path. In addition, privacy issues may affect the car's ability to share information about its trajectory [37]. Therefore, as indicated in Figure 6.2a, the vehicle, despite being closer to the ZSP, may be going in a direction opposite to the desired one. Thus, the choice of the next node should consider the prediction of the node's path whenever possible.

- **Environment Interference:** Urban centers may have obstacles that hinder communication. Urban canyons can make communication difficult for land vehicles. The choice of the transmission node must take this factor into account. Specifically, nodes that have less interference must have priority. Thus, it is reasonable to assume that drones have a preference among other nodes, as airspace has fewer obstacles compared to land routes. Figure 6.2b illustrates an example where the source drone wants to send a message to the ZSP. The source drone has two neighbors within range, a bus, and another drone. Both the bus and the drone are heading toward the ZSP. Besides, the expected message delivery is similar in the two neighbors. Despite being closer, the bus has a lower average speed than the drone due to bus stops. Thus, it is more advantageous to choose the drone as the next node, as the risk of the message being lost is lower and, consequently, the delivery guarantee is higher.

- **Hardware Restrictions:** Each vehicle has different hardware restrictions due to distinct factors, including the size and value of the vehicle. For example, it is reasonable to assume that cars, buses, and trains have a higher value and more space to store extra equipment than bicycles. Thus, bicycles are expected to have more straightforward transmission equipment and a lower range capacity. Such factors must also be analyzed and considered when choosing the transmission node. Hence, there must be a priority of choice among the vehicles available for IoD collaboration.

6.2.4 Definitions and Assumptions

This section details the assumptions and definitions used to model our proposal. Also, we describe the IoD, VANET, BSNs, PTN and bus network.

6.2.4.1 IoD

IoD has three components. The airspace architecture is composed of the airways and their connections. The nodes are composed of a set of drones D and a single ZSP that coordinates the drones. And a set PD of drone path plannings for the environment used. In the sequence, we defined each component as follows:

- Airspace architecture: In this work, we defined the airways based on the roads described in the scenario modeling. As mentioned, the roads are based on a real region of Ottawa, Canada. The set of airways and their intersections form a directed graph $GD = (VD, ED)$. Each intersection is a vertex $vd \in VD$, and the airways segments are an edge $ed \in ED$;
- Nodes: one ZSP responsible for coordinating the set of drones D. And a set of drones D, where each drone d has a path planning;
- Path planning: Each drone has a predefined path $pd \in PD$. We created the path randomly using the airway graph GD for each drone. The path planning is a sequence of vertices $\langle vd_1, vd_2, \ldots, vd_i \rangle$, where i is the number of vertices in the path. The first vertex vd_1 has an edge $ed = \langle vd_1, vd_2 \rangle$ that connects it with the vertex vd_2 and so on.

6.2.4.2 PTN

In this work, the PTN has two types of vehicles: buses and trains. In the sequence, we define the nodes and the PTN path planning:

- Nodes: we created the set of buses B and the set of trains T based on the actual bus lines and trains in the region used. The city of Ottawa has a public transport called O-train. This system has underground and non-underground parts. In this work, we consider a stretch of the O-train that is not underground;
- Path planning: Each bus and train has a predefined path $pr \in PR$. The bus paths were created using the real route for each line operated in the region. Also, the train paths were created using the train railway tracks. All the paths are composed of a set of vertices $vr \in VR$.

6.2.4.3 VANETs and BSNs

VANETs are defined as a set of vehicles C. We also used the GR graph to model the path planning of the VANET. However, not all routes are available for all types of vehicles. In the case of VANETs, we exclude dedicated bicycle-only lanes and edges representing train railway tracks. We used all other edges to generate a predefined path $pr \in PR$. For bicycles, we have a set of bikes K. For generating the path planning, we consider all available lanes except train railway tracks. The city region has some parks, such as Mooney's Bay and Hog's Back Park, with bike lanes. Consequently, some areas of the GR graph are only traversed by bicycles.

6.2.5 IoDSCF

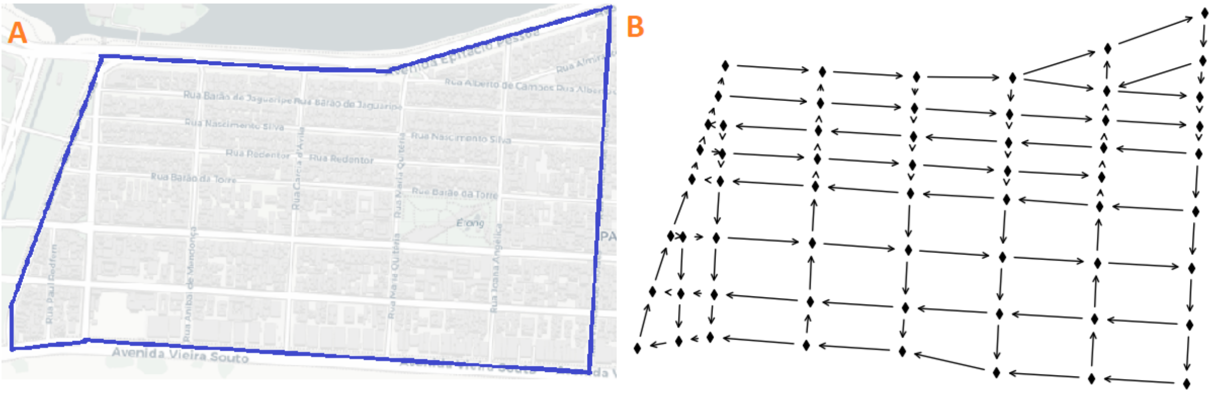
In this section, we will present the IoDSCF protocol which is the seminal protocol of IoDMix. In this protocol, we explore a partnership between IoD and Bus Network (BN). BNs are part of Public Transportation Networks (PTNs). PTNs have the characteristic of being frequent and maintaining the periodicity of the routes almost every day. Thus, it is possible to plan the path of the drones considering the public transport routes. For instance, [112] showed that the use of PTN can enable drones to extend their effective range in drone delivery.

6.2.5.1 Definitions and Assumptions

A BN similar to a real scenario is used to simulate the hybrid network. We use the RioBuses[1] dataset. RioBuses dataset had its data collected from the public transport system in Rio de Janeiro, Brazil. The dataset was collected on October 1, 2014. The area used is part of the Ipanema neighborhood, as illustrated in Figure 6.3 – Part A.

In this work, we use the architecture introduced in Section 4.1. Thus, we have the airways parallel to the streets. A graph of the main roads was generated to create the simulation of routes and airways, illustrated in Figure 6.3 – Part B. Also, most of the airways have the same direction as their correspondent road in Ipanema.

Figure 6.3: A - Ipanema neighborhood, Rio de Janeiro, Brazil. B - Graph of Ipanema



Source: Elaborated by the author

Each airway has only one direction. To better adapt the map to the simulator used, only the rightmost section had its direction changed in relation to the real roads. In real life, all streets are bottom-up, and we use top-down. The scenario used has approximately 15,154 m of airways. The airways are cylindrical (see Figure 6.4) with a radius of 1 m. Thus, using Equation 6.1, the volume is 47,583.56 m^3. Another point is that we used the same scenario in the ground, so we have 15,154 m of roads.

$$vc = \pi \times radius^2 \times h \tag{6.1}$$

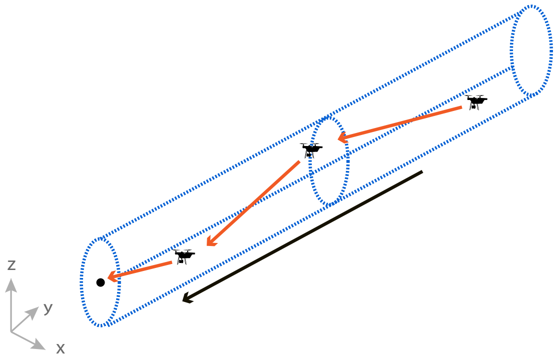
Since the airways are cylindrical, the drone can fly within this space but not beyond the edges. Drones do not fly in a straight line and may suffer some action from the sternum

---

[1] https://crawdad.org/coppe-ufrj/RioBuses/20180319/

that affects their trajectory, such as the wind. Thus, the drone always seeks to follow the center of the airway. In the developed scenario, the takeoff and landing airways were not specified. Each simulated drone has a starting point and an ending point on the route. Thus, it is assumed that there is an airway for takeoff and another for landing at the points chosen for the beginning and end of the route.

Figure 6.4: Example of airway



Source: Elaborated by the author

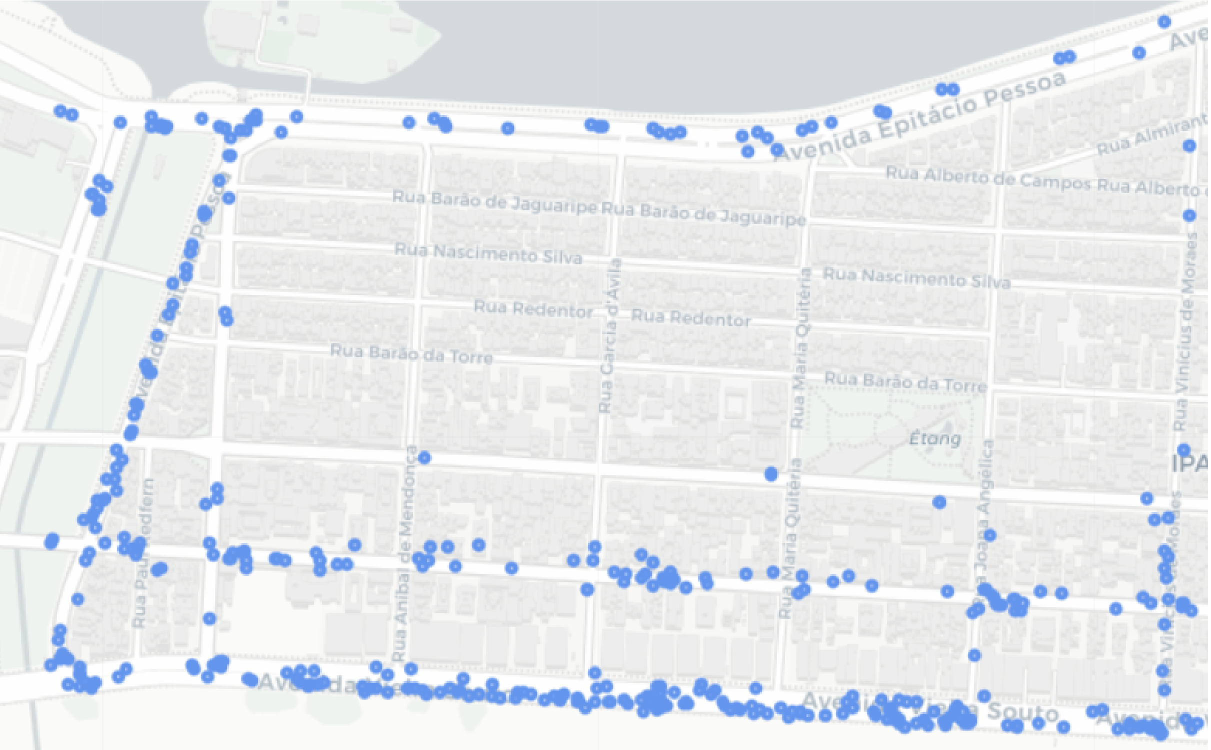
6.2.5.2 Bus trajectories-aided Drones path planning

Considering a scenario where drones will work together with other networks in a smart city, drones need to take advantage of some intrinsic characteristics of cities traditional networks. PTNs are very consolidated networks. Tracking buses using IoT is one of the most explored applications in this field [276]. For instance, users can track the buses in real time and know better if a bus already pass a bus stop or not [276]. An essential feature of the network is that it is quite predictable, considering the day of the week and the time of day.

Considering these facts, BNs can collaborate with IoD. Drones can have their path plannings linked to the bus routes for this collaboration to be efficient. Figure 6.5 illustrates the bus density in Ipanema, Rio de Janeiro, Brazil, on October 1, 2014, between 11:00 am and 11:10 am. It is possible to observe which streets have bus routes and which are often used in this case. Thus, let us assume that a drone delivery company is located in Ipanema and has delivery in that same area. The drone that will make the delivery can have its route planned to give preference to airways in which there are also bus routes on the ground. This fact allows collaboration between networks.

It can be observed in Figure 6.5 that in the observed time interval, there are areas in which the bus density is high. There are also areas where density is low and some

Figure 6.5: Density of bus routes in the Ipanema neighborhood, Rio de Janeiro, Brazil. Date: 2014-10-01 from 11:00:00 to 11:10:00



Source: Elaborated by the author

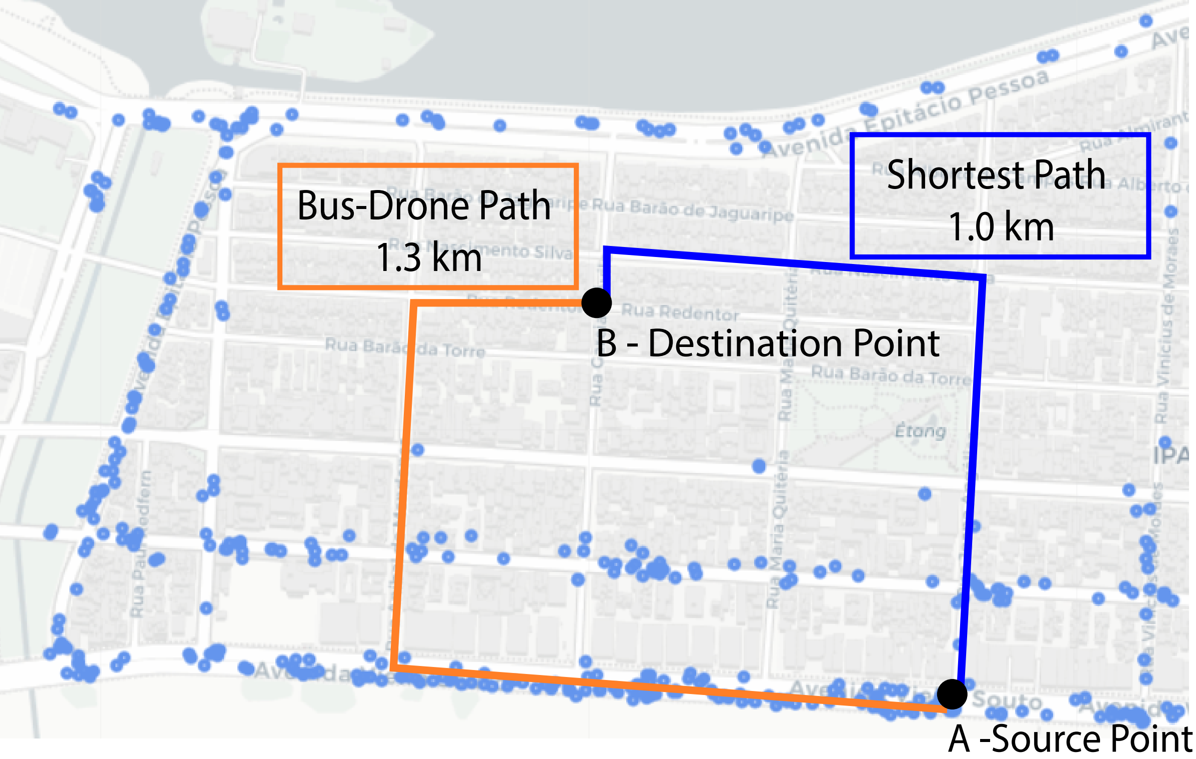
where buses do not pass. It is quite reasonable to assume that there are many similar real-life scenarios. Thus, as shown in the example, some deliveries may be set outside the coverage area of BNs. The ideal scenario for a drone delivery would be for the shortest path to be fully covered by the BN. However, this is not always possible. In Figure 6.6, a drone needs to deliver a package from point A to point B.

The shortest path is 1.0 km, but only a small part of it allows collaboration between buses and drones. The important point is that the airways and land routes in this example are one-way. On the other hand, the bus-drone path presents a possibility in which communication between drone and bus is possible in almost all the paths. In particular, this path is a little longer, totaling 1.3 km .

Algorithm 10 generates the drone paths, and we considered two options. The first is the shortest path based on distance (line 1) and the second is the shortest path based on bus routes (line 2). For both cases, the Dijkstra algorithm was used. For the generation of the shortest path based on bus routes, each edge of the graph had its weight based on the number of buses that traveled that edge in the period shown in Figure 6.5. Thus, the more buses traveled that edge, the more likely it was to be chosen.

Equation 6.2 defines which path the drone would take (line 3). As the distances in the scenario used are relatively small, we chose to use a *maxFactor* of 30%. So, the path bus-drone route is only chosen if the distance of the source to the destination point was

Figure 6.6: Example of shortest path and bus-drone path to delivery a good of point A to point B



Source: Elaborated by the author

```
Algorithm 10: Drones path planning generation considering bus routes
   Input   : matrix $G$, $source$, $destination$, $maxFactor$
   Output: path
1  $sp \leftarrow$ shortestPath($G$, $source$, $destination$)
2  $bdp \leftarrow$ busDronePath($G$, $source$, $destination$)
3  $maxSortestPath \leftarrow$ sp.distance $\times$ $maxFactor$
4  if ($bp.distance \geq maxSortestPath$) then
5      return $sp.path$
6  return $bdp.path$
```

less than or equal to 30% (line 4).

$$maxSortestPath = distance \times maxFactor \tag{6.2}$$

In real delivery scenarios, the route chosen can make a difference in delivery cost. Regardless of the distance of the route, the drone will only be authorized to start the delivery if it has adequate energy capacity to do so. Routes that allow hybrid communication, such as bus-drone routes, can be considered safer, as they have greater connectivity when compared to routes that do not have bus traffic. Thus, the final user will be able to choose the type of delivery, and safer routes can have a higher price when compared to traditional routes.

6.2.5.3 Protocol Description

We present two versions of IoDSCF. The first does not consider the BN and second has hybrid communication between BN and IoD. In both cases we have the following assumptions:

- Each drone has an identification number;
- Each drone has a GPS and consequently know its own position;
- Each drone know the position of the ZSP;
- This environment has airways in only one altitude;
- The airways have a cylindrical shape.

The IoDSCF protocol works as follows. Each drone transmits a beacon every 1 s. In this version, the nodes are just drones. When a drone receives a beacon from other drones, it stores the position of its neighbors. Algorithm 11 shows the packet routing to deliver a message. In this case, the variable *Node* is a drone. To start transmitting a message targeting ZSP, the source drone first checks if it has neighbors (line 1). If no neighbors are found, the drone will store-carry the message and wait for neighbors (line 10).

Algorithm 11: Packet routing algorithm

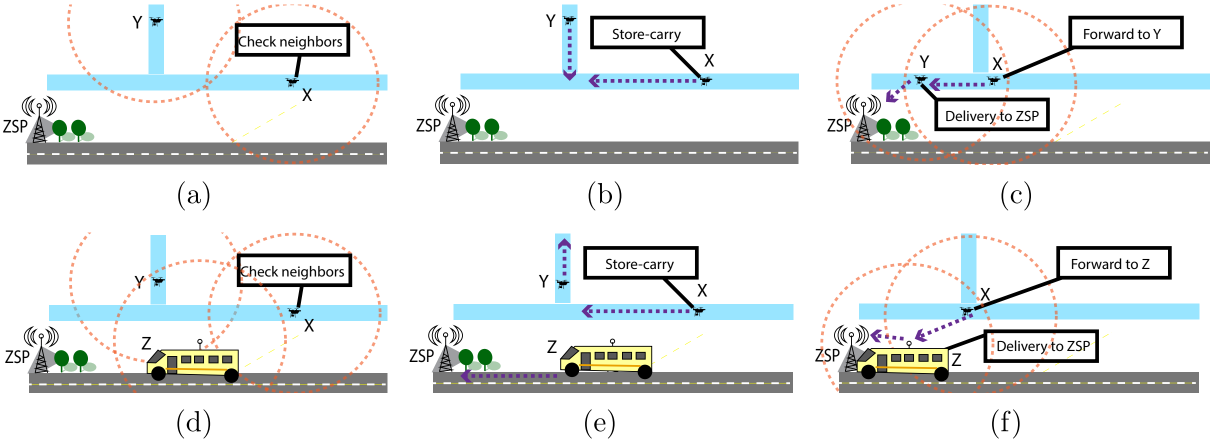
```
   Input  : Packet $M$
1  if ( Node has neighbors ) then
2      if (ZSP in Node.neighbors) then
3          Deliver $M$ to ZSP
4      else
5          if Node has drone $\in$ neighbors then
6              Forward $M$ to the nearest drone neighbor to the ZSP
7          else
8              Forward $M$ to the nearest bus neighbor to the ZSP
9  else
10     Store, carry and wait for neighbors
```

If the drone has neighbors, two scenarios can occur. First, the ZSP is a neighbor of the source drone. In this case, the drone deliveries the packet P to ZSP (line 3). Otherwise, the source drone needs to choose a neighbor to forward the message (line 6).

To choose a node, the drone checks which neighbor is closer to the ZSP position. As buses are not considered in this version, lines 5, 7 and 8 are not needed.

Figure 6.7 illustrates how IoDSCF protocol works. Drone X wants to transmit a message to the ZSP. Initially, it checks if it has neighbors (Figure 6.7a). As it has no neighbors, it store-carry the message (Figure 6.7b). Upon finding a neighbor (Figure 6.7c) drone X forwards the message to drone Y. Drone Y checks if it has neighbors and then delivers the message to the ZSP.

Figure 6.7: Example of IoDSCF without using Bus Network and with Bus Networks



(a) (b) (c)

(d) (e) (f)

Source: Elaborated by the author

When buses is aid to IoD the objective is to fill the gaps in drones communication with buses. Algorithm 11 also represents the Bus-Drone packet routing. The main difference is that the variable *Node* can now be a drone or a bus. When a drone wants to transmit a message, it checks if it has neighbors. Neighbors can be either drones or buses. Another difference is the addition of lines 5, 6 and 8. In this case, preference is given for the message to be transmitted to another drone.

Assuming that more than one neighbor is found, it checks if drones are neighbors (line 5). If there are drones, the drone closest to the ZSP is chosen to transmit the message (line 6). Otherwise, the bus closest to the ZSP is chosen (line 8). Preference is given to drones, as drones have more speed and a longer range than buses. In addition, buses may be subject to interventions such as tunnels and urban canyons. Thus, buses are used only when necessary to fill the gaps in the IoD.

Figure 6.7 also illustrates how IoDSCF protocol works with BNs as nodes. First drone X checks if it has neighbors (Figure 6.7d). Similar to Figure 6.7d, the drone has no neighbors. Thus, it is necessary store and carry the message (Figure 6.7e). If the same scenario in Figure 6.7b occurs, drone X will transmit the message to drone Y, and, in the sequence, drone Y will deliver the message to the ZSP. However, it drone Y flies to the opposite direction, the message will never reach the ZSP. In this case, bus Z can be used as intermediate node to deliver the message to ZSP (Figure 6.7f).

Assuming the Z bus does not exist in the example in Figure 6.7. Two scenarios can occur. First, drone X heads towards the ZSP. In this case, the message would be delivered to ZSP, but the end-to-end delay would be more significant. In the second case, drone X could take the same direction as drone Y and stop going towards the ZSP. In this case, without the presence of the Z bus, the message would never be delivered to the ZSP. Therefore, buses can be essential to deliver messages in sparse scenarios.

6.2.5.4 Performance Evaluation

To simulate the network, we use IoDSIM. IoDSIM admits an xml file that specifies the airways as input. In the evaluation, we use the scenario shown in Figure 6.3. Table 6.2 shows the configuration parameters used in the simulator. We perform 30 simulations for each configuration of number of drones $(10, 20, 30, 40, 50)$. Table 6.3 shows the drone density for each number of drones used. We opted for low drone densities because the focus of the study is sparse scenarios. The drones density was calculated using Equation 6.1. The drone speed was based on the real speed of drones DJI phantom 4 pro[2] and DJ Spark[3]. For the bus speed, we set an interval between 8 m/s and 12 m/s. We consider that a bus has an average speed lower than the drone. Specifically, buses often stop at bus stops for boarding and alighting passengers. Thus, we consider them faster in urban scenarios than buses due to the drones' speed.

Table 6.2: IoDSCF Simulation configurations

Parameter	**Drones**	**Bus-Drones**
Mobility model	Gauss-Markov	Gauss-Markov
Number of Drones	10–50	10–50
Number of Buses	-	39
Simulated area	airways	roads and airways
Drone Speed	13 m/s – 16 m/s	13 m/s – 16 m/s
Bus Speed	-	8 m/s – 12 m/s
Transmission Range Drones	300 m	300 m
Transmission Range Buses	-	200 m
Simulation runs	30	30
Simulation time	120 s	120 s

Another point was the transmission range of the nodes. In the case of drones, when compared to buses, they are in an environment with less interference from obstacles such as buildings and houses. This fact provides a better transmission range. In the case of

---

[2] https://www.dji.com/ca/phantom-4-pro/info

[3] https://www.dji.com/ca/spark/info

Table 6.3: IoDSCF Drones density (m^3)

#Drones	Drones Density
10	0,0210 $\times 10^{-2}$
20	0.0420 $\times 10^{-2}$
30	0.0630 $\times 10^{-2}$
40	0.0840 $\times 10^{-2}$
50	0,1050 $\times 10^{-2}$

buses, they can suffer from tunnels and urban canyons that can make data transmission difficult [277]. Thus, we set the transmission range of buses smaller than that of drones. The value of the drone transmission range was defined based on the scenario presented in Figure 6.3. It is reasonable to assume that drones can reach neighboring drones on parallel airways.

Each of the 30 simulations has different path plannings for the drones. The same position was used for the ZSP in all path plannings. In addition, in each simulation, the source and destination points of each drone were randomly generated. Algorithm 1 determined which path each drone should follow to reach its destination. The path back from the destination to the origin was also designed. It is reasonable to assume that the drone returns to its origin after delivering a package.

An adaptation of the actual path traveled was considered to generate the bus routes. The paths were extracted from the RioBuses dataset. In total, 39 buses were used. In the simulator, all drones start up on the ground. Due to positioning of drones, the message transmission occurs at time $t = 12$. However, the first beacon is transmitted at time $t = 2$. Regardless of whether the simulation considers the BNs or not, it is always the drone with $id = 1$ that initializes the message. One message is initialized every 10 s. For the performance evaluation, the following metrics were considered:

- Packet Delivery Ratio: the number of packets successfully received for the ZSP divided by the number of packets generated by a drone;

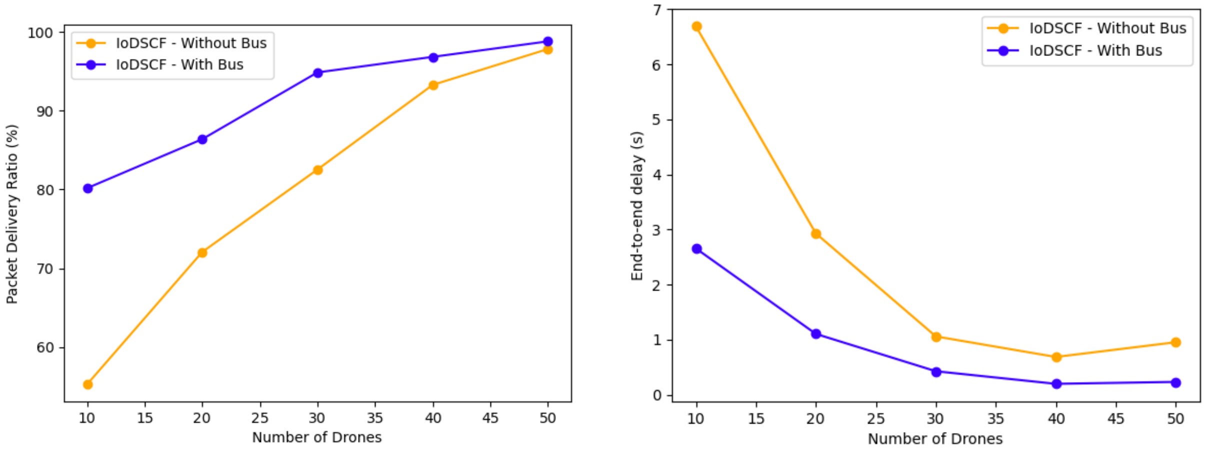
- End-to-end Delay: The average time needed for a packet generated by a drone to reach the ZSP;

- Nodes type chosen per route: The average of nodes type chosen per route. It was considering the total number of nodes used to reach the destination and each type of node separately. Specifically, how many of these nodes were buses and how many were drones;

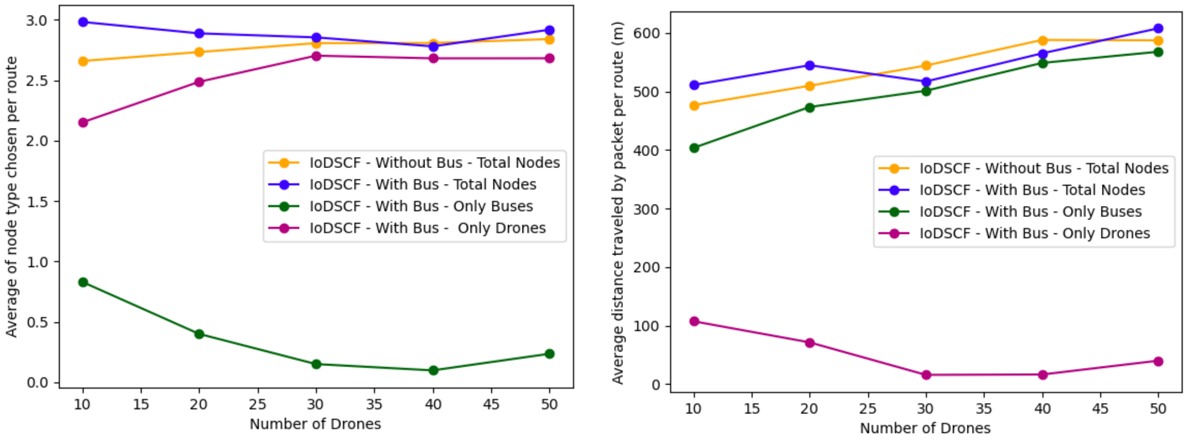
- Distance traveled by packet per route: The average distance traveled by packet per route. It considered the total distance and the distance traveled using drone transmissions, and the distance traveled using bus transmissions.

Figure 6.8a illustrates the PDR. The number of drones varied, as described in Table 6.2. The results show that the IoDSCF protocol using buses as part of the network allowed a greater number of packet delivery in all scenarios. However, as the number of drones approaches the number of buses (39), the difference between the results decreases. Thus, for 50 drones, the results are very close. Table 6.4 presents the summarized results of the packet delivery ratio metric.

Figure 6.8: IoDSCF evaluation results



(a) PDR (b) End-to-end Delay



(c) Avg. of Node Type Chosen per Route (d) Avg. Distance Traveled by Pckt. per Route

Source: Elaborated by the author

Figure 6.8b shows the end-to-end delay metric × number of drones. In the beginning, the delay is close to seven seconds in IoDSCF - without buses. This fact is because the network is sparse in this scenario. Therefore, the drone may need to carry the message for a long time before finding another drone to relay the message. With the use of the BN, it is possible to notice that the delay decreases are statistically smaller. As the number of drones increases, it can be seen that the delay decreases for both network configurations (with and without buses). However, as the drone number approaches the bus number,

Table 6.4: IoDSCF results for PDR

#Drones	Packet Delivery Ratio (%)	
	IoDSCF Without Buses	IODSCF With Buses
10	55.26 ± 1.92	80.16 ± 1.25
20	72.06 ± 1.36	86.37 ± 1.19
30	82.53 ± 1.02	94.84 ± 0.53
40	93.26 ± 0.58	96.81 ± 0.61
50	97.81 ± 0.46	98.78 ± 0.20

the difference between the delays is smaller. Table 6.5 presents the data summarization of the end-to-end delay metric.

Table 6.5: IoDSCF results for end-to-end delay

#Drones	Delay (s)	
	IoDSCF - Without Buses	IoDSCF - With Buses
10	2.6568 ± 0.2360	6.6920 ± 0.6683
20	1.1033 ± 0.0999	2.9289 ± 0.2196
30	0.4251 ± 0.0262	1.0575 ± 0.0955
40	0.1963 ± 0.0110	0.6843 ± 0.0484
50	0.2315 ± 0.0054	0.9526 ± 0.1233

Figure 6.8c shows the average of node types chosen per route. In this scenario, it can be observed that, on average, both for the case where BNs are used and for those that are not used, the messages are delivered with a maximum of 3 hops. For the IoDSCF with BN, it is observed that in cases where the number of drones is smaller than the number of buses, the network takes on average more hops to deliver the message than the IoDSCF network without BN. Again, when the number of drones and buses is close, the protocols perform similarly. It is also observed that as the number of drones increases, the number of buses used decreases. Table 6.6 summarizes the average of nodes types chosen per route.

Table 6.6: IoDSCF results for average of type node chosen per route

	Average number of hops			
#D	IoDSCF - Without Bus	IoDSCF - With Bus		
		Total	Drones	Buses
10	2.658 ± 0.017	2.981 ± 0.016	2.150 ± 2.485	0.831 ± 0.013
20	2.732 ± 0.012	2.886 ± 0.012	2.485 ± 0.009	0.400 ± 0.007
30	2.805 ± 0.011	2.779 ± 0.010	2.702 ± 0.009	0.151 ± 0.007
40	2.807 ± 0.009	2.779 ± 0.009	2.680 ± 0.009	0.099 ± 0.003
50	2.839 ± 0.009	2.917 ± 0.010	2.680 ± 0.009	0.235 ± 0.004

Figure 6.8d shows the average distance traveled by the packet per route. Note that, in the beginning, the distance traveled when using BNs is greater than when not using

it. Then this fact is reversed, the distance being greater when drones are not used. It is believed that this is due to the fact that at the beginning, as the message travels more by bus and buses have a smaller range, more nodes are used, making the distance traveled greater in the end. However, when the number of drones approaches the number of buses, the simulation with only drones starts to travel a greater distance because drones have a longer range, so other nodes can be reached, causing the distance traveled to increase.

As with other metrics, as the number of drones increases, the distance traveled by buses decreases. On the other hand, the distance traveled by drones increases. In the scenario of 10 drones, the package traveled 107 m. For 30 drones, the message travels, on average, only 15 m. This fact also reflects what was observed in Figure 6.8c, because as the number of hops of each type of node decreases, so does the average distance traveled in each network. Table 6.7 shows the summarized results for the distance metric.

Table 6.7: IoDSCF results for average distance traveled by packet per route

	Average distance (m)			
#D	IoDSCF Without Bus	IoDSCF Total	IoDSCF Drones	IoDSCF Buses
10	476.49 ± 6.65	510.81 ± 5.95	403.43 ± 5.40	107.38 ± 3.07
20	509.64 ± 5.05	544.60 ± 1.94	473.32 ± 4.15	71.27 ± 1.94
30	544.13 ± 4.29	516.8 ± 3.37	501.10 ± 3.30	15.75 ± 0.71
40	587.88 ± 3.90	564.99 ± 4.08	548.60 ± 3.96	16.39 ± 0.75
50	587.21 ± 3.92	607.69 ± 3.44	567.69 ± 3.35	40.00 ± 0.96

6.2.6 IoDMix

This section introduces IoDMix – a routing protocol for delay-tolerant Internet of Drones in a multi-vehicle scenario. First, we detail the packet routing algorithm. After, we present the selection of nodes using the expected message delivery time. Finally, we describe the selection of nodes using the distance metric.

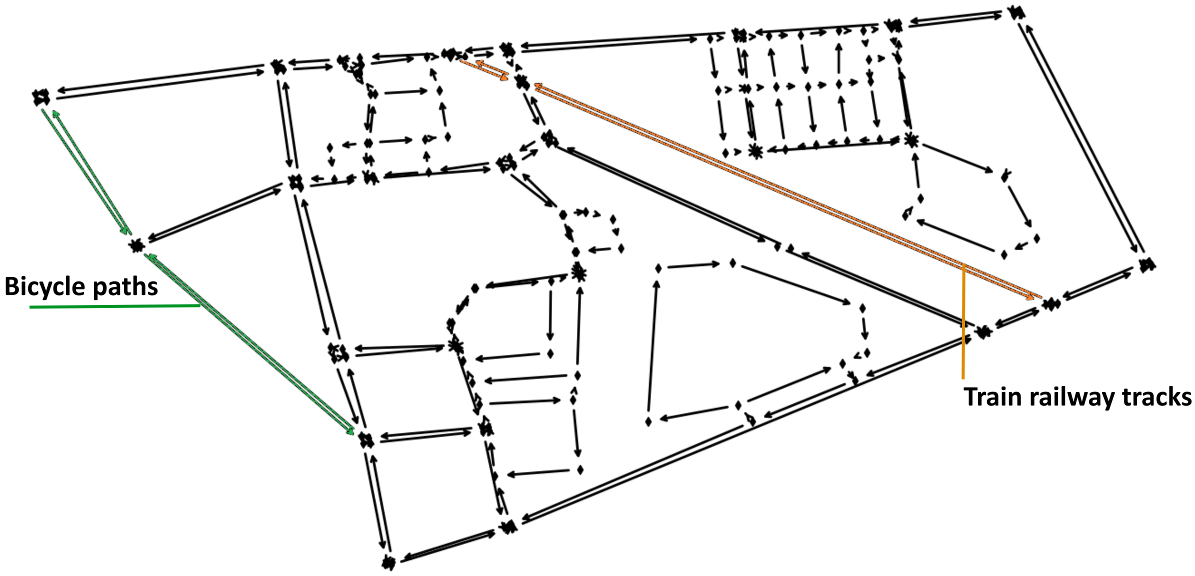
6.2.6.1 Scenario Modeling

IoDMix is a store-carry-forward protocol. This protocol considers that the drone wants to send a message to a ZSP. However, the protocol can send a message to order nodes with a fixed and known position (e.g., roadside units, IoT node). We make the following assumptions:

- Each drone has an identification number;
- Each drone has a GPS and, consequently, knows its position;
- Each drone knows the position of the ZSP;
- This environment has airways at only one altitude. However, other number of altitudes for airways can be considered, and our proposals are not limited nor dependent on these topologies;
- The drones, buses, and trains know their path.

Figure 6.9 represents the map used in this work. We considered the region of Riverside and Ridgemont neighborhoods in Ottawa, Ontario, Canada. We chose this city region because it has a stretch of track for the local train, park areas with bicycle paths, and some main roads. We define land routes as follows:

Figure 6.9: Map based on Riverside Park and Ridgemont regions in Ottawa, Canada



Source: Elaborated by the author

- Roads Architecture: The set of roads and their intersections form a directed graph $GR = (VR, ER)$. Each intersection is a vertex $vr \in VR$, and a road is an edge $er \in ER$. Also, we include in the graph the O-Train railway tracks that pass through this region and a stretch of paths exclusively for bicycles;
- Exclusive land routes: Thus, as in the real scenario used, some roads are exclusive for a single type of vehicle use. Figure 6.9 highlights the bike-only lanes and train tracks.

6.2.6.2 Packet Routing Algorithm

The IoDMix protocol works as follows. Each drone transmits a beacon every 1 s. The drone that receives the beacon stores the time and the sender drone *id*. This protocol generates a message by choosing a random drone that targets the ZSP (message recipient). Algorithm 12 shows the overview of packet routing. The variable *node* is a drone. First, the node checks if it has neighbors (Line 1). If the drone has no neighbors, it stores the message and waits for new neighbors (Line 19). If so, it checks whether the ZSP is among them (Line 2). If the ZSP belongs to the neighbors, the drone delivers the message to it (Line 3). Otherwise, the drone needs to assess whether to transmit the message to an available neighbor or to continue carrying the message.

Algorithm 12: Packet routing algorithm

```
   Input : Packet $M$
 1 if (node has neighbors) then
 2     if (ZSP in Node.neighbors) then
 3         Deliver $M$ to ZSP
 4     else
 5         nodeTime $\leftarrow$ bestNeighborTime(node) (Algorithm 13)
 6         if (nodeTime $\neq -1$) then
 7             Forward $M$ to nodeTime
 8         else
 9             currentTime $\leftarrow$ getTimeToZSP(node)
10             if (currentTime $\neq -1$) then
11                 Store, carry and wait for new neighbors
12             else
13                 nodeDistance $\leftarrow$ bestNeighborDistance(node) (Algorithm 14)
14                 if (nodeDistance $\neq -1$) then
15                     Forward $M$ to nodeDistance
16                 else
17                     Store, carry and wait for new neighbors
18 else
19     Store, carry and wait for neighbors
```

The first step is to check if any neighboring node has an expected message delivery time to the ZSP (Line 5). Some types of vehicles, such as drones, buses, and trains, can verify if they will be able to deliver the message to the ZSP. Specifically, these vehicles can give a time expectation for delivering a message to the ZSP. This process is possible because drones can access their path planning and provide an expected message delivery time considering their average speed. The same is possible for vehicles such as buses and trains, as they have a predictable route. We use Algorithm 13 to search for a node with an expected message delivery time among the neighbors of the current node.

If Algorithm 13 returns a valid node, the message is forwarded to it (Lines 6 and 7). Otherwise, we check if the current node has an expected message delivery time (Line 9). If so, two possible situations have occurred in Algorithm 13. The first possibility is that no neighboring node has a better message delivery time expectation than the current node. The second case is that no neighbor can calculate the expected message delivery time. Therefore, we keep the message with this node and wait for new neighbors (Lines 10 and 11). This is because we assume it is more advantageous to keep the message in a node that expects to reach the ZSP than to forward it to a node with no expected message delivery time.

If the current node has no expected message delivery time, we check which neighbor is the closest to the ZSP to forward the message (Line 13). In addition to the distance, we consider the priority of the vehicle, according to Algorithm 14. If a neighbor is found, the message is forwarded to it (Lines 14 and 15). Otherwise, the current node continues loading the message until new neighbors are found (Line 17).

- **Searching for nodes with expected message delivery time.** The node search algorithm considers the message delivery time expectation as follows. First, the expected message delivery time of the current node is calculated using the getTimeToZSP() function (Line 2). Afterward, all neighbors are traversed, and for each one, the expected message delivery time is obtained using the getTimeToZSP() function (Lines 3 and 4). This function returns -1 when the expected delivery time cannot be calculated. If the expectation of the neighbor n is different from -1 (Line 5), it is also necessary to check if the expectation of the *currentTime* is also different from -1 (Line 6). If *currentTime* is different from -1, we need to check if *newTime* is less than *currentTime* (Line 7). If it is smaller, the *currentTime* and the *neighbor* variable are updated (Lines 8 and 9). Thus, the message is not transmitted if the identified neighbors do not have a better delivery time expectation than the current one. If *currentTime* equals -1, we update the variables *currentTime* and *neighbor* to the corresponding values (Lines 11 and 12).

- **Discovering the nearest neighbor**. Algorithm 14 works as follows. First, we obtain the distance from the current node to the ZSP (Line 2). Then we check if there are drones among the neighbors (Lines 3 and 4). If there is, we look for the closest drone to the ZSP (Line 5). Also, that drone found must be closer than the distance from the current node to the ZSP. We prioritize drones as they have a greater range than land vehicles. This range is because land vehicles can suffer interference from obstacles, as mentioned in Section6.2.3. If no drone is found, priority is given to the PTN nodes (Lines 7, 8, and 9). This priority is because PTNs are more reliable since cars and bicycles are more likely to be disconnected from the network at any time by the user. For example, when a person parks the car, the node is no longer connected to the network. Priority is given to

Algorithm 13: Best neighbor

```
Input  : node
Output: neighbor
$neighbor \leftarrow -1$
$currentTime \leftarrow$ getTimeToZSP($node$)
for ($n$ in $node.neighbors$) do
    $newTime \leftarrow$ getTimeToZSP($n$)
    if ($newTime \neq -1$) then
        if ($currentTime \neq -1$) then
            if ($newTime < currentTime$) then
                $currentTime \leftarrow newTime$
                $neighbor \leftarrow n$
        else
            $currentTime \leftarrow newTime$
            $neighbor \leftarrow n$
return $neighbor$
```

VANETs and bicycles in the sequence (Lines 11 to 17). This is because VANETs have a bigger range than bicycles, as they have a greater chance of having more robust hardware.

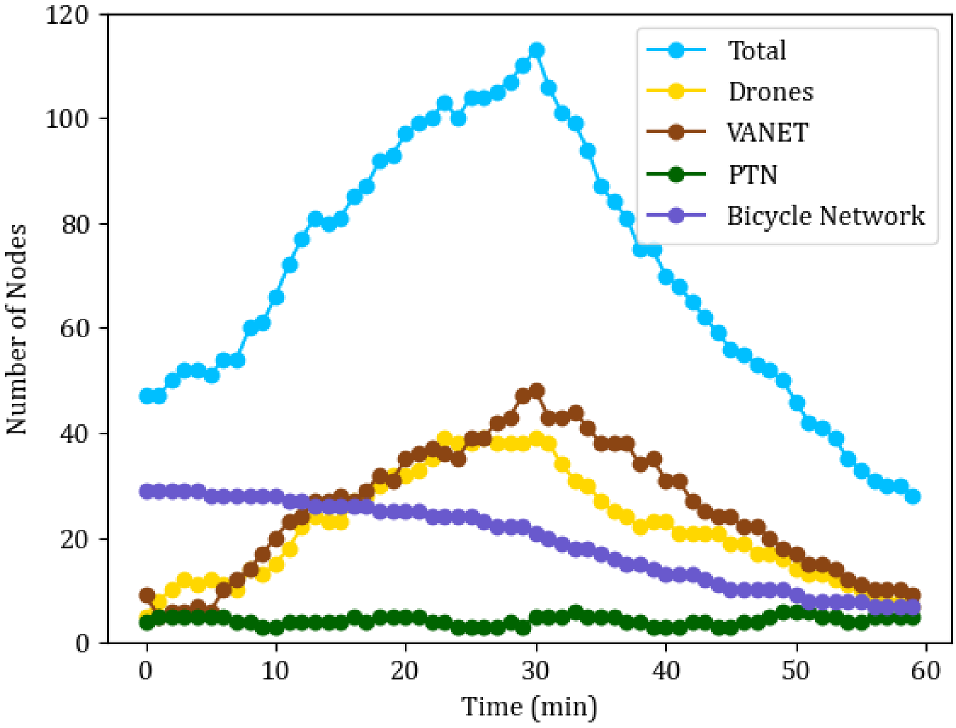
Algorithm 14: Best neighbor distance

```
   Input  : node
   Output: neighbor
1  $neighbor \leftarrow -1$
2  $currentDistance \leftarrow$ distanceToZSP($node$)
3  $list \leftarrow$ getDrones($node.neighbors$)
4  if ($list.size \neq 0$) then
5      $neighbor \leftarrow$ nearestNeighbor($list, currentDistance$)
6  else
7      $list \leftarrow$ getPTN($node.neighbors$)
8      if ($list.size \neq 0$) then
9          $neighbor \leftarrow$ nearestNeighbor($list, currentDistance$)
10     else
11         $list \leftarrow$ getCars($node.neighbors$)
12         if ($list.size \neq 0$) then
13             $neighbor \leftarrow$ nearestNeighbor($list, currentDistance$)
14         else
15             $list \leftarrow$ getBikes($node.neighbors$)
16             if ($list.size \neq 0$) then
17                 $neighbor \leftarrow$ nearestNeighbor($list, currentDistance$)
18 return $neighbor$
```

6.2.6.3 Simulation Modeling

To evaluate IoDMix, we used the scenario described in Section 6.2.4. To model the PTN, we used the bus lines that pass through the map in Figure 6.9. Specifically, eight main bus lines were considered in this region (bus lines 6, 88, 90, 92, 111, 141, 190, and 290). During the week, the lines have a time interval between 15 and 20 min. In addition, we also consider that the O-Train passes every 7 min. We model the traffic of IoD and VANET following a normal distribution. Therefore, until the middle of the simulation, there is an increase in vehicles. After that, the traffic of vehicles decreases. For bicycle modeling, we consider a decreasing curve. The scenario considered was in the morning, when there is greater traffic of bicycles that decreases as people start to go to work. Therefore, the traffic of VANETs increases. Figure 6.10 illustrates the number of vehicles in the network every 1 min.

Figure 6.10: IoDMix Number of nodes vs. Time (min)



Source: Elaborated by the author

Table 6.8 presents the settings for each type of vehicle. Since we considered a sparse scenario, we have a low number of drones. As mentioned, we use eight bus lines, so 16 buses can be active simultaneously (two from each line, going in opposite directions). We also consider that two trains can be operational at the same time. Node speeds were based on actual vehicle speeds. In the case of drones, we use the average speed reached by the DJI Phantom 4 pro[4] and DJ Spark[5] models. In this simulation, we assumed that the storage capacity of the drones was unlimited. However, in cases of limitation, it is possible to save only a limited number of neighbors using criteria such as nodes with higher transmission priority (e.g., node type, and route predictability, to name a few).

Table 6.8: IoDMix Simulation configurations

Parameter	**Drones**	**Buses**	**Trains**	**Cars**	**Bicycles**
Max no. of nodes	50	16	2	50	30
Region	airways	roads	track	roads	roads
Speed (m/s)	13 – 16	8 – 12	9 – 10	9 – 13	4 – 5
Transmission range (m)	400	300	350	300	250

Another point is the transmission range of each node. The transmission range of the drone was based on previous studies in the literature [38, 58]. The drone should have the greatest range when choosing the transmission radius of the other vehicles. We assume that the drone has the most extended range between the nodes because the airspace has fewer obstacles for transmissions. We also consider that cars and buses have a similar transmission capacity, as they travel in similar situations and suffer from urban canyons. However, we suppose trains have a slightly higher transmission capacity due to their hardware equipment. In the case of BSN, given the vehicle size, it is reasonable to assume that the used hardware will not be as robust as in other vehicles, so the bicycles have the lowest range in the network.

We use IoDSIM[6] – a simulator for the Internet of Drones developed on OMNeT++[7] and INET[8] library. IoDSIM allows drones to use only airways to fly and configure them to operate at different altitudes. The time simulated was 1 h. We made 30 simulations. Each has different path plannings for the drones and random locations for the ZSP. However, we placed the ZSP always at one road crossing. The simulation starts at time $t = 2$. In every simulation, the drone with $id = 0$, i.e., drone 0, starts transmitting the message at a different location and is active during the entire simulation. One message is generated every minute. In this simulation we used the IEEE 802.11n standard for the MAC and physical layers.

---

[4] https://www.dji.com/ca/phantom-4-pro/info

[5] https:/ /www.dji.com/ca/spark/info

[6] https://iodsim.manna.team/

[7] https://omnetpp.org/

[8] https://inet.omnetpp.org/

6.2.6.4 Results and Discussion

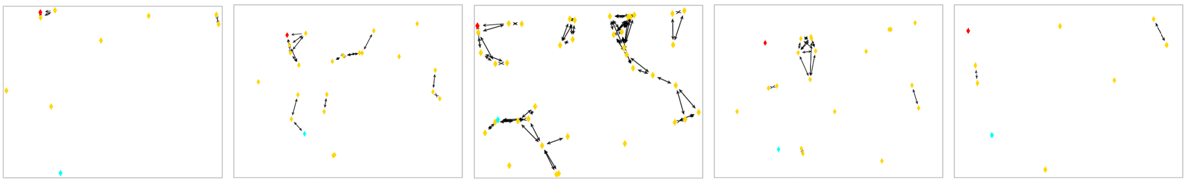
In this section, we present the simulation results and their discussion. First, we analyze the network connectivity in the simulation scenario. After, we discuss the IoDMix protocol.

► **Network connectivity**. IoD has high network dynamics, where nodes can connect and disconnect quickly. Furthermore, in the DTIoD, all the network elements do not stay connected. Figure 6.11a illustrates the connectivity of the IoD at $t = 1\,\text{min}$. At the beginning of the simulation, the network has few active drones, making it impossible to maintain the connection between them. Figure 6.11c shows the IoD connectivity at $t = 30\,\text{min}$, middle of the simulation. Around this time, the network will register the largest number of nodes in the simulation. Even so, we see that the drone network is fragmented. In this example, the source drone (in blue color) does not have a way to send a message to the ZSP (in red color).

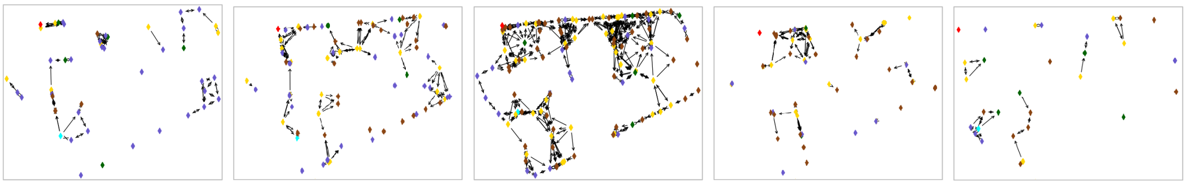
Figure 6.11f shows the network connection for simulation time = 1 min. The scenario is the same as in Figure 6.11a, but now with all nodes in the network. We observed that the communication between nodes is still challenging because there are few nodes in the network. Figure 6.11h shows the network connectivity for time = 30 min. In this case, with the help of other vehicles, it is possible to build a connection between the different fragments in Figure 6.11c. However, as the speed of IoD and VANET nodes is high, and bicycles and cars can disconnect from the network at any time, the network still has a high dynamicity and a high chance of fragmenting frequently.

The network connectivity is intrinsically related to the density of nodes. IoD network density can vary in each city region. In downtown, we can have constant traffic of drones, while in the city outskirts, we can have sporadic nodes. Classifying the entire IoD as a sparse or dense network is unrealistic since each region has characteristics that can affect the number of nodes. Consequently, some city regions tend to have more nodes (main roads and urban centers), and others consistently fewer (small neighborhoods and residential areas only). Furthermore, the time of the day and the day of the week can impact the IoD density. For instance, large centers can have more drone traffic during lunch and dinner hours due the food delivery. All these aspects reinforce the need to apply a store carry-forward protocol. Figure 6.11 shows a change in the network density in the used scenario. Even in cases with the highest network density, it is possible to have isolated or small groups of isolated nodes. Figures 6.11f to 6.11j show the evolution of the network density during the simulation considering all nodes.

Figure 6.11: IodMix Network connectivity overview



(a) ZSP and Drones; time = 1 min (b) ZSP and Drones; time = 15 min (c) ZSP and Drones; time = 30 min (d) ZSP and Drones; time = 45 min (e) ZSP and Drones; time = 60 min



(f) All nodes; time = 1 min (g) All nodes; time = 15 min (h) All nodes; time = 30 min (i) All nodes; time = 45 min (j) All nodes; time = 60 min

🟥 ZSP 🟨 IoD 🟦 Source Drone

🟫 VANET 🟩 PTN 🟪 Bicycle Network

Source: Elaborated by the author

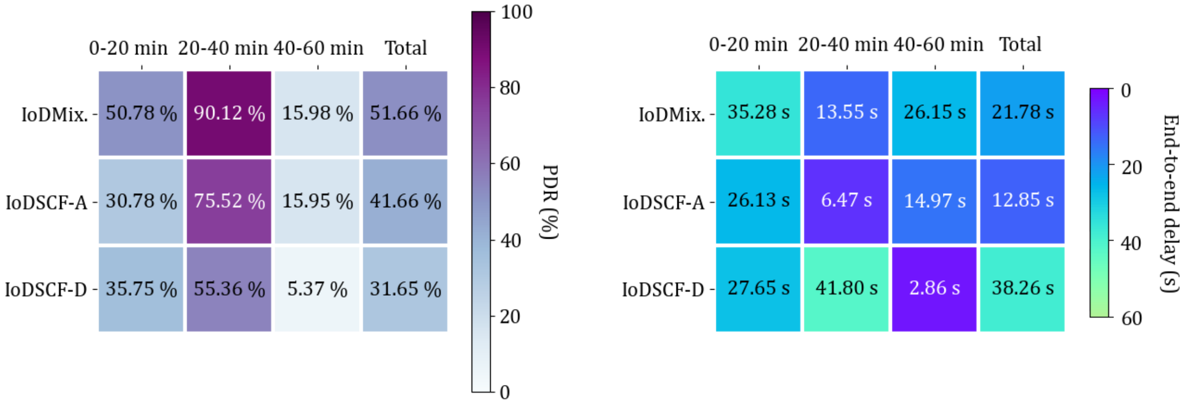
► **IoDMix analysis**. This section presents the analysis of the IoDMix protocol. We compared our results with the IoDSCF Protocol in two scenarios. Specifically, the first scenario uses all network nodes (IoDSCF-A), and the second uses only drones (IoDSCF-D). We chose this protocol because it is a store-carry-forward protocol for IoD. Therefore, considers airways for drones. The main criterion for IoDSCF to select the next node is the distance to the ZSP. For the IoDMix evaluation, the following metrics were used:

- Packet Delivery Ratio (PDR): the number of packets successfully received for the ZSP divided by the number of packets generated by the source drone;
- End-to-end Delay: the average time needed for a packet generated by the source drone to reach the ZSP;
- Type of nodes chosen per route: the average type of nodes chosen per route. It was considering the total number of nodes used to reach the destination and each type of node separately;
- Carry-time per node type (s): the average time a node carries a message. It was considered the total number of nodes used to reach the destination and each type of node separately;
- Packet Loss per Type of Vehicle: the average number of packets lost per each type of vehicle. We also considered the number of packets lost in the source drone.

Packet Delivery Ratio: To analyze the results, we divided the simulation into three parts. The first is up to 20 min, the second from 20 min up to 40 min, and the third from 40 min up to 60 min. Figure 6.12a shows a heatmap of the PDR results. The total column represents the overall delivery rate in the entire simulation. In all scenarios, IoDMix has a higher delivery rate than the other scenarios. However, in the final part of the simulation, the performance of IoDMix is similar to the IoDSCF-A. The interesting point is that the IoDSCF-D performs better than the IoDSCF-A in the first 20 min. IoDSCF-A may choose to send a message to a node with a low range, such as a bicycle, and as the network is very dynamic, the message can be lost more easily. Nevertheless, in the middle of the simulation, due to the density of nodes, the IoDSCF-A has a higher PDR than the drone-only version. At the end of the simulation, the IoDSCF-D has difficulty delivering packets because the network is sparse.

End-to-end Delay: The end-to-end delay metric was also analyzed by dividing the simulation into three stages. Figure 6.12b shows the heatmap of the results. We observed that the delay between messages is higher for IoDMix than for the IoDSCF-A scenarios. Thus, IoDMix prioritizes the choice of nodes that guarantee the delivery of the message. In the DTIoD scenario, it is more important to deliver the message than the time taken to deliver it. In the case of the IoDSCF-D scenario, the nodes need to carry the message for a long time until they find a neighbor closer to the ZSP to transmit it. As observed in Figure 6.11c, even in cases where the simulation has more drones, the network remains disconnected. Another point is that in the last part of the simulation, the IoDSCF-D has a low delay. However, as seen in Figure 6.12a, the delivery is very low in this case. Specifically, the packets that can be delivered are already very close to the ZSP.

Figure 6.12: IoDMix results divided into three simulation parts



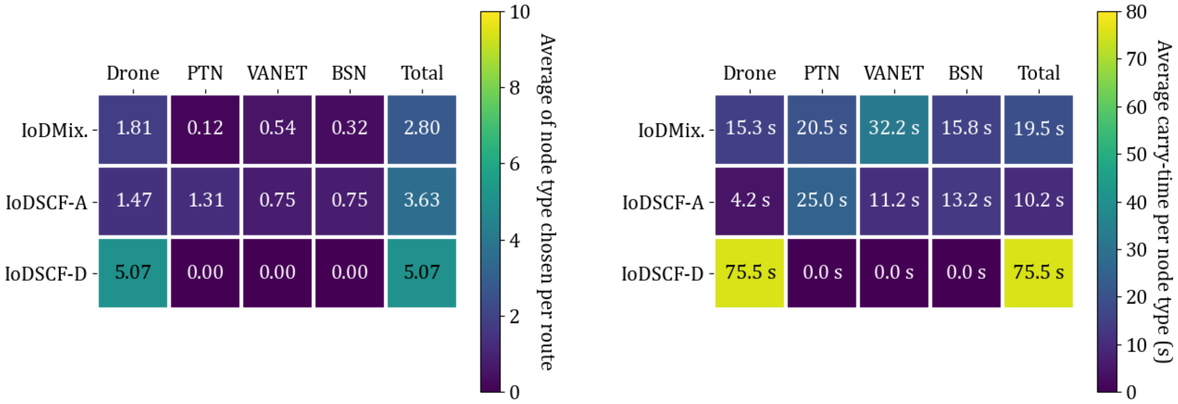
(a) PDR

(b) End-to-end Delay (s)

Source: Elaborated by the author

Type of nodes chosen per route. Figure 6.13b presents the results for the type of node chosen per route. The total column represents the average number of nodes used to deliver a message. IoDMix uses fewer nodes than IoDSCF-A because it carries the message longer, while IoDSCF-A forwards the message to any neighbor closer to the ZSP. In the case of IoDMix, it is observed that the different available vehicles were not frequently used. The reason for this is related to this scenario, in which the protocol did not find nodes that could easily guarantee the delivery of the message. As observed in Figure 6.13b, the use of the PTN network was low due to the small number of nodes in the region. In addition to the drones, recall that PTN is the only network where it is possible to predict the path a node takes.

Figure 6.13: IoDMix results by type of vehicle



(a) Number of Nodes type chosen per route (b) Average carry-time per node type (s)

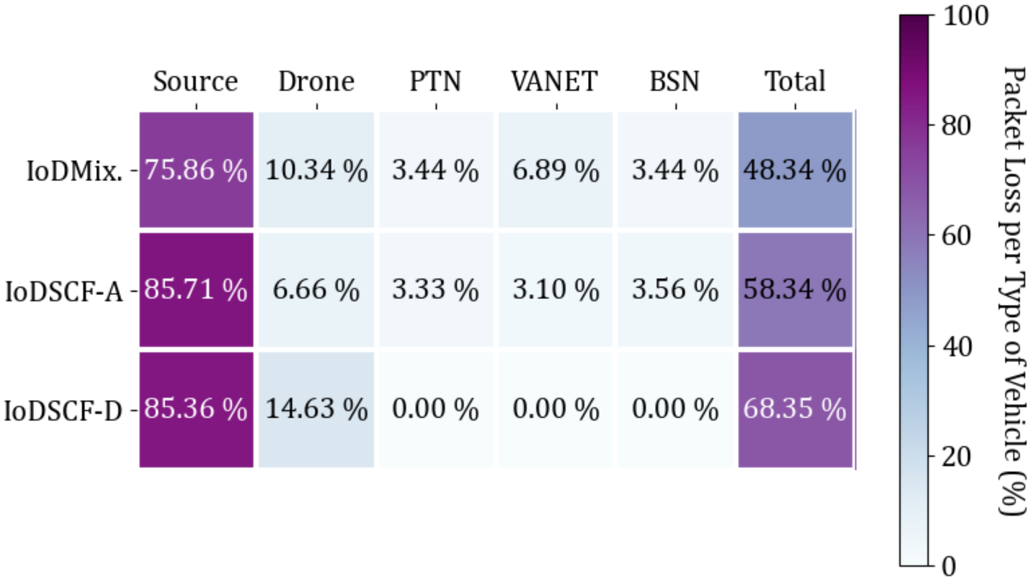
Source: Elaborated by the author

Carry-time per node type: Figure 6.13b presents the average time that each node type carried a message. The total column represents the average time a message was carried during the simulation. We observed that the IoDSCF-D does not find neighbors easily, so the drone takes the message for an extended period. Comparing the IoDMix with the IoDSCF-A, we observed that, in general, the IoDSCF-A forwards the message with greater speed. Particularly, IoDMix only forwards the message if it finds a node more likely to deliver it than the current node.

Packet loss per type of vehicle: Figure 6.14 presents the result of the Packet Loss per vehicle. The first column represents the drone that generated the message. We observed that most messages could not reach the ZSP, as the drone source could not contact any other network element. This fact occurs in all scenarios. In the first and last part of the simulation, about 20 messages are generated (33% of the total messages generated). At the beginning and end of the simulation, the network could not maintain connected components to transmit the message to the ZSP. They were not delivered in any of the scenarios and protocols evaluated. Therefore, the number of lost messages from source drones is high.

Drones lose most messages. The following factors contributed to this result: (i)

Figure 6.14: IoDMix Packet loss per type of vehicle



	Source	Drone	PTN	VANET	BSN	Total
IoDMix.	75.86 %	10.34 %	3.44 %	6.89 %	3.44 %	48.34 %
IoDSCF-A	85.71 %	6.66 %	3.33 %	3.10 %	3.56 %	58.34 %
IoDSCF-D	85.36 %	14.63 %	0.00 %	0.00 %	0.00 %	68.35 %

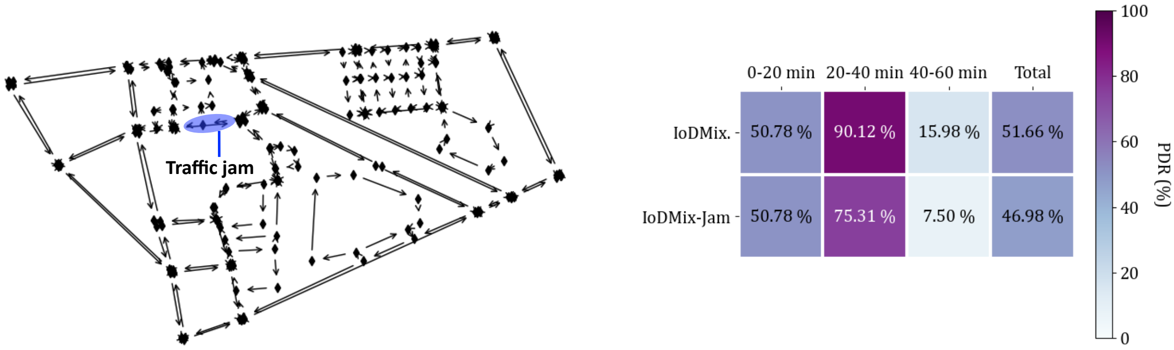
Source: Elaborated by the author

drones have the highest speed between nodes and can frequently disconnect from the network; (ii) drones can enter roads that may not receive another land vehicle for a while, affecting message loss. For example, depending on the network's density and the map's characteristics, when a drone enters an airway above the train line, it may not find neighbors to relay the message. For IoDMix, the second vehicle that loses messages the most is the car. This probably occurred because there are more cars on the network than other nodes. As for the IoDSCF-A, the loss of messages is similar in other vehicles on the network. This occurred because the protocol does not use priority between vehicles to transmit the message. And finally, IoDSCF-D has only messages lost by drones, as it is the only type of vehicle on the network.

► **Traffic jam analysis**. As mentioned earlier, PTNs have high predictability. However, two types of predictability must be considered: location and time. Since PTNs have fixed routes, vehicles try to stay on the established route as much as possible. However, path changes may occur due to accidents, infrastructure renovations, or temporary road blockages, which may cause vehicles to make detours. Many of these deviations are programmed and can already be considered in the vehicle's path planning.

Another type of predictability in PTNs is time, as vehicles are expected to arrive at a particular time in a specific location so that users can plan to use public transport. However, the time may not always be entirely predictable due to traffic jams that may occur on the road. Therefore, we analyze the occurrence of traffic jams in the network. Their location was chosen by identifying a point where there was necessarily a bus line, as shown in Figure 6.15a. We consider that a traffic jam occurred on a land road after 20 min of simulation. The traffic jam affected cars and buses and was considered the same in all 30 simulations.

Figure 6.15: IoDMix Traffic jam location and PDR result



	0-20 min	20-40 min	40-60 min	Total
IoDMix.	50.78 %	90.12 %	15.98 %	51.66 %
IoDMix-Jam	50.78 %	75.31 %	7.50 %	46.98 %

(a) Traffic jam location

(b) PDR

Source: Elaborated by the author

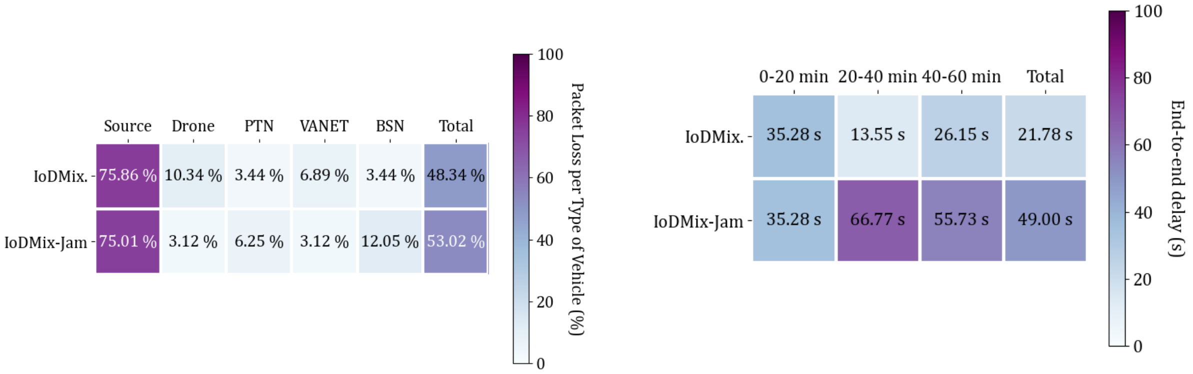
As a result, it was observed that there was a significant decrease in the PDR. As shown in Figure 6.15b, when considering the total number of delivered messages, there was an average decrease of 5%. This decrease occurred because the traffic jam prevented vehicles from moving, even if they did not have a message to deliver. This meant they could not fill communication gaps in other areas of the network if they were on their usual route. Additionally, it was observed that buses stuck in congestion played a crucial role in bringing more distant messages closer to the ZSP.

To better understand where packets were lost in the traffic jam, Figure 6.16a illustrates the packet loss by vehicle type. There was a significant increase in packet loss for BSNs. In our simulation, BSNs have a low transmission range, making them highly dependent on other vehicles to deliver the message to the final destination. Another factor is that bicycles are unpredictable and can easily disconnect from the network, increasing the chances of packet loss. Although there are packet losses in VANETs and PTNs, they are not as high. This is because vehicles with the highest transmission priority are drones. Therefore, even in a traffic jam, drones can pass through the region and receive messages from cars or buses.

It was also observed that there was an increase in delay for delivered packets. Figure 6.16b summarizes the obtained results. Considering that the traffic jam did not affect UAVs and bikes, a stopped vehicle could forward the message to another node (UAV or bike), allowing for message delivery. However, waiting for a node caused delivery delays, increasing traffic jams.

In summary, congestion can affect protocol performance. For the analyzed scenario, it was observed that the traffic jam prevented bicycles from finding more efficient nodes

Figure 6.16: IoDMix Traffic jam packet loss and end-to-end delay result



(a) Packet loss per type of vehicle (b) End-to-end Delay (s)

Source: Elaborated by the author

to forward the message. In conclusion, the consequences of a traffic jam depend on several factors, such as the congestion's location, the traffic jam's time, and the network's density. However, a traffic jam can generally result in increased message loss and delays in message delivery.

6.2.7 Discussion

IoD has unique features, such as the use of airways and highly dynamic topology. In addition, the IoD aims to provide coordinated access to drones to airspace. This feature will provide an environment with dynamic network density. In cases where the network is sparse, the IoD might consider a delay-tolerant network. In this work, we address a sparse scenario. We propose a partnership between IoD and VANET, PTNs, and BSNs to fill IoD communication gaps and integrate them into Intelligent Transportation Systems. In an urban application, it is reasonable to assume that distinct networks can collaborate to improve their network metrics.

The IoD-terrestrial networks collaboration is promising. Works in the literature show that ground vehicles can be a means to allow more extended trips, energy savings, and even be a charging point for drones in operation. These advantages and the high predictability of some terrestrial networks allow routing protocols to exploit the collaboration IoD-terrestrial networks to improve efficiency, message delivery, and decrease delay, as shown in the IoDSCF and Delay-tolerant Internet of Drones Protocol in a Multi-vehicle Scenario (IoDMix) results.

6.3 Join IoD and Other Aerial Networks

It is expected that the airspace will soon contemplate multiple networks with different types of nodes. Among these networks are IoD and the Internet of Flying Cars (IoFC). Although they share the same airspace, each network has different characteristics such as node type, altitude, speed, power supply, and processing capacity. Recently, a new trend was raised to evolve these networks towards an integrated network [278]: Hybrid Aerial-terrestrial Network (HATN). Therefore, in this section, we will explore the integration of IoD and other aerial networks.

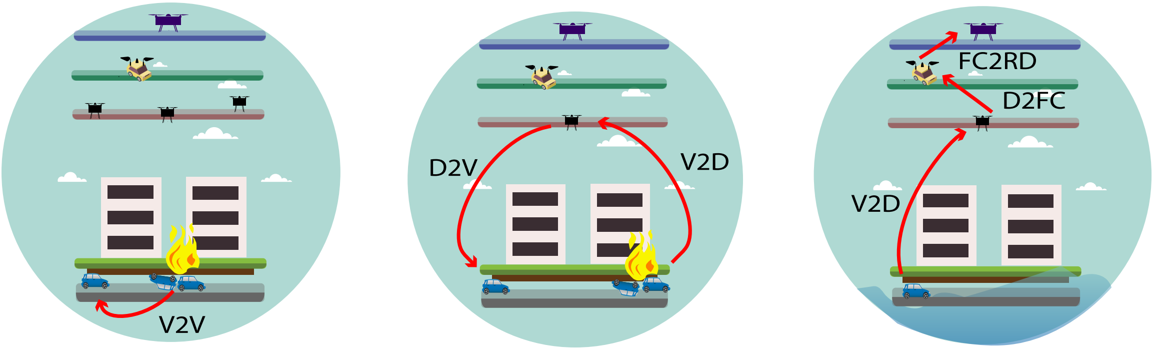
6.3.1 Preliminaries

Similar to IoD, flying cars also have the potential to form constant traffic of vehicles in the sky. This idea aims to solve the problem of traffic congestion and release tensions on existing city transport networks [279]. Different services can be provided by autonomous flying cars to the population shortly. Thus, it will be necessary for these cars to communicate with each other to assemble an autonomous network, forming the IoFC. In addition, this network will compete for airspace with IoD. Therefore, it is reasonable to assume that these networks communicated to maintain the organization and security of the devices and their users.

IoD is composed of different types of drones. For instance, drones can have distinct sizes, processing capacities, and communication ranges. A recent trend is to use high and low-altitude UAVs to improve network coverage [238]. High Altitude Platform Stations (HAPS) contribute significantly to ubiquitous connectivity due to their location (stratosphere) [280] and because nodes can process, store, and collect data from a plurality of applications [280], providing an extensive coverage among nodes. On the other hand, low altitude platforms such as drones and helikites [281] can improve local connectivity by providing a small coverage among nodes [238]. In this context, Robust Drones Platforms Networks (RDP) can support other networks with the same features that HAPS but locally.

For this protocol, we divided the IoD into two categories. First, the Task Drones (TD) is used to perform the applications, such as the delivery, monitoring, and urban sensing. Secondly, RDPs are used to support TD. These drones have higher processing and storing capacities when compared with TD. As the objective of all these networks is to provide services to the population, they must communicate with terrestrial networks

Figure 6.17: Different scenarios where a car needs to broadcast an emergency message



(a) Using Only Terrestrial Communication (b) Using UAVs to Fill Communication Gaps (c) Using Only Aerial Communication

Source: Elaborated by the author

for applications and services purposes. In addition, there can be a mutual collaboration between networks to provide constant node connectivity. Although the airspace can be divided into operating ranges for each air network, and there is no horizontal dispute, the nodes of each network can compete for the airspace vertically (for takeoffs and landings). As RDPs have ample storage and processing capacity, they can collaborate with many applications in other networks. Each RDP may be responsible for maintaining and processing information from a network and application type.

In the literature, it is common to find studies that evaluate the use of drones to support routing protocols in VANETs [110]. This protocol considers different aerial networks, and each of them can help VANETs in different situations. The decision depends on two factors. The first is the network density. The second is the emergency type that directly influences the availability of nodes. For example, when an accident or emergency occurs, it is common to find works where an alert message warns the responsible authorities and nearby vehicles [282]. The message can be transmitted only through terrestrial networks or with the help of other networks. As illustrated in Figure 6.17, the alert message can go through different nodes. Next, we discuss three scenarios where a car wants to send a message in an emergency.

- **Using Only Terrestrial Communication**: As illustrated in Figure 6.17a, some scenarios are suitable for broadcasting an alert message using only terrestrial communication. For example, in a car accident, if the network is dense, the alert is forwarded using only VANET. In the literature are many efforts to transmit an alert message in this situation [282]. In this scenario, it is reasonable to assume that the other cars in the network will be available to retransmit messages because only the vehicles involved in the accident are in an emergent situation. Thus, a message can be forwarded using Vehicle-to-Vehicle Communication (V2V);

- **Using UAVs to Fill the Communication Gaps**: As previously discussed, it is possible to broadcast a message using only terrestrial communication, depending on the network density. However, VANETs can present sparse scenarios, and drones are a way to help with routing [110] (Figure 6.17b). An example is when a car accident occurs, and no vehicles nearby can relay the emergency message generated by the accident. In these cases, the purpose of drones is to be an intermediary node between two cars to fill gaps in VANET communication. In addition to V2V communication,Vehicle-to-Drone (V2D) and Drone-to-Vehicle (D2V) communications are used;
- **Using Only Aerial Communication**: Figure 6.17c presents a scenario in which the message cannot travel through cars. A possible scenario would be an urban flooding. This scenario is unpredictable, and the affected vehicle cannot determine whether adjacent vehicles can relay an emergency message. Thus, it is not possible to use other networks only to fill the communication gaps in VANETs. Therefore, the message needs to go through only available nodes. One option would be to send the message to an RDP. In that case, RDP would use 6G technology or beyond to notify authorities and other cars around the disaster region. Send a message to an RDP to make the alert. To reach an RDP, the message can go through different aerial networks like TD and IoFC. Thus, it is necessary to use V2D, Drone-to-Flight Car (D2FC), and Flight Car-to-Robust Drone (FC2RD) communications.

This Section aims to introduce a routing protocol called Altitude-based Routing Protocol for Hybrid Aerial-terrestrial Networks (ARAT). The focus of this protocol is to be a hybrid reactive protocol for emergencies where a lower layer node needs to send information to a specific RDP. ARAT is a hybrid protocol that uses geocast and store-carry-forward approaches.

6.3.2 ARAT Protocol Description

This section describes the Altitude-based Routing Protocol for Hybrid Aerial-terrestrial Networks (ARAT). Considering the multi-network scenario, ARAT is a hybrid geocast and store-carry-forward protocol that takes advantage of the network topology to send the node to the highest altitude. Algorithm 15 details the packet sending. In ARAT, when a node sends a packet, it checks if it is still relevant (Line 1). A packet $packet_i$ is relevant if and only if the following conditions are satisfied:

- Node receives $packet_i$ for the first time;

- $t \in [t_{start}, t_{end}]$;
- Node is inside the transmission zone TZ.

After sending $packet_i$ (Line 2), the protocol schedules a new sending of $packet_i$ for the time $t + 5$ (Line 3), where t represents the current time. As long as the source node does not hear the $packet_i$ retransmission (Line 4), it assumes that no neighbors have received it. Thus, the node stores/carries the packet until the time $t + 5$, in which it tries a new transmission. This process continues until the source node listens to the packet or it reaches the t_{end} maximum wait time limit. If this packet is not relevant, it has already been received by the node at least once. Thus, if there is a transmission schedule for this packet, the function in Line 8 removes it from the queue.

Algorithm 15: Send Packet

```
  Input : packet pckt
1 if (pckt is still relevant) then
2     send(pckt)
3     scheduleSend(pckt, $t + 5$)
4     while not listen pckt in time $t_{end}$ do
5         send(pckt)
6         scheduleSend(pckt, $t + 5$)
7 else
8     cancelSchedule(pckt)
```

We assume that the node knows its position and type of node. Each message has a unique code. The sent packet has its id, destination information, and the sender's position. Algorithm 16 details packet dissemination. The message transmission region (TR) is a parallelepiped, composed of three dimensions $\langle length, width, A \rangle$. The A corresponds to the height of the highest airway, and the *length* and *width* are application-dependent. Upon receiving a relevant packet, the node checks its type. If it is not an RDP, Algorithm 17 calculates the time T for scheduling a message. If it is an RDP, the node knows where to send the message to reach the destination, and the schedule is made for the next second (Lines 6 and 7).

Algorithm 17 shows the calculation of time for scheduling a message. First, to calculate the time T, we consider only the altitude A and calculate the time t_a based on the Euclidean distance. We compute the distance between the source node and the highest airway and the current node and the highest airway (Line 1). Then we calculate the time t_d, which is related to the other two dimensions.

In the previous step, we used altitude A to compute the first value to determine the scheduled time (Line 1). In this step (Lines 2–5), we consider the longest dimension not yet used, i.e., the *length* or *width*, and form a rectangle using these measurements, as shown in Figure 6.18. After, we calculate the Euclidean distance $D(Node_s, R)$, which

Algorithm 16: Packet Dissemination

```
Input   : packet $pckt$
if ($pckt$ is relevant and $Node.id \neq pcket.dest$) then
    if ($Node \notin RDP$) then
        $T \leftarrow$ timeSchedule()
        scheduleSend($pckt, T$) (see Algorithm 3)
    else
        $T \leftarrow$ currentTime() + 1
        $nextNode \leftarrow$ getNextNode(Node.id, pcket.dest)
        scheduleSend($pckt, T, nextNode$)
$delete(pckt)$
```

Algorithm 17: Time Schedule

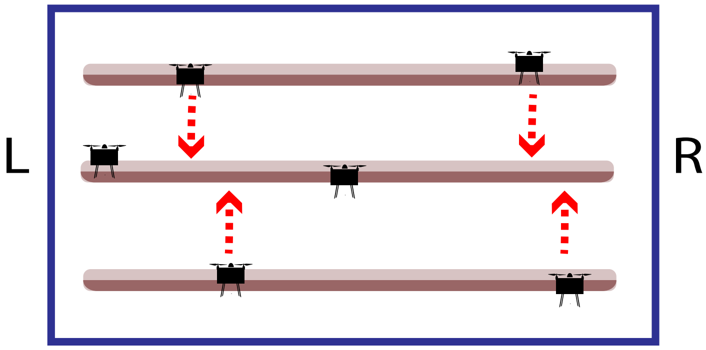
```
  Input   : packet $pckt$
  Output: time $T$
1 $t_a \leftarrow D(Node_c, A)/D(Node_s, A)$
2 if ($D(Node_c, R) \leq D(Node_s, R)$) then
3     $t_d \leftarrow D(Node_c, R)/D(Node_s, R)$
4 else
5     $t_d \leftarrow D(Node_c, L)/D(Node_s, L)$
6 $T \leftarrow (c_1 t_a + c_2 t_d) T_*$
```

represents the distance from the source node to the R-side, and the distance $D(Node_c, R)$, which represents the distance from the current node to the R-side. Thus, if the current node is further to the R-side than the source node, t_d is calculated according to Line 3, otherwise according to Line 5. This step's purpose is to give priority so that further away are the relays. Finally, the total time T is calculated by:

$$T = (c_1 t_a + c_2 t_d) T_*, \tag{6.3}$$

where c_1 and c_2 are coefficients that determine the priority of height and distance, and T_* is a random number between $[0, 1]$. Thus, we will have a set of N nodes with a message scheduled according to priorities $N_i < N_{i+1} < \ldots < N_n$. When node N_i retransmits the message, nodes $\in N$ that listen are no longer relevant and are removed from the priority list. This process eliminates unnecessary transmissions.

Figure 6.18: Altitude and environment example



Source: Elaborated by the author

6.3.3 Performance Setup

We use the IoDSIM[9] framework to evaluate the performance of our proposed ARAT protocol. IoDSIM is an IoD framework built on top of OMNeT++[10]. This simulator is composed of a mobility module that allows overlapping airways. Table 6.9 shows the number of nodes used in the performance evaluation. The number of cars and RDPs has not changed. The idea is to form a sparse network for cars that will need other communication forms to transmit an emergency message. So, we set the number of cars as 50. This is a reasonable assumption because cars are used mainly as message sources in this simulation setup. Also, it is appropriate to use a fixed number of RDPs as the network forms a constellation in the sky, and its topology does not frequently change.

Table 6.9: Number of Nodes

Cars	Drones	FC	RDPs	Total of Nodes
50	10 – 200	20 – 100	30	110 – 380

Table 6.10 presents the parameter settings for each network. We assume the following speed order FC > TD > C > RDP. We consider that the airways are parallel to the streets and are overlapping. The mobility model used for the Drones, FC, and RPDs was Gauss-Markov. At the MAC layer, we use the 802.11 protocol. In Equation 6.3, we set $c_1 = 0.8$ and $c_2 = 0.2$, because we considera higher priority to altitude. Also, we consider TZ to be the entire simulated area.

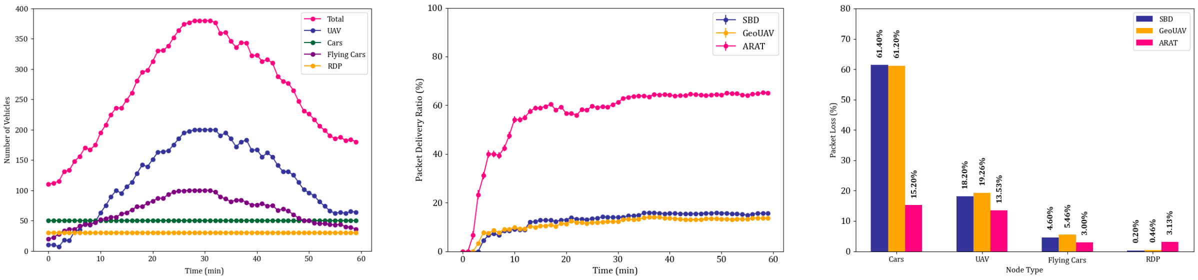
We performed 30 simulations for each scenario in Table 6.9. We used a simulation area of 10000 m × 10500 m × 1100 m. The duration of the simulation was 1 h. We set the cars' Transmission Range (TR) based on [283]. Due to the altitude, we consider that

---

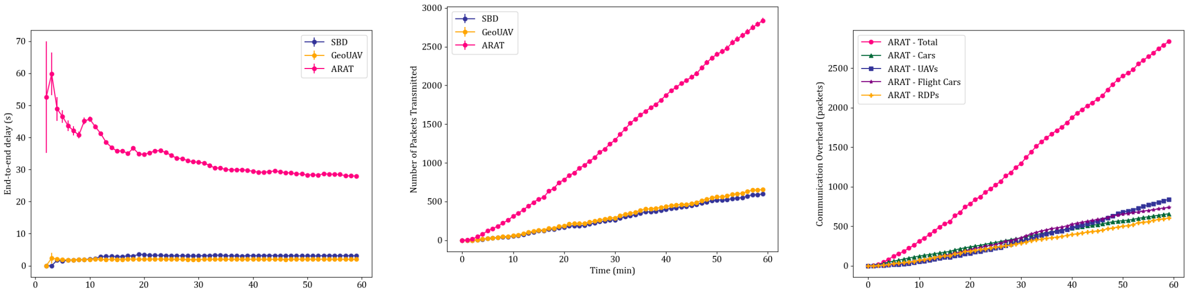
[9] https://iodsim.manna.team/

[10] https://omnetpp.org/

Figure 6.19: Simulation results



(a) Number of Vehicles (b) Packet Delivery Ratio (c) Packet Loss



(d) End-to-end Delay (e) Number of Packets Transmitted (f) Communication Overhead

Source: Elaborated by the author

FC will have more range than terrestrial cars. The drone transmission range was based on the drone type. TD has less transmission capacity, and RDP has a high transmission capacity. A random car sends a new alert message for a random RDP every 70 s.Also, for ARAT, the time to live the message depends on the type of vehicle. For cars, the time to live is 90 s and for other nodes is 70 s.

Table 6.10: Networks Configurations

Parameter	Cars	TD	FC	RDPs
Speed	10–13 m/s	14–17 m/s	25–30 m/s	5–7 m/s
TR	300 m	400 m	600 m	2000 m
Altitudes	Terrestrial	200 m	500 m	1000 m

6.3.4 Experiment Results

We compared our solution with two other approaches. The first is Schedule by Distance (SBD) – a geocast protocol in which packet dissemination is distance-based. In this case, the highest transmission focus is for the node closest to the RDPs airway. We chose this approach because prioritizing the node closer to the destination is a common approach in the literature used by different proposals [284]. The second approach is the GeoUAV protocol [58], a geocast protocol for fleet UAVs. In this case, the transmission priority considers the nodes' positions and the TZ. The closer a node is to a TZ edge, the more importance it has. We chose this approach because it is a geocast protocol for beacon-less drones with similarities related to node priority determination. In addition to node priority, the main difference for ARAT is that these protocols do not have a message retransmission mechanism when the scenario is sparse.

The results correspond to an average value of 30 runs with a 95% confidence interval. Figure 6.19a presents the network traffic. As we mentioned in Section 6.3.3, the cars and the RDPs have a fixed configuration. UAVs and flying cars have a traffic flow that follows a uniform distribution. As we observe in Figure 6.19a, 30 m is the time that we have more network nodes. Figure 6.19a is essential to understand the following metrics analyzed. First, Figure 6.19b presents the PDR metric. ARAT performs better in sparse networks due to its isolated node recovery mechanism (store-carry-forward). In the entire simulation, the PDR of SBD, GeoUAV, and ARAT are 15%, 13%, and 65%, respectively. ARAT represents an increase of around 50% compared to the other protocols.

SBD and GeoUAV have very low PDR because these protocols do not perform well in a sparse network. Figure 6.19c shows the packet loss rate in each protocol considering the different types of vehicles. In most cases, cars do not reach other types of nodes in GeoUAV and SBD. Both results are very similar because the message source is a car. On the other hand, due to the story-carry-forward strategy used in ARAT, we have a lower packet loss.

Figure 6.19d shows the results for the end-to-end delay metric. The use of the store-carry-forward technique in ARAT has two consequences, a higher delay and a higher number of packets generated. We observed that in the beginning, ARAT had a delay more significant than 50 s. Also, the confidence interval is higher because we have fewer nodes in the network. We can see in Figure 6.19a that for up to 10 min, the number of drones and flying cars is very low. So, the nodes must carry the message for a long time until they reach the other node. After the start, the number of delivered messages increases as the network becomes denser, and consequently, the end-to-end delay drops. Although we use a uniform traffic modeling for the network, in the end, it is not as sparse as in the beginning. Thus, the delay does not increase again and stabilizes around 28 s. In the case of SBD and GeoUAV, the performance in this metric is similar because, as we saw in the PDR analysis, the delivery of messages is low. Also, delivered messages can mean that the destination was closer and, therefore, faster to reach.

Figure 6.19e presents the results for the metric number of packets transmitted. As mentioned, the number of packets transmitted increases due to the store-carry-forward approach. The ARAT protocol sends more packets but has a higher delivery in sparse scenarios. This performance is because the protocol tries new transmissions every 5 s for a maximum time, which is different for each node type. Figure 6.19f shows the ARAT communication overhead. First, we can observe that UAVs and flying cars are the nodes that transmit more packets compared with the other nodes. Considering that we have more drones, the packets broadcast by UAVs are higher than those sent by flying cars. After we have cars, which are more relevant at the beginning of a message transmission since they are the source of the messages, and disseminate a significant amount of packets. Finally, RDPs transmit fewer packets than other vehicles considering the network has its nodes connected.

6.3.5 Discussion

It is important to note that due to the network dynamics, many scenarios are possible, and several solutions are needed. Therefore, the integration between networks can allow different solutions for emergency scenarios. Protocol development for hybrid networks must consider heterogeneous nodes with different sizes, mobility, speeds, and processing capabilities. In the literature, we have a few geocast protocols for UAVs [58]. However, they do not have a recovery mechanism when a message reaches an isolated node.

Scenario emergencies can be unpredictable and require efficient message delivery. Aiming at this scenario, we proposed ARAT protocol. This is the first protocol for HATNs, including IoD, to the best of our knowledge. ARAT performed better in sparse scenarios, as observed in the results presented in the previous sections. We expected this fact since the ARAT protocol was designed precisely for sparse network cases. That means that the denser the scenario, the greater the chance of other networks being available to collaborate and disseminate emergency messages.

6.4 Final Remarks

In this Chapter, we presented two routing protocols for joint IoD and Other Networks. The first one explores the IoD-BN partnership. The IoDSCF protocol is designed for delivering packets in scenarios with drones communication gaps. We evaluated our proposal using only drones and with the collaboration of Bus Networks. BNs are very predictable and have the potential to help in IoD communications. To enable the IoD, drones need to have an environment where they can be organized in the airspace, for this we used the topological architecture for smart presented in Section 4.1.

Additionally, we introduced a drone path-planning that considers the Bus routes to improve the collaboration between the networks. Simulation results demonstrated that IoDSCF protocol with Bus Networks collaboration outperforms the protocol without BNs by reducing the average end-to-end delay and maximizing the packet delivery ratio in sparse environments. An important fact observed in the results is that Bus Networks are more useful when the number of buses is greater than the number of drones.

The second protocol introduced was ARAT. ARAT uses the position information of the nodes to forward packets to the highest airway. We proposed a hybrid protocol that efficiently uses geocast and store-carry-forward approaches to deliver packets in a sparse network. Sparse scenarios can occur mainly in emergency scenarios. Simulation results showed that this new algorithm improves the packet delivery ratio and end-to-end delay compared with the baseline routing protocols.

Chapter 7

Conclusion and Future Work

This chapter summarizes our results 7.1. Following, we present the future research directions 7.2.

7.1 Summary of this Thesis

In this thesis, we divided our contributions into four parts. The First one is related to understanding the IoD characteristics and necessities through IoD-UC applications. Our survey in Chapter 3 demonstrates that IoD has unique features compared with other mobile networks, needing its routing protocols. Therefore, we identify a gap in the literature for the elemental network problems: communication and mobility. First, we address the communication problem by developing two routing protocols for IoD. Our results are summarized in Chapter 5. The first protocol IoDGR considers a geocast scenario where drones need to communicate with each other to transmit a message in an emergency scenario. The second, PSDIoD, is a protocol where we controlled the network using SDN concepts. The two protocols present good results, and to the best of our knowledge, these protocols are the only ones to consider airspace organized into airways and controlled by ZSPs.

After, we integrate the IoD with other networks. As IoD has the potential for multiple applications in the urban scenario, we expected that shortly, IoD needs to communicate with other networks. We summarized our results on integration between IoD and other networks in Chapter 6. The first protocol regards joint IoD and Terrestrial Networks. Our results show that it is suitable to use Bus Networks, PTN, VANET and BSN to fill the communication gaps in IoD in urban areas. The second protocol ARAT considers a futuristic scenario where we will have more than one drone network type in the sky. Our results demonstrate that in emergency scenarios, IoD can be used to transmit messages more quickly and efficiently to a base station.

The last contribution is related to the mobility problem. Using airways can bring different perspectives to traditional problems. In Chapter 4, we summarized our contributions to IoD path planning. We explore drone delivery applications in urban scenarios and compare them with typical deliveries. As a result, we propose guidelines to discover if in a determined scenario is advantageous or not to use drone delivery. We investigate a Coverage Path Planning problem in the IoD scenario. Our results show that our method for AntIoD has a good solution. These researches enable us to advance in studies about communication and mobility in the IoD environment.

7.2 Future Work

In the near future, it is anticipated that a substantial number of drones will provide services to the population, thereby forming a robust and intricate network in the airspace. Considering this scenario, future work on this thesis can be categorized into two main areas. The first area pertains to the development of IoD protocols, taking into account that IoD is still an emerging technology and its regulations and guidelines may evolve over time. Although the proposed protocols serve as a foundation for IoD protocols, it might be necessary to create new protocols as the network is implemented, aligning with updated guidelines, laws, and emerging technologies. These changes can influence not only protocols but also mobility, management, and the way of building applications in the urban environment.

The second point revolves around the utilization of data generated by drones. Over time, drones will produce increasingly more data. Some prospective research based on data from drones may encompass the following topics: identifying points of interest in the city by analyzing drone trajectories and comparing urban mobility patterns before the introduction of drone services; analyzing social behavior and discerning the spatiotemporal characteristics of cities for urban planning, particularly in environments where the city has IoD coverage; comprehending the impact of integrating drones with other modes of transportation, especially in the context of goods delivery and its contribution to smart resource utilization, including energy; leveraging aerial views from drones to enhance the semantic content of existing datasets used in UC (Urban Computing). Therefore, it will be possible to create solutions for the integration of heterogeneous datasets that have semantic information extracted with drones; and devise solutions for integrating diverse datasets enriched with semantic information extracted via drones and creating frameworks that mitigate data redundancy and overhead, ultimately determining the cost-effectiveness and energy efficiency of employing drones for specific tasks. These research directions are essential for unlocking the full potential of drones in urban environments and addressing the challenges associated with their integration into the urban scenario.

References

[1] Lav Gupta, Raj Jain, and Gabor Vaszkun. Survey of important issues in uav communication networks. *IEEE Communications Surveys & Tutorials*, 18(2):1123–1152, 2015.

[2] Samira Hayat, Evşen Yanmaz, and Raheeb Muzaffar. Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint. *IEEE Communications Surveys & Tutorials*, 18(4):2624–2661, 2016.

[3] Aziz Altaf Khuwaja, Yunfei Chen, Nan Zhao, Mohamed-Slim Alouini, and Paul Dobbins. A survey of channel modeling for uav communications. *IEEE Communications Surveys & Tutorials*, 20(4):2804–2821, 2018.

[4] Mohammad Mozaffari, Walid Saad, Mehdi Bennis, Young-Han Nam, and Mérouane Debbah. A tutorial on uavs for wireless networks: Applications, challenges, and open problems. *IEEE Communications Surveys & Tutorials*, 21(3):2334–2360, 2019.

[5] Mirmojtaba Gharibi, Raouf Boutaba, and Steven L. Waslander. Internet of drones. *IEEE Access*, 4:1148–1162, 2016.

[6] Federal Aviation Administration FAA. Uas remote identification, 2020. Available: `https://www.faa.gov/uas/getting_started/remote_id/`. Accessed: 2022-06-06.

[7] Muhammad Yeasir Arafat and Sangman Moh. Routing protocols for unmanned aerial vehicle networks: A survey. *IEEE Access*, 7:99694–99720, 2019.

[8] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. Urban computing: Concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(3):1–55, 2014.

[9] Paul McBride. Beyond orwell: The application of unmanned aircraft systems in domestic surveillance operations. *J. Air L. & Com.*, 74:627, 2009.

[10] Rachel L Finn and David Wright. Unmanned aircraft systems: Surveillance, ethics and privacy in civil applications. *Computer Law & Security Review*, 28(2):184–194, 2012.

[11] Christos Kyrkou, Stelios Timotheou, Panayiotis Kolios, Theocharis Theocharides, and Christos Panayiotou. Drones: Augmenting our quality of life. *IEEE Potentials*, 38(1):30–36, 2019.

[12] 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*, 7:128125–128152, 2019.

[13] Aicha Idriss Hentati and Lamia Chaari Fourati. Comprehensive survey of uavs communication networks. *Computer Standards & Interfaces*, 72(.):103451, 2020.

[14] Paul Fahlstrom and Thomas Gleason. *Introduction to UAV Systems*. John Wiley & Sons, New Delhi, India, 2012.

[15] Daniel Fernando Pigatto, Mariana Rodrigues, João Vitor de Carvalho Fontes, Alex Sandro Roschildt Pinto, James Smith, and Kalinka Regina Lucas Jaquie Castelo Branco. The internet of flying things. *Internet of Things A to Z: Technologies and Applications*, .:529–562, 2018.

[16] Naser Hossein Motlagh, Tarik Taleb, and Osama Arouk. Low-altitude unmanned aerial vehicles-based internet of things services: Comprehensive survey and future perspectives. *IEEE Internet of Things Journal*, 3(6):899–922, 2016.

[17] Seng W. Loke. The internet of flying-things: Opportunities and challenges with airborne fog computing and mobile cloud in the clouds. *arXiv preprint arXiv:1507.04492*, .:1–5, 2015.

[18] Thomas Lagkas, Vasileios Argyriou, Stamatia Bibi, and Panagiotis Sarigiannidis. Uav iot framework views and challenges: Towards protecting drones as "things". *Sensors*, 18(11):4015, 2018.

[19] Bin Li, Zesong Fei, and Yan Zhang. UAV Communications for 5G and Beyond: Recent Advances and Future Trends. *IEEE Internet of Things Journal*, 6(2):2241–2263, 2019. doi: 10.1109/JIOT.2018.2887086.

[20] Amira Chriki, Haifa Touati, Hichem Snoussi, and Farouk Kamoun. FANET: Communication, Mobility Models and Security Issues. *Computer Networks*, 163:106877, 2019. ISSN 1389-1286. doi: https://doi.org/10.1016/j.comnet.2019.106877. URL `https://www.sciencedirect.com/science/article/pii/S1389128618309034`.

[21] Alex Koohang, Carol Springer Sargent, Jeretta Horn Nord, and Joanna Paliszkiewicz. Internet of Things (IoT): From Awareness to Continued Use. *International Journal of Information Management*, 62:102442, 2022.

[22] Xiangjie Kong, Yuhan Wu, Hui Wang, and Feng Xia. Edge Computing for Internet of Everything: A Survey. *IEEE Internet of Things Journal*, 9(23):23472–23485, 2022.

[23] Sofiane Zaidi, Mohammed Atiquzzaman, and Carlos T. Calafate. Internet of flying things (ioft): A survey. *Computer Communications*, 165:53–74, 2021.

[24] Zhiqing Wei, Mingyue Zhu, Ning Zhang, Lin Wang, Yingying Zou, Zeyang Meng, Huici Wu, and Zhiyong Feng. UAV-assisted Data Collection for Internet of Things: A Survey. *IEEE Internet of Things Journal*, 9(17):15460–15483, 2022.

[25] Ruofei Ma, Ruisong Wang, Gongliang Liu, Hsiao-Hwa Chen, and Zhiliang Qin. UAV-Assisted Data Collection for Ocean Monitoring Networks. *IEEE Network*, 34 (6):250–258, 2020.

[26] 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):1211–1223, 2020.

[27] Haijun Wang, Haitao Zhao, Jiao Zhang, Dongtang Ma, Jiaxun Li, and Jibo Wei. Survey on Unmanned Aerial Vehicle Networks: A Cyber Physical System Perspective. *IEEE Communications Surveys & Tutorials*, 22(2):1027–1070, 2019.

[28] Vuk Marojevic, Ismail Guvenc, Rudra Dutta, Mihail L. Sichitiu, and Brian A. Floyd. Advanced Wireless for Unmanned Aerial Systems: 5G Standardization, Research Challenges, and AERPAW Architecture. *IEEE Vehicular Technology Magazine*, 15 (2):22–30, 2020.

[29] Muhammad Morshed Alam and Sangman Moh. Joint Topology Control and Routing in a UAV Swarm for Crowd Surveillance. *Journal of Network and Computer Applications*, 204:103427, 2022.

[30] Oussama Bekkouche, Konstantinos Samdanis, Miloud Bagaa, and Tarik Taleb. A Service-based Architecture for Enabling UAV Enhanced Network Services. *IEEE Network*, 34(4):328–335, 2020.

[31] Alfonso Alcántara, Jesús Capitán, Rita Cunha, and Aníbal Ollero. Optimal Trajectory Planning for Cinematography with Multiple Unmanned Aerial Vehicles. *Robotics and Autonomous Systems*, 140:103778, 2021.

[32] Demeke Shumeye Lakew, Umar Sa'ad, Nhu-Ngoc Dao, Woongsoo Na, and Sungrae Cho. Routing in Flying Ad Hoc Networks: A Comprehensive Survey. *IEEE Communications Surveys & Tutorials*, 22(2):1071–1120, 2020.

[33] Ali H. Wheeb, Rosdiadee Nordin, Asma'Abu Samah, Mohammed H. Alsharif, and Muhammad Asghar Khan. Topology-based Routing Protocols and Mobility Models for Flying Ad Hoc Networks: A Contemporary Review and Future Research Directions. *Drones*, 6(1):9, 2021.

[34] Kai-Yun Tsao, Thomas Girdler, and Vassilios G. Vassilakis. A Survey of Cyber Security Threats and Solutions for UAV Communications and Flying Ad-Hoc Networks. *Ad Hoc Networks*, 133:102894, 2022.

[35] Pietro Boccadoro, Domenico Striccoli, and Luigi Alfredo Grieco. An extensive survey on the internet of drones. *Ad Hoc Networks*, 122:102600, 2021.

[36] Navid Ali Khan, Noor Zaman Jhanjhi, Sarfraz Nawaz Brohi, and Anand Nayyar. Emerging Use of UAV's: Secure Communication Protocol Issues and Challenges. In *Drones in Smart-cities*, pages 37–55. Elsevier, 2020.

[37] Alisson R. Svaigen, Azzedine Boukerche, Linnyer B. Ruiz, and Antonio A. F. Loureiro. Design guidelines of the internet of drones location privacy protocols. *IEEE Internet of Things Magazine*, 5(2):175–180, 2022.

[38] Alisson R. Svaigen, Azzedine Boukerche, Linnyer B. Ruiz, and Antonio A. F. Loureiro. Is the remote id a threat to the drone's location privacy on the internet of drones? In *Proceedings of the 20th ACM International Symposium on Mobility Management and Wireless Access*, pages 81–88, 2022.

[39] Lailla M. S. Bine, Azzedine Boukerche, Linnyer B. Ruiz, and Antonio A. F. Loureiro. Leveraging urban computing with the internet of drones. *IEEE Internet of Things Magazine*, 5(1):160–165, 2022.

[40] Luigi Di Puglia Pugliese, Francesca Guerriero, and Maria Grazia Scutellá. The Last-mile Delivery Process with Trucks and Drones Under Uncertain Energy Consumption. *Journal of Optimization Theory and Applications*, 191(1):31–67, 2021.

[41] Alicia Esquivel Morel, Chengyi Qu, Prasad Calyam, Cong Wang, Komal Thareja, Anirban Mandal, Eric Lyons, Michael Zink, George Papadimitriou, and Ewa Deelman. FlyNet: Drones on the Horizon. *IEEE Internet Computing*, 27(3):35–43, 2023.

[42] Soohwan Oh and Yoonjin Yoon. Urban Drone Operations: A Data-centric and Comprehensive Assessment of Urban Airspace with a Pareto-based Approach. *Transportation Research Part A: Policy and Practice*, 182:104034, 2024.

[43] Nader Samir Labib, Grégoire Danoy, Jedrzej Musial, Matthias R. Brust, and Pascal Bouvry. Internet of Unmanned Aerial Vehicles—A Multilayer Low-altitude Airspace Model for Distributed UAV Traffic Management. *Sensors*, 19(21):4779, 2019.

[44] Ravil I. Mukhamediev, Kirill Yakunin, Margulan Aubakirov, Ilyas Assanov, Yan Kuchin, Adilkhan Symagulov, Vitaly Levashenko, Elena Zaitseva, Dmitry Sokolov, and Yedilkhan Amirgaliyev. Coverage Path Planning Optimization of Heterogeneous UAVs Group for Precision Agriculture. *IEEE Access*, 2023.

[45] Sung Won Cho, Hyun Ji Park, Hanseob Lee, David Hyunchul Shim, and Sun-Young Kim. Coverage Path Planning for Multiple Unmanned Aerial Vehicles in Maritime Search and Rescue Operations. *Computers & Industrial Engineering*, 161:107612, 2021.

[46] E. Viridiana Vazquez-Carmona, Juan Irving Vasquez-Gomez, Juan Carlos Herrera-Lozada, and Mayra Antonio-Cruz. Coverage Path Planning for Spraying Drones. *CAIE*, 168:108125, 2022.

[47] Alisson R. Svaigen, Azzedine Boukerche, Linnyer B. Ruiz, and Antonio A. F. Loureiro. Trajectory matters: Impact of jamming attacks over the drone path planning on the internet of drones. *Ad Hoc Networks*, 146:103179, 2023.

[48] Alisson R. Svaigen, Azzedine Boukerche, Linnyer B. Ruiz, and Antonio A. F. Loureiro. Biomixd: A bio-inspired and traffic-aware mix zone placement strategy for location privacy on the internet of drones. *Computer Communications*, 195: 111–123, 2022.

[49] Lexu Du, Yankai Fan, Mingzhen Gui, and Dangjun Zhao. A Multi-Regional Path-Planning Method for Rescue UAVs with Priority Constraints. *Drones*, 7(12):692, 2023.

[50] Saeed Hamood Alsamhi, Alexey V. Shvetsov, Santosh Kumar, Jahan Hassan, Mohammed A. Alhartomi, Svetlana V. Shvetsova, Radhya Sahal, and Ammar Hawbani. Computing in the Sky: A Survey on Intelligent Ubiquitous Computing for UAV-assisted 6G Networks and Industry 4.0/5.0. *Drones*, 6(7):177, 2022.

[51] Daniela Renga and Michela Meo. Can High Altitude Platform Stations Make 6G Sustainable? *IEEE Communications Magazine*, 60(9):75–80, 2022.

[52] Qubeijian Wang, Hong-Ning Dai, Qiu Wang, Mahendra K. Shukla, Wei Zhang, and Carlos Guedes Soares. On Connectivity of UAV-assisted Data Acquisition for Underwater Internet of Things. *IEEE Internet of Things Journal*, 7(6):5371–5385, 2020.

[53] Mauro Tropea, Mattia Giovanni Spina, and Floriano De Rango. Supporting Dynamic IDS Deployment with Load Balancing Strategy for SDN-enabled Drones in Emergency Scenarios. In *Proceedings of the 26th MSWiM*, pages 297–300, 2023.

[54] Ning Hu, Zhihong Tian, Yanbin Sun, Lihua Yin, Baokang Zhao, Xiaojiang Du, and Nadra Guizani. Building agile and Resilient UAV Networks based on SDN and Blockchain. *IEEE Network*, 35(1):57–63, 2021.

[55] Haque Nawaz, Husnain Mansoor Ali, and Asif Ali Laghari. UAV Communication Networks Issues: A Review. *Archives of Computational Methods in Engineering*, 28 (3):1349–1369, 2021.

[56] Laith Abualigah, Ali Diabat, Putra Sumari, and Amir H. Gandomi. Applications, Deployments, and Integration of Internet of Drones (IoD): A Review. *IEEE Sensors Journal*, 21(22):25532–25546, 2021. doi: 10.1109/JSEN.2021.3114266.

[57] Dave Fraser. Thousands of drones form incredible globe flying over tokyo's stadium as olympic games begin with epic opening ceremony, 2021. URL `https://www.thesun.co.uk/sport/tokyo-olympics-2020/15671454/opening-ceremony-coronavirus-pandemic-games/`. Last accessed 31 July 2023.

[58] Fatima Zohra Bousbaa, Chaker Abdelaziz Kerrache, Zohra Mahi, Abdou El Karim Tahari, Nasreddine Lagraa, and Mohamed Bachir Yagoubi. Geouavs: A new geocast routing protocol for fleet of uavs. *Computer Communications*, 149:259–269, 2020.

[59] Muhammad Yeasir Arafat and Sangman Moh. Location-aided Delay Tolerant Routing Protocol in UAV Networks for Post-disaster Operation. *IEEE Access*, 6:59891–59906, 2018.

[60] Syed Kamran Haider, Ali Nauman, Muhammad Ali Jamshed, Aimin Jiang, Sahar Batool, and Sung Won Kim. Internet of Drones: Routing Algorithms, Techniques and Challenges. *Mathematics*, 10(9):1488, 2022.

[61] Ender Cetin, Cristina Barrado, Guillem Muñoz, Miquel Macias, and Enric Pastor. Drone Navigation and Avoidance of Obstacles Through Deep Reinforcement Learning. In *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*, pages 1–7. IEEE, 2019.

[62] Jawad Naveed Yasin, Sherif Abdelmonem Sayed Mohamed, Mohammad-Hashem Haghbayan, Jukka Heikkonen, Hannu Tenhunen, Muhammad Mehboob Yasin, and Juha Plosila. Energy-efficient Formation Morphing for Collision Avoidance in a Swarm of Drones. *IEEE Access*, 8:170681–170695, 2020.

[63] Je-Kwan Park and Tai-Myoung Chung. Boundary-RRT* Algorithm for Drone Collision Avoidance and Interleaved Path Re-planning. *Journal of Information Processing Systems*, 16(6):1324–1342, 2020.

[64] Debashisha Mishra, Anna Maria Vegni, Valeria Loscri, and Enrico Natalizio. Drone networking in the 6g era: A technology overview. *IEEE Communications Standards Magazine*, 5(4):88–95, 2021.

[65] Di Wu, Dmitri I. Arkhipov, Minyoung Kim, Carolyn L. Talcott, Amelia C. Regan, Julie A. McCann, and Nalini Venkatasubramanian. Addsen: Adaptive data processing and dissemination for drone swarms in urban sensing. *IEEE Transactions on Computers*, 66(2):183–198, 2016.

[66] 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):12092, 2021.

[67] P. Ramkumar, R. Uma, S. Usha, and R. Valarmathi. Real Time Path Planning Using Intelligent Transportation System for Hybrid VANET. In *2020 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS)*, pages 1–7. IEEE, 2020.

[68] Hamza Toulni, Mounia Miyara, Youssef Filali, and Stéphane Cédric Koumétio Tékouabou. Preventing Urban Traffic Congestion Using VANET Technology in Urban Area. In *E3S Web of Conferences*, volume 418, page 02005. EDP Sciences, 2023.

[69] Melaouene Noussaiba, Abdul Razaque, and Romadi Rahal. Heterogeneous Algorithm for Efficient-Path Detection and Congestion Avoidance for a Vehicular-Management System. *Sensors*, 23(12):5471, 2023.

[70] Md Sahabul Alam, Gunes Karabulut Kurt, Halim Yanikomeroglu, Peiying Zhu, and Ngoc Dũng DJào. High Altitude Platform Station Based Super Macro Base Station Constellations. *IEEE Communications Magazine*, 59(1):103–109, 2021.

[71] Ruifeng She and Yanfeng Ouyang. Efficiency of UAV-based Last-mile Delivery Under Congestion in Low-altitude Air. *Transportation Research Part C: Emerging Technologies*, 122:102878, 2021.

[72] Jiwon Kim, Yonghun Choi, Seunghyeok Jeon, Jaeyun Kang, and Hojung Cha. Optrone: Maximizing Performance and Energy Resources of Drone Batteries. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(11): 3931–3943, 2020.

[73] Roberto Pinto and Alexandra Lagorio. Point-to-point Drone-based Delivery Network Design with Intermediate Charging Stations. *Transportation Research Part C: Emerging Technologies*, 135:103506, 2022.

[74] Salih Safa Bacanli, Enas Elgeldawi, Begümhan Turgut, and Damla Turgut. UAV Charging Station Placement in Opportunistic Networks. *Drones*, 6(10):293, 2022.

[75] Angelo Raciti, Santi Agatino Rizzo, and Giovanni Susinni. Drone charging stations over the buildings based on a wireless power transfer system. In *IEEE Industrial and Commercial Power Systems Technical Conference (I&CPS)*, pages 1–6, 2018.

[76] Angelo Trotta, Fabio D. Andreagiovanni, Marco Di Felice, Enrico Natalizio, and Kaushik Roy Chowdhury. When UAVs Ride a Bus: Towards Energy-efficient City-scale Video Surveillance. In *Ieee infocom 2018-ieee conference on computer communications*, pages 1043–1051. IEEE, 2018.

[77] Hailong Huang and Andrey V. Savkin. Deployment of Charging Stations for Drone Delivery Assisted by Public Transportation Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):15043–15054, 2021.

[78] Shuang Wang, Changhong Zheng, and Sebastian Wandelt. Policy Challenges for Coordinated Delivery of Trucks and Drones. *Journal of the Air Transport Research Society*, page 100001, 2024.

[79] Mohammad Moshref-Javadi, Ahmad Hemmati, and Matthias Winkenbach. A Truck and Drones Model for Last-Mile Delivery: A Mathematical Model and Heuristic Approach. *Applied Mathematical Modelling*, 80:290–318, 2020.

[80] Alireza Ermagun and Nazanin Tajik. Multiple-drones-multiple-trucks Routing Problem for Disruption Assessment. *Transportation Research Record*, 2677(2):725–740, 2023.

[81] Alexander Rave, Pirmin Fontaine, and Heinrich Kuhn. Drone Location and Vehicle Fleet Planning with Trucks and Aerial Drones. *European Journal of Operational Research*, 308(1):113–130, 2023.

[82] Jiajing Gao, Lu Zhen, and Shuaian Wang. Multi-trucks-and-drones Cooperative Pickup and Delivery Problem. *Transportation Research Part C: Emerging Technologies*, 157:104407, 2023.

[83] Yassine Mekdad, Ahmet Aris, Leonardo Babun, Abdeslam El Fergougui, Mauro Conti, Riccardo Lazzeretti, and A Selcuk Uluagac. A Survey on Security and Privacy Issues of UAVs. *Computer Networks*, 224:109626, 2023.

[84] Abdelouahid Derhab, Omar Cheikhrouhou, Azza Allouch, Anis Koubaa, Basit Qureshi, Mohamed Amine Ferrag, Leandros Maglaras, and Farrukh Aslam Khan. Internet of Drones Security: Taxonomies, Open Issues, and Future Directions. *Vehicular Communications*, page 100552, 2022.

[85] Rohana Sham, Ching Sin Siau, Steven Tan, Dawn Chii Kiu, Hasminulhadi Sabhi, Hui Zhu Thew, Ganeshsree Selvachandran, Shio Gai Quek, Noorsiah Ahmad, and Mohd Hanif Mohd Ramli. Drone Usage for Medicine and Vaccine Delivery During the COVID-19 Pandemic: Attitude of Health Care Workers in Rural Medical Centres. *Drones*, 6(5):109, 2022.

[86] Jean-Paul Yaacoub, Hassan Noura, Ola Salman, and Ali Chehab. Security Analysis of Drones Systems: Attacks, Limitations, and Recommendations. *Internet of Things*, 11:100218, 2020.

[87] Xian-Chun Zheng and Hung-Min Sun. Hijacking Unmanned Aerial Vehicle by Exploiting Civil GPS Vulnerabilities Using Software-defined Radio. *Sensors & Materials*, 32, 2020.

[88] Muktar Yahuza, Mohd Yamani Idna Idris, Ismail Bin Ahmedy, Ainuddin Wahid Abdul Wahab, Tarak Nandy, Noorzaily Mohamed Noor, and Abubakar Bala. Internet of Drones Security and Privacy Issues: Taxonomy and Open Challenges. *IEEE Access*, 9:57243–57270, 2021.

[89] Diana Pamela Moya Osorio, Ijaz Ahmad, José David Vega Sánchez, Andrei Gurtov, Johan Scholliers, Matti Kutila, and Pawani Porambage. Towards 6G-enabled Internet of Vehicles: Security and Privacy. *IEEE Open Journal of the Communications Society*, 3:82–105, 2022.

[90] L. Oyuki Rojas-Perez, Roberto Munguia-Silva, and Jose Martinez-Carranza. Real-time Landing Zone Detection for UAVs Using Single Aerial Images. In *10th international micro air vehicle competition and conference, Melbourne, Australia*, pages 243–248, 2018.

[91] Oghuz Bektash, Jacob Juul Naundrup, and Anders la Cour-Harbo. Analyzing Visual Imagery for Emergency Drone Landing on Unknown Environments. *International Journal of Micro Air Vehicles*, 14:17568293221106492, 2022.

[92] Miguel Saavedra-Ruiz, Ana Maria Pinto-Vargas, and Victor Romero-Cano. Monocular Visual Autonomous Landing System for Quadcopter Drones Using Software in the Loop. *IEEE Aerospace and Electronic Systems Magazine*, 37(5):2–16, 2021.

[93] Hao Du, Wei Wang, Xuerao Wang, and Yuanda Wang. Autonomous Landing Scene Recognition Based on Transfer Learning for Drones. *Journal of systems engineering and electronics*, 34(1):28–35, 2023.

[94] Alisson R. Svaigen, Azzedine Boukerche, Linnyer B. Ruiz, and Antonio A. F. Loureiro. MixRide: An Energy-aware Location Privacy Protection Mechanism for

the Internet of Drones. In *GLOBECOM 2022-2022 IEEE Global Communications Conference*, pages 3527–3532. IEEE, 2022.

[95] Ilker Bekmezci, Ozgur Koray Sahingoz, and Şamil Temel. Flying ad-hoc networks (fanets): A survey. *Ad Hoc Networks*, 11(3):1254–1270, 2013.

[96] Armir Bujari, Claudio E. Palazzi, and Daniele Ronzani. Fanet application scenarios and mobility models. In *Proceedings of the 3rd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*, pages 43–46, New York, NY, USA, 2017. ACM.

[97] Jianguo Sun, Wenshan Wang, Liang Kou, Yun Lin, Liguo Zhang, Qingan Da, and Lei Chen. A data authentication scheme for uav ad hoc network communication. *The Journal of Supercomputing*, 76(6):4041–4056, 2020.

[98] Armir Bujari, Carlos T. Calafate, Juan-Carlos Cano, Pietro Manzoni, Claudio Enrico Palazzi, and Daniele Ronzani. Flying ad-hoc network application scenarios and mobility models. *International Journal of Distributed Sensor Networks*, 13(10): 1550147717738192, 2017.

[99] Yapu Li, Weimin Wang, Huaqiang Gao, Yongle Wu, Ming Su, Jingchao Wang, and Yuanan Liu. Air-to-ground 3d channel modeling for uav based on gauss-markov mobile model. *AEU-International Journal of Electronics and Communications*, 114: 152995, 2020.

[100] Ouns Bouachir, Alinoe Abrassart, Fabien Garcia, and Nicolas Larrieu. A mobility model for uav ad hoc network. In *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 383–388, Orlando, FL, USA, 2014. IEEE.

[101] Yongs Zeng, Qingqing Wu, and Rui Zhang. Accessing from the sky: A tutorial on uav communications for 5g and beyond. *Proceedings of the IEEE*, 107(12):2327–2375, 2019.

[102] Sungpeel Kim and Jaehoon Choi. Beam steering antenna with reconfigurable parasitic elements for fpv drone applications. *Microwave and Optical Technology Letters*, 60(9):2173–2177, 2018.

[103] Anna Vegni, Valeria Loscri, Carlos Calafate, and Pietro Manzoni. Communication technologies enabling effective uav networks: a standards perspective. *IEEE Communications Standards Magazine*, 2021.

[104] Felipe Domingos Da Cunha, Azzedine Boukerche, Leandro Villas, Aline Carneiro Viana, and Antonio A. F. Loureiro. Data communication in vanets: a survey, challenges and applications. *INRIA*, RR(8498):25, 2014.

[105] Sahil Vashisht, Sushma Jain, and Gagangeet Singh Aujla. Mac protocols for unmanned aerial vehicle ecosystems: Review and challenges. *Computer Communications*, 160:443–463, 2020.

[106] Omar Sami Oubbati, Abderrahmane Lakas, Fen Zhou, Mesut Güneş, and Mohamed Bachir Yagoubi. A survey on position-based routing protocols for flying ad hoc networks (fanets). *Vehicular Communications*, 10:29–56, 2017.

[107] Jingjing Wang, Chunxiao Jiang, Zhu Han, Yong Ren, Robert G. Maunder, and Lajos Hanzo. Taking drones to the next level: Cooperative distributed unmanned-aerial-vehicular networks for small and mini drones. *Ieee VehIcular Technology MagazIne*, 12(3):73–82, 2017.

[108] Maxim Raya, Panos Papadimitratos, and Jean-Pierre Hubaux. Securing vehicular communications. *IEEE Wireless Communications*, 13(5):8–15, 2006.

[109] Marc Bechler, Sven Jaap, and Lars Wolf. An optimized tcp for internet access of vehicular ad hoc networks. In *International Conference on Research in Networking*, pages 869–880, Berlin, Heidelberg, 2005. Springer.

[110] Rezoan Ahmed Nazib and Sangman Moh. Routing protocols for unmanned aerial vehicle-aided vehicular ad hoc networks: A survey. *IEEE Access*, 8:77535–77560, 2020.

[111] Hamid Menouar, Ismail Guvenc, Kemal Akkaya, A Selcuk Uluagac, Abdullah Kadri, and Adem Tuncer. Uav-enabled intelligent transportation systems for the smart city: Applications and challenges. *IEEE Communications Magazine*, 55(3):22–28, 2017.

[112] Shushman Choudhury, Kiril Solovey, Mykel J. Kochenderfer, and Marco Pavone. Efficient large-Scale Multi-Drone Delivery Using Transit Networks. *Journal of Artificial Intelligence Research*, 70:757–788, 2021.

[113] Qixun Zhang, Menglei Jiang, Zhiyong Feng, Wei Li, Wei Zhang, and Miao Pan. IoT Enabled UAV: Network Architecture and Routing Algorithm. *IEEE Internet of Things Journal*, 6(2):3727–3742, 2019.

[114] Dimitrios Zorbas and Christos Douligeris. Computing optimal drone positions to wirelessly recharge iot devices. In *IEEE Conference on Computer Communications Workshops*, pages 628–633, 2018.

[115] Rasheed Hussain, Jooyoung Lee, and Sherali Zeadally. Trust in vanet: A survey of current solutions and future research opportunities. *IEEE Transactions on Intelligent Transportation Systems*, 22(5):2553–2571, 2020.

[116] Yixian Zheng, Wenchao Wu, Yuanzhe Chen, Huamin Qu, and Lionel M. Ni. Visual analytics in urban computing: An overview. *IEEE Transactions on Big Data*, 2(3): 276–296, 2016.

[117] Tim Kindberg, Matthew Chalmers, and Eric Paulos. Guest editors' introduction: Urban computing. *IEEE Pervasive Computing*, 6(3):18–20, 2007.

[118] H. T. Lally, I. O'Connor, O.P. Jensen, and C.T. Graham. Can drones be used to conduct water sampling in aquatic environments? a review. *Science of the Total Environment*, 670:569–575, 2019.

[119] Qijun Gu, Drew R. Michanowicz, and Chunrong Jia. Developing a modular unmanned aerial vehicle (uav) platform for air pollution profiling. *Sensors*, 18(12): 4363, 2018.

[120] Saptarshi Ghosh, Krishnachur Ghosh, Sayan Karamakar, Shubham Prasad, Nilava Debabhuti, Prolay Sharma, Bipan Tudu, Nabarun Bhattacharyya, and Rajib Bandyopadhyay. Development of an iot based robust architecture for environmental monitoring using uav. In *2019 IEEE 16th India Council International Conference (INDICON)*, pages 1–4, Rajkot, India, 2019. IEEE.

[121] Jasmin Ćatić, Admir Mulahusić, Nedim Tuno, and Jusuf Topoljak. Using the semi-professional uav system in surveying the medium size area of complex urban surface. In *International Conference "New Technologies, Development and Applications"*, pages 853–860, Cham, 2020. Springer.

[122] Junwon Seo, Luis Duque, and Jim Wacker. Drone-enabled bridge inspection methodology and application. *Automation in Construction*, 94:112–126, 2018.

[123] Shanyue Guan, Zhen Zhu, and George Wang. A Review on UAV-based Remote Sensing Technologies for Construction and Civil Applications. *Drones*, 6(5):117, 2022.

[124] Nader Mohamed, Jameela Al-Jaroodi, Imad Jawhar, Ahmed Idries, and Farhan Mohammed. Unmanned aerial vehicles applications in future smart cities. *Technological Forecasting and Social Change*, 153(April):119293, 2018.

[125] Keunhyun Park and Reid Ewing. The usability of unmanned aerial vehicles (uavs) for measuring park-based physical activity. *Landscape and Urban Planning*, 167: 157–164, 2017.

[126] M. Koeva, M. Muneza, C. Gevaert, M. Gerke, and F. Nex. Using uavs for map creation and updating. a case study in rwanda. *Survey Review*, 50(361):312–325, 2018.

[127] Manoranjan Muthusamy, Monica Rivas Casado, Gloria Salmoral, Tracy Irvine, and Paul Leinster. A remote sensing based integrated approach to quantify the impact of fluvial and pluvial flooding in an urban catchment. *Remote Sensing*, 11(5):577, 2019.

[128] Ian L. Turner, Mitchell D. Harley, and Christopher D. Drummond. Uavs for coastal surveying. *Coastal Engineering*, 114:19–24, 2016.

[129] Martin Peter Christiansen, Morten Stigaard Laursen, Rasmus Nyholm Jørgensen, Søren Skovsen, and René Gislum. Designing and testing a uav mapping system for agricultural field surveying. *Sensors*, 17(12):2703, 2017.

[130] Sebeom Park and Yosoon Choi. Applications of unmanned aerial vehicles in mining from exploration to reclamation: A review. *Minerals*, 10(8):663, 2020.

[131] F. L. Bonali, A. Tibaldi, F. Marchese, L. Fallati, E. Russo, C. Corselli, and A. Savini. Uav-based surveying in volcano-tectonics: An example from the iceland rift. *Journal of Structural Geology*, 121:46–64, 2019.

[132] E. S. Malinverni, C. Conati Barbaro, R. Pierdicca, C. A. Bozzi, and A. N. Tassetti. Uav surveying for a complete mapping and documentation of archaeological findings. the early neolithic site of portonovo. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 41:1149, 2016.

[133] Eugen Valentin Butilă and Răzvan Gabriel Boboc. Urban Traffic Monitoring and Analysis Using Unmanned Aerial Vehicles (UAVs): A Systematic Literature Review. *Remote Sensing*, 14(3):620, 2022.

[134] Chunsun Zhang. An uav-based photogrammetric mapping system for road condition assessment. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci*, 37:627–632, 2008.

[135] David Sziroczak, Daniel Rohacs, and Jozsef Rohacs. Review of Using Small UAV Based Meteorological Measurements for Road Qeather Management. *Progress in Aerospace Sciences*, 134:100859, 2022.

[136] Pablo Garcia-Aunon, Juan Jesús Roldán, and Antonio Barrientos. Monitoring traffic in future cities with aerial swarms: Developing and optimizing a behavior-based surveillance algorithm. *Cognitive Systems Research*, 54:273–286, 2019.

[137] E. N. Barmpounakis and Nikolaos Geroliminis. Utilizing a swarm of drones for large-scale traffic measurements. In *Proc., 19th Swiss Transport Research Conference*, page ., Ascona, Switzerland, 2019. STRC.

[138] Emmanouil Barmpounakis and Nikolas Geroliminis. On the new era of urban traffic monitoring with massive drone data: The pneuma large-scale field experiment. *Transportation research part C: emerging technologies*, 111:50–71, 2020.

[139] Juan Jesús Roldán-Gómez, Pablo Garcia-Aunon, Pablo Mazariegos, and Antonio Barrientos. Swarmcity project: Monitoring traffic, pedestrians, climate, and pollution with an aerial robotic swarm. *Personal and Ubiquitous Computing*, pages 1–17, 2020.

[140] Alexander Baklanov, Luisa T. Molina, and Michael Gauss. Megacities, air quality and climate. *Atmospheric Environment*, 126:235–249, 2016.

[141] Marius Minea and Cătălin Marian Dumitrescu. Urban Traffic Noise Analysis Using UAV-Based Array of Microphones. *Sensors*, 23(4):1912, 2023.

[142] Luis Carlos Rodríguez Timaná, Diego Fernando Saavedra Lozano, María Fernanda Díaz Velásquez, and Javier Ferney Castillo García. Technical feasibility for the mobile measurement of noise pollution by remotely piloted aircraft system. In *International Conference on Applied Technologies*, pages 219–230, Cham, 2019. Springer.

[143] Natalia Sliusar, Timofey Filkin, Marion Huber-Humer, and Marco Ritzkowski. Drone Technology in Municipal Solid Waste Management and Landfilling: A Comprehensive Review. *Waste Management*, 139:1–16, 2022.

[144] Josefa Wivou, Lanka Udawatta, Ali Alshehhi, Ebrahim Alzaabi, Ahmed Albeloshi, and Saeed Alfalasi. Air quality monitoring for sustainable systems via drone based technology. In *2016 IEEE International Conference on Information and Automation for Sustainability (ICIAfS)*, pages 1–5, Galle, Sri Lanka, 2016. IEEE, IEEE.

[145] Jason D. Renwick, Levente J. Klein, and Hendrik F. Hamann. Drone-based reconstruction for 3d geospatial data processing. In *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pages 729–734, Reston, VA, USA, 2016. IEEE.

[146] T. Landolsi, A. Sagahyroon, M. Mirza, O. Aref, F. Maki, and S. Maki. Pollution monitoring system using position-aware drones with 802.11 ad-hoc networks. In *2018 IEEE Conference on Wireless Sensors (ICWiSe)*, pages 40–43, Langkawi, Malaysia, 2018. IEEE.

[147] Godall Rohi, Ejofodomi O'tega, and Godswill Ofualagba. Autonomous monitoring, analysis, and countering of air pollution using environmental drones. *Heliyon*, 6(1): e03252, 2020.

[148] Joshuah K. Stolaroff, Constantine Samaras, Emma R. O'Neill, Alia Lubers, Alexandra S. Mitchell, and Daniel Ceperley. Energy use and life cycle greenhouse gas emissions of drones for commercial package delivery. *Nature communications*, 9(1): 1–13, 2018.

[149] Andrew Boggio-Dandry and Tolga Soyata. Perpetual flight for uav drone swarms using continuous energy replenishment. In *2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pages 478–484, New York, NY, USA, 2018. IEEE.

[150] 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):13, 2019.

[151] Paul D. Mitcheson, David Boyle, George Kkelis, David Yates, Juan Arteaga Saenz, Samer Aldhaher, and Eric Yeatman. Energy-autonomous sensing systems using drones. In *2017 IEEE SENSORS*, pages 1–3, Glasgow, UK, 2017. IEEE.

[152] Weisen Shi, Haibo Zhou, Junling Li, Wenchao Xu, Ning Zhang, and Xuemin Shen. Drone assisted vehicular networks: Architecture, challenges and opportunities. *IEEE Network*, 32(3):130–137, 2018.

[153] Omar Sami Oubbati, Abderrahmane Lakas, Fen Zhou, Mesut Güneş, Nasreddine Lagraa, and Mohamed Bachir Yagoubi. Intelligent uav-assisted routing protocol for urban vanets. *Computer Communications*, 107:93–111, 2017.

[154] David Gallacher. Drone applications for environmental management in urban spaces: A review. *International Journal of Sustainable Land Use and Urban Planning*, 3(4):1–14, 2016.

[155] Syed Ahsan Raza Naqvi, Syed Ali Hassan, Haris Pervaiz, and Qiang Ni. Drone-aided communication as a key enabler for 5g and resilient public safety networks. *IEEE Communications Magazine*, 56(1):36–42, 2018.

[156] Prabhu Jyot Singh, Rohan de Silva, and Indra Seher. Comparison of communication protocols for uavs and vanets. In *2016 International Conference on Computing, Communication and Automation (ICCCA)*, pages 616–619, Greater Noida, India, 2016. IEEE.

[157] Omar Sami Oubbati, Noureddine Chaib, Abderrahmane Lakas, Salim Bitam, and Pascal Lorenz. U2rv: Uav-assisted reactive routing protocol for vanets. *International Journal of Communication Systems*, 33(10):e4104, 2020.

[158] Na Lin, Luwei Fu, Liang Zhao, Geyong Min, Ahmed Al-Dubai, and Haris Gacanin. A novel multimodal collaborative drone-assisted vanet networking model. *IEEE Transactions on Wireless Communications*, 19(7):4919–4933, 2020.

[159] M. Mageswari. Spreading joy and love to people via drones, 2020. URL `https://www.thestar.com.my/news/nation/2020/02/15/spreading-joy-and-love-to-people-via-drones`.

[160] Adarsh Kumar, Kriti Sharma, Harvinder Singh, Sagar Gupta Naugriya, Sukhpal Singh Gill, and Rajkumar Buyya. A drone-based networked system and methods for combating coronavirus disease (covid-19) pandemic. *Future Generation Computer Systems*, 115:1–19, 2021.

[161] Vinay Chamola, Vikas Hassija, Vatsal Gupta, and Mohsen 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*, 8:90225–90265, 2020.

[162] Chathura Suduwella, Akarshani Amarasinghe, Lasith Niroshan, Charith Elvitigala, Kasun De Zoysa, and Chamath Keppetiyagama. Identifying mosquito breeding sites via drone images. In *Proceedings of the 3rd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*, pages 27–30, New York, USA, 2017. Association for Computing Machinery.

[163] Sudarshan Sreeram and Lokesh Shanmugam. Autonomous robotic system based environmental assessment and dengue hot-spot identification. In *2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe)*, pages 1–6, Palermo, Italy, 2018. IEEE.

[164] M. Poljak and A. Šterbenc. Use of drones in clinical microbiology and infectious diseases: current status, challenges and barriers. *Clinical Microbiology and Infection*, 26(4):425–430, 2020.

[165] Majed Al Zayer, Sam Tregillus, Jiwan Bhandari, Dave Feil-Seifer, and Eelke Folmer. Exploring the use of a drone to guide blind runners. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 263–264, New York, NY, USA, 2016. ACM.

[166] Mauro Avila Soto, Markus Funk, Matthias Hoppe, Robin Boldt, Katrin Wolf, and Niels Henze. Dronenavigator: Using leashed and free-floating quadcopters to navigate visually impaired travelers. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 300–304, New York, NY, USA, 2017. ACM.

[167] Mauro Avila Soto and Markus Funk. Look, a guidance drone! assessing the social acceptability of companion drones for blind travelers in public spaces. In *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 417–419, New York, NY, USA, 2018. ACM.

[168] Md Tahmid Rashid, Daniel Yue Zhang, and Dong Wang. Socialdrone: An integrated social media and drone sensing system for reliable disaster response. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pages 218–227, Toronto, ON, Canada, 2020. IEEE.

[169] G. Quiroz and Si Jung Kim. A confetti drone: Exploring drone entertainment. In *2017 IEEE International Conference on Consumer Electronics (ICCE)*, pages 378–381, Las Vegas, NV, USA, 2017. IEEE.

[170] Milan Erdelj and Enrico Natalizio. Uav-assisted disaster management: Applications and open issues. In *2016 International Conference on Computing, Networking and Communications (ICNC)*, pages 1–5, Kauai, HI, USA, 2016. IEEE.

[171] Calvin Zheng, Andreina Breton, Wajeeh Iqbal, Ibaad Sadiq, Elsayed Elsayed, and Kang Li. Driving-behavior monitoring using an unmanned aircraft system (uas). In *International Conference on Digital Human Modeling and Applications in Health, Safety, Ergonomics and Risk Management*, pages 305–312, Los Angeles, CA, USA, 2015. Springer.

[172] Chris Sandbrook. The social implications of using drones for biodiversity conservation. *Ambio*, 44(4):636–647, 2015.

[173] Amanda Hodgson, Natalie Kelly, and David Peel. Unmanned aerial vehicles (uavs) for surveying marine fauna: a dugong case study. *PloS one*, 8(11):e79556, 2013.

[174] Luis Merino, Fernando Caballero, J. Ramiro Martínez-De-Dios, Iván Maza, and Aníbal Ollero. An unmanned aircraft system for automatic forest fire monitoring and measurement. *Journal of Intelligent & Robotic Systems*, 65(1-4):533–548, 2012.

[175] Gary A. Krupnick. Conservation of tropical plant biodiversity: What have we done, where are we going? *Biotropica*, 45(6):693–708, 2013.

[176] Dubravko Miljković. Methods for attenuation of unmanned aerial vehicle noise. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 0914–0919, New York, NY, USA, 2018. IEEE.

[177] James Johnson. Artificial intelligence, drone swarming and escalation risks in future warfare. *The RUSI Journal*, 165(2):26–36, 2020.

[178] Evan Ackerman and Eliza Strickland. Medical delivery drones take flight in east africa. *IEEE Spectrum*, 55(1):34–35, 2018.

[179] Michelle Sing Yee Hii, Patrick Courtney, and Paul G. Royall. An evaluation of the delivery of medicines using drones. *Drones*, 3(3):52, 2019.

[180] Brent Skorup and Connor Haaland. How drones can help fight the coronavirus. *Mercatus Center Research Paper Series, Special Edition Policy Brief (2020)*, 2020.

[181] Sri Anggraeni, Aulia Maulidina, Mauseni Wantika Dewi, Salma Rahmadianti, Yulian Putri Chandra Rizky, Zulfa Fathi Arinalhaq, Dian Usdiyana, Asep Bayu Dani Nandiyanto, Abdulkareem Sh Al-Obaidi, et al. The deployment of drones in sending drugs and patient blood samples covid-19. *Indonesian Journal of Science and Technology*, pages 18–25, 2020.

[182] Emmanuel Lamptey and Dorcas Serwaa. The use of zipline drones technology for covid-19 samples transportation in ghana. *HighTech and Innovation Journal*, 1(2): 67–71, 2020.

[183] Kavita Gupta, Sandhya Bansal, and Rajiv Goel. Uses of drones in fighting covid-19 pandemic. In *2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART)*, pages 651–655. IEEE, 2021.

[184] Marie Paul Nisingizwe, Pacifique Ndishimye, Katare Swaibu, Ladislas Nshimiyimana, Prosper Karame, Valentine Dushimiyimana, Jean Pierre Musabyimana, Clarisse Musanabaganwa, Sabin Nsanzimana, and Michael R. Law. Effect of unmanned aerial vehicle (drone) delivery on blood product delivery time and wastage in rwanda: a retrospective, cross-sectional study and time series analysis. *The Lancet Global Health*, 10(4):e564–e569, 2022.

[185] Connie A. Lin, Karishma Shah, Lt Col Cherie Mauntel, and Sachin A. Shah. Drone delivery of medications: Review of the landscape and legal considerations. *The Bulletin of the Amer. Soc. of Hospital Pharmacists*, 75(3):153–158, 2018.

[186] Andrew T. Sage, Marcelo Cypel, Mikael Cardinal, Jimmy Qiu, Atul Humar, and Shaf Keshavjee. Testing the delivery of human organ transportation with drones in the real world. *Science Robotics*, 7(73):eadf5798, 2022.

[187] James F. Campbell. Will drones revolutionize home delivery? let's get real.... *Patterns*, 3(8):100564, 2022.

[188] Megan Camponovo. Amazon begins drone deliveries in the first two cities. one is in northern california. *FOX 40*, 2022.

URL `https://fox40.com/news/local-news/san-joaquin-county/amazon-begins-drone-deliveries-lockeford-college-station/`.

[189] Jinkyung Jenny Kim, Insin Kim, and Jinsoo Hwang. A change of perceived innovativeness for contactless food delivery services using drones after the outbreak of covid-19. *International Journal of Hospitality Management*, 93:102758, 2021.

[190] Hossein Eskandaripour and Enkhsaikhan Boldsaikhan. Last-mile drone delivery: Past, present, and future. *Drones*, 7(2):77, 2023.

[191] Yixi Zeng, Kin Huat Low, Michael Schultz, and Vu N. Duong. Future demand and optimum distribution of droneports. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6. IEEE, 2020.

[192] Hong Jiang and Xinhui Ren. Comparative analysis of drones and riders in on-demand meal delivery based on prospect theory. *Discrete Dynamics in Nature and Society*, 2020:1–13, 2020.

[193] Marc P. Narkus-Kramer. Future demand and benefits for small unmanned aerial systems (uas) package delivery. In *17th AIAA Aviation Technology, Integration, and Operations Conference*, page 4103, 2017.

[194] Charlie Chen, Steve Leon, and Peter Ractham. Will customers adopt last-mile drone delivery services? an analysis of drone delivery in the emerging market economy. *Cogent Business & Management*, 9(1):2074340, 2022.

[195] Hullysses Sabino, Rodrigo V. S. Almeida, Lucas Baptista de Moraes, Walber Paschoal da Silva, Raphael Guerra, Carlos Malcher, Diego Passos, and Fernanda G. O. Passos. A systematic literature review on the main factors for public acceptance of drones. *Technology in Society*, page 102097, 2022.

[196] Idrees Waris, Rashid Ali, Anand Nayyar, Mohammed Baz, Ran Liu, and Irfan Hameed. An empirical evaluation of customers' adoption of drone food delivery services: An extended technology acceptance model. *Sustainability*, 14(5):2922, 2022.

[197] Eitan Frachtenberg. Practical drone delivery. *Computer*, 52(12):53–57, 2019.

[198] Travis B. Glick, Miguel A. Figliozzi, and Avinash Unnikrishnan. Case study of drone delivery reliability for time-sensitive medical supplies with stochastic demand and meteorological conditions. *Transportation Research Record*, 2676(1):242–255, 2022.

[199] Judy Scott and Carlton Scott. Drone delivery models for healthcare. 2017.

[200] Thomas Kirschstein. Comparison of energy demands of drone-based and ground-based parcel delivery services. *Transportation Research Part D: Transport and Environment*, 78:102209, 2020.

[201] Adrienne Welch Sudbury and E. Bruce Hutchinson. A cost analysis of amazon prime air (drone delivery). *J for Economic Educators*, 16(1):1–12, 2016.

[202] Wen-Chyuan Chiang, Yuyu Li, Jennifer Shang, and Timothy L. Urban. Impact of drone delivery on sustainability and cost: Realizing the uav potential through vehicle routing optimization. *Applied energy*, 242:1164–1175, 2019.

[203] Babar Shahzaad, Balsam Alkouz, Jermaine Janszen, and Athman Bouguettaya. Optimizing drone delivery in smart cities. *IEEE Internet Comput.*, 2023.

[204] IBGE. Estimativas da população residente no brasil e unidades da federação com data de referencia em 1º de julho de 2021. *Instituto Brasileiro de Geografia e Estatística*, 2021. URL `https://www.ibge.gov.br/estatisticas/sociais/populacao/9103-estimativas-de-populacao.html`. In portuguese.

[205] Ming Lu, Xiaohan Liao, Huanyin Yue, Yaohuan Huang, Huping Ye, Chenchen Xu, and Shifeng Huang. Optimizing distribution of droneports for emergency monitoring of flood disasters in china. *Journal of Flood Risk Management*, 13(1):e12593, 2020.

[206] Stuart Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[207] Phillip Bonacich. Power and centrality: A family of measures. *American Journal of Sociology*, 92(5):1170–1182, 1987.

[208] Tauã M. Cabreira, Lisane B. Brisolara, and Ferreira Jr Paulo R. Survey on coverage path planning with unmanned aerial vehicles. *Drones*, 3(1):4, 2019.

[209] Enric Galceran and Marc Carreras. A Survey on Coverage Path Planning for Robotics. *Robotics and Autonomous systems*, 61(12):1258–1276, 2013.

[210] Randa Almadhoun, Tarek Taha, Lakmal Seneviratne, and Yahya Zweiri. A Survey on Multi-robot Coverage Path Planning for Model Reconstruction and Mapping. *SN Applied Sciences*, 1:1–24, 2019.

[211] Abdelwahhab Bouras, Yasser Bouzid, and Mohamed Guiatni. Multi-uavs Coverage Path Planning. In *ICEECA, 17–19 December 2019, Constantine, Algeria*, pages 23–36. Springer, 2021.

[212] Julien Schleich, Athithyaa Panchapakesan, Grégoire Danoy, and Pascal Bouvry. UAV Fleet Area Coverage with Network Connectivity Constraint. In *MobiWac*, pages 131–138, 2013.

[213] Abdul Mannan, Mohammad S. Obaidat, Khalid Mahmood, Aftab Ahmad, and Rodina Ahmad. Classical Versus Reinforcement Learning Algorithms for Unmanned Aerial Vehicle Network Communication and Coverage Path Planning: A Systematic Literature Review. *International Journal of Communication Systems*, 36(5):e5423, 2023.

[214] Javier Muñoz, Blanca López, Fernando Quevedo, Concepción A. Monje, Santiago Garrido, and Luis E. Moreno. Multi UAV Coverage Path Planning in Urban Environments. *Sensors*, 21(21):7365, 2021.

[215] Rutuja Shivgan and Ziqian Dong. Energy-efficient Drone Coverage Path Planning Using Genetic Algorithm. In *HPSR*, pages 1–6. IEEE, 2020.

[216] Samira Hayat, Evşen Yanmaz, Christian Bettstetter, and Timothy X. Brown. Multi-objective Drone Path Planning for Search and Rescue with Quality-of-Service Requirements. *Autonomous Robots*, 44(7):1183–1198, 2020.

[217] Alia Ghaddar and Ahmad Merei. EAOA: Energy-aware Grid-based 3D-obstacle Avoidance in Coverage Path Planning for UAVs. *Future Internet*, 12(2):29, 2020.

[218] Mirco Theile, Harald Bayerlein, Richard Nai, David Gesbert, and Marco Caccamo. UAV Coverage Path Planning Under Varying Power Constraints Using Deep Reinforcement Learning. In *IROS*, pages 1444–1449. IEEE, 2020.

[219] Juan Irving Vasquez-Gomez, Magdalena Marciano-Melchor, Luis Valentin, and Juan Carlos Herrera-Lozada. Coverage Path Planning for 2D Convex Regions. *J. of Intelligent & Robotic Sys.*, 97:81–94, 2020.

[220] Younghoon Choi, Youngjun Choi, Simon Briceno, and Dimitri N. Mavris. Energy-constrained multi-uav coverage path planning for an aerial imagery mission using column generation. *J. of Intel. & Robotic Sys.*, 97:125–139, 2020.

[221] Yu Wu, Shaobo Wu, and Xinting Hu. Multi-constrained Cooperative Path Planning of Multiple Drones for Persistent Surveillance in Urban Environments. *Complex & Intel. Sys.*, 7:1633–1647, 2021.

[222] Aurelio G. Melo, Milena F. Pinto, Andre L. M. Marcato, Leonardo M. Honório, and Fabrício O. Coelho. Dynamic Optimization and Heuristics Based Online Coverage Path Planning in 3D Environment for UAVs. *Sensors*, 21(4):1108, 2021.

[223] Iago Z. Biundini, Milena F. Pinto, Aurelio G. Melo, Andre L. M. Marcato, Leonardo M. Honório, and Maria J. R. Aguiar. A Framework for Coverage Path Planning Optimization Based on Point Cloud for Structural Inspection. *Sensors*, 21(2):570, 2021.

[224] Konstantinos Bezas, Georgios Tsoumanis, Constantinos T. Angelis, and Konstantinos Oikonomou. Coverage Path Planning and Point-of-Interest Detection Using Autonomous Drone Swarms. *Sensors*, 22(19):7551, 2022.

[225] Jinchao Chen, Fuyuan Ling, Ying Zhang, Tao You, Yifan Liu, and Xiaoyan Du. Coverage path planning of heterogeneous unmanned aerial vehicles based on ant colony system. *Swarm and Evolutionary Computation*, 69:101005, 2022.

[226] Alisson R. Svaigen, Azzedine Boukerche, Linnyer B. Ruiz, and Antonio A. F. Loureiro. Analyzing the UAVs Traffic Flow to Enhance the Drone's Anonymization on the Internet of Drones. In *DIVANET*, pages 45–52, 2022.

[227] Dinh Dung Nguyen, Jozsef Rohacs, and Daniel Rohacs. Autonomous Flight Trajectory Control System for Drones in Smart City Traffic Management. *Int. J. of Geo-Information*, 10(5):338, 2021.

[228] Igone Morais-Quilez and Manuel Graña. Identification of Critical Subgraphs in Drone Airways Graphs by Graph Convolutional Networks. In *SOCO, Salamanca, Spain, September 5–7, 2022, Proceedings*, pages 444–453. Springer, 2022.

[229] B. Nasirian, Mehran Mehrandezh, and Farrokh Janabi-Sharifi. Efficient Coverage Path Planning for Mobile Disinfecting Robots Using Graph-based Representation of Environment. *Frontiers in Robotics and AI*, 8:624333, 2021.

[230] Richard Bormann, Florian Jordan, Joshua Hampp, and Martin Hägele. Indoor Coverage Path Planning: Survey, Implementation, Analysis. In *2018 ICRA*, pages 1718–1725. IEEE, 2018.

[231] Daniel Engelsons, Mattias Tiger, and Fredrik Heintz. Coverage Path Planning in Large-scale Multi-floor Urban Environments with Applications to Autonomous Road Sweeping. In *ICRA*, pages 3328–3334. IEEE, 2022.

[232] Tong Wang, Gangqi Dong, and Panfeng Huang. Pheromone-Diffusion-based Conscientious Reactive Path Planning for Road Network Persistent Surveillance. In *ICRA*, pages 7922–7928. IEEE, 2021.

[233] Niankai Yang, Weifan Zhang, and Wenbo Yu. Coverage Path Planning for Autonomous Road Sweepers in Obstacle-cluttered Environments. In *CCTA*, pages 491–497. IEEE, 2022.

[234] Xiaotian Fan, Yang Chen, and Yifei Shu. Multi-robot Cooperative Patrol Path Planning Method under Road Network Constraints. In *CYBER*, pages 420–425. IEEE, 2022.

[235] Hasini Viranga Abeywickrama, Beeshanga Abewardana Jayawickrama, Ying He, and Eryk Dutkiewicz. Comprehensive Energy Consumption Model for Unmanned Aerial Vehicles, Based on Empirical Studies of Battery Performance. *IEEE access*, 6:58383–58394, 2018.

[236] Carmelo Di Franco and Giorgio Buttazzo. Coverage Path Planning for UAVs Photogrammetry with Energy and Resolution Constraints. *Journal of Intelligent & Robotic Sys.*, 83:445–462, 2016.

[237] Salim Allal and Saadi Boudjit. Geocast routing protocols for vanets: Survey and guidelines. In *Sixth Int'l Conf. on Innovative Mobile and Internet Services in Ubiquitous Computing*, pages 323–328, 2012.

[238] Muhammad Fahad Khan, Kok-Lim Alvin Yau, Rafidah Md Noor, and Muhammad Ali Imran. Routing schemes in fanets: A survey. *Sensors*, 20(1):38, 2020.

[239] Abdelmalik Bachir and Abderrahim Benslimane. A multicast protocol in ad hoc networks inter-vehicle geocast. In *The 57th IEEE Semiannual Vehicular Technology Conference, 2003. VTC 2003-Spring.*, volume 4, pages 2456–2460. IEEE, 2003.

[240] Harshvardhan P. Joshi, Mihail L. Sichitiu, and Maria Kihl. Distributed robust geocast multicast routing for inter-vehicle communication. In *Proceedings of WEIRD workshop on WiMax, wireless and mobility*, volume 3, pages 27–30, 2007.

[241] Maria Kihl, Mihail Sichitiu, Ted Ekeroth, and Michael Rozenberg. Reliable geographical multicast routing in vehicular ad-hoc networks. In *International Conference on Wired/Wireless Internet Communications*, pages 315–325. Springer, 2007.

[242] Talar Atechian and Lionel Brunie. Dg-castor for query packets dissemination in vanet. In *2008 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, pages 547–552. IEEE, 2008.

[243] Hamidreza Rahbar, Kshirasagar Naik, and Amiya Nayak. Dtsg: Dynamic time-stable geocast routing in vehicular ad hoc networks. In *2010 The 9th IFIP Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, pages 1–7. IEEE, 2010.

[244] Roniel S. de Sousa, Felipe S. da Costa, Andre C. B. Soares, Luiz F. M. Vieira, and Antonio A. F. Loureiro. Geo-sdvn: A geocast protocol for software defined vehicular networks. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2018.

[245] Yuh-Shyan Chen, Yun-Wei Lin, and Sing-Ling Lee. A mobicast routing protocol in vehicular ad-hoc networks. In *GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference*, pages 1–6, 2009. doi: 10.1109/GLOCOM.2009.5426207.

[246] Luis R. Gallego-Tercero, Rolando Menchaca-Mendez, Mario E. Rivero-Angeles, and Ricardo Menchaca-Mendez. Efficient time-stable geocast routing in delay-tolerant vehicular ad-hoc networks. *IEEE Access*, 8:171034–171048, 2020.

[247] Ezgi Tetik Saglam, Yusuf Yaslan, and Sema F. Oktug. Geoakom: A smart geocasting protocol for vehicular networks. *Procedia Computer Science*, 184:364–371, 2021.

[248] Oussama Senouci, Zibouda Aliouat, and Saad Harous. Geo-sid: A new geocast safety information dissemination protocol in vanet for urban areas. In *2021 2nd International Conference on Computation, Automation and Knowledge Management (ICCAKM)*, pages 287–292. IEEE, 2021.

[249] Ganbayar Gankhuyag, Anish Prasad Shrestha, and Sang-Jo Yoo. Robust and reliable predictive routing strategy for flying ad-hoc networks. *IEEE Access*, 5:643–654, 2017.

[250] Jinfang Jiang and Guangjie Han. Routing protocols for unmanned aerial vehicles. *IEEE Communications Magazine*, 56(1):58–63, 2018.

[251] Sung-Chan Choi, Hassen Redwan Hussen, Jong-Hong Park, and Jaeho Kim. Geolocation-based routing protocol for flying ad hoc networks (fanets). In *2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 50–52. IEEE, 2018.

[252] Mehbub Alam, Nurzaman Ahmed, Rakesh Matam, and Ferdous A Barbhuiya. IEEE 802.11 ah-Enabled Internet of Drone Architecture. *IEEE Internet of Things Magazine*, 5(1):174–178, 2022.

[253] Wenfeng Xia, Yonggang Wen, Chuan Heng Foh, Dusit Niyato, and Haiyong Xie. A survey on software-defined networking. *IEEE Communications Surveys & Tutorials*, 17(1):27–51, 2014.

[254] James McCoy and Danda B. Rawat. Software-defined networking for unmanned aerial vehicular networking and security: A survey. *Electronics*, 8(12):1468, 2019.

[255] Kamal Benzekki, Abdeslam El Fergougui, and Abdelbaki Elbelrhiti Elalaoui. Software-defined networking (sdn): a survey. *Security and Communication Networks*, 9(18):5803–5833, 2016.

[256] Eirini Kalogeiton and Torsten Braun. On the impact of sdn for transmission power adaptation and fib population in ndn-vanets. In *MSWiM*, pages 57–66, 2020.

[257] Glena Aziz Qadir, Shavan Askar, et al. Software defined network based vanet. *IJSAB*, 5(3):83–91, 2021.

[258] Christophe Guerber, Nicolas Larrieu, and Mickal Royer. Software defined network based architecture to improve security in a swarm of drones. In *ICUAS*, pages 51–60. IEEE, 2019.

[259] Zhongliang Zhao, Pedro Cumino, Arnaldo Souza, Denis Rosario, Torsten Braun, Eduardo Cerqueira, and Mario Gerla. Software-defined uav networking for video dissemination services. *Ad Hoc Networks*, 83:68–77, 2019.

[260] Adarsh Kumar, Rajalakshmi Krishnamurthi, Anand Nayyar, Ashish Kr Luhach, Mohammad S. Khan, and Anuraj Singh. A novel software-defined drone network (sddn)-based collision avoidance strategies for on-road traffic monitoring and management. *Vehicular Communications*, 28:100313, 2021.

[261] Charles E. Perkins and Elizabeth M. Royer. Ad-hoc on-demand distance vector routing. In *WMCSA'99*, pages 90–100. IEEE, 1999.

[262] Moataz Samir, Dariush Ebrahimi, Chadi Assi, Sanaa Sharafeddine, and Ali Ghrayeb. Leveraging uavs for coverage in cell-free vehicular networks: A deep reinforcement learning approach. *IEEE Transactions on Mobile Computing*, 20(9): 2835–2847, 2020.

[263] Hailong Huang, Andrey V. Savkin, and Chao Huang. Control of a Novel Parcel Delivery System Consisting of a UAV and a Public Train. In *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, volume 1, pages 1047–1050. IEEE, 2019.

[264] Hailong Huang, Andrey V. Savkin, and Chao Huang. When Drones Take Public Transport: Towards Low Cost and Large Range Parcel Delivery. In *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, volume 1, pages 1657–1660. IEEE, 2019.

[265] Yan Pan, Shining Li, Qianwu Chen, Nan Zhang, Tao Cheng, Zhigang Li, Bin Guo, Qingye Han, and Ting Zhu. Efficient Schedule of Energy-Constrained UAV Using Crowdsourced Buses in Last-Mile Parcel Delivery. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(1):1–23, 2021.

[266] Weiwei Qi, Lianjie Ruan, Yue Zhi, and Bin Shen. Risk Area Identification Model of Bus Bay Stops Based on Distribution of Conflicts. *Discrete Dynamics in Nature and Society*, 2021, 2021.

[267] Yong Jin, Jia Xu, Sixu Wu, Lijie Xu, Dejun Yang, and Kaijian Xia. Bus Network Assisted Drone Scheduling for Sustainable Charging of Wireless Rechargeable Sensor Network. *Journal of Systems Architecture*, 116:102059, 2021.

[268] Nahid Parvaresh and Burak Kantarci. Deep q-learning-enabled deployment of aerial base stations in the presence of mobile users. In *Proceedings of the 20th ACM International Symposium on Mobility Management and Wireless Access*, pages 73–80, 2022.

[269] Zhaoyang Du, Celimuge Wu, Tsutomu Yoshinaga, Xianfu Chen, Xiaoyan Wang, Kok-Lim Alvin Yau, and Yusheng Ji. A Routing Protocol for UAV-Assisted Vehicular Delay Tolerant Networks. *IEEE Open Journal of the Computer Society*, 2: 85–98, 2021.

[270] Youcef Azzoug and Abdelmadjid Boukra. Enhanced UAV-Aided Vehicular Delay Tolerant Network (VDTN) Routing for Urban Environment Using a Bio-Inspired Approach. *Ad Hoc Networks*, page 102902, 2022.

[271] 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*, 9:115466–115487, 2021.

[272] Saeed H. Alsamhi, Fatemeh Afghah, Radhya Sahal, Ammar Hawbani, AA Alqaness, Brian Lee, and Mohsen Guizani. Green Internet of Things Using UAVs in B5G Networks: A Review of Applications and Strategies. *Ad Hoc Networks*, page 102505, 2021.

[273] Osama M. Bushnaq, Anas Chaaban, and Tareq Y. Al-Naffouri. The Role of UAV-IoT Networks in Future Wildfire Detection. *IEEE Internet of Things Journal*, 2021.

[274] Chaker Abdelaziz Kerrache, Abderrahmane Lakas, Nasreddine Lagraa, and Ezedin Barka. UAV-Assisted Technique for the Detection of Malicious and Selfish Nodes in VANETs. *Vehicular Communications*, 11:1–11, 2018.

[275] Karsten Heimann, Benjamin Sliwa, Manuel Patchou, and Christian Wietfeld. Modeling and Simulation of Reconfigurable Intelligent Surfaces for Hybrid Aerial and Ground-based Vehicular Communications. In *Proceedings of the 24th International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 67–74, 2021.

[276] Dharti Patel, Zunnun Narmawala, Sudeep Tanwar, and Pradeep Kumar Singh. A systematic review on scheduling public transport using iot as tool. *Smart innovations in communication and computational sciences*, pages 39–48, 2019.

[277] Clayson Celes, Azzedine Boukerche, and Antonio A. F. Loureiro. Mobility data assessment for vehicular networks. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2019.

[278] Yongpeng Shi, Jiajia Liu, Zubair Md Fadlullah, and Nei Kato. Cross-layer data delivery in satellite-aerial-terrestrial communication. *IEEE Wireless Communications*, 25(3):138–143, 2018.

[279] Gaofeng Pan and Mohamed-Slim Alouini. Flying car transportation system: Advances, techniques, and challenges. *IEEE Access*, 9:24586–24603, 2021.

[280] Gunes Karabulut Kurt, Mohammad G. Khoshkholgh, Safwan Alfattani, Ahmed Ibrahim, Tasneem S. J. Darwish, Md Sahabul Alam, Halim Yanikomeroglu, and Abbas Yongacoglu. A vision and framework for the high altitude platform station (haps) networks of the future. *IEEE Communications Surveys & Tutorials*, 23(2): 729–779, 2021.

[281] Hamed Hellaoui, Oussama Bekkouche, Miloud Bagaa, and Tarik Taleb. Aerial control system for spectrum efficiency in uav-to-cellular communications. *IEEE Communications Magazine*, 56(10):108–113, 2018.

[282] Ankita Srivastava, Arun Prakash, and Rajeev Tripathi. Location based routing protocols in vanet: Issues and existing solutions. *Vehicular Communications*, 23: 100231, 2020.

[283] Omar Sami Oubbati, Abderrahmane Lakas, Nasreddine Lagraa, and Mohamed Bachir Yagoubi. Uvar: An intersection uav-assisted vanet routing protocol. In *2016 ieee wireless communications and networking conference*, pages 1–6. IEEE, 2016.

[284] Rodolfo W. L. Coutinho, Azzedine Boukerche, Luiz F. M. Vieira, and Antonio A. F. Loureiro. Geographic and opportunistic routing for underwater sensor networks. *IEEE Transactions on Computers*, 65(2):548–561, 2015.