UNIVERSITÀ DEGLI STUDI DI PADOVA

Department of Mathematics

Ph.D Course in Brain, Mind and Computer Science

Computer Science and Innovation for Societal Challenges

XXXI cycle

Data Gathering and Dissemination over Flying Ad-hoc Networks in Smart Environments

Supervisor:
Prof. Claudio E. Palazzi

Co-Supervisor:
Prof. Andrea Facoetti

Ph.D. Student
Daniele Ronzani

September 2018

To my Family and my Friends.

Acknowledgements

This is undoubtedly one of the most great experience I've ever done. These three years of Ph.D. course have enriched me both personally and professionally.

Foremost, I want to express my sincere gratitude to my advisor Prof. Claudio Enrico Palazzi for the continuous support of my Ph.D. study and research, for his patience, motivation, and knowledge. He guided me to greater awareness and experience.

A special thanks to my colleague and friend Armir Bujari, for his sustain and advices during my Ph.D. journey.

My sincere thanks goes to the Universitat Politècnica de València, in particular to Prof. Carlos Travares Calafate and Prof. Pietro Manzoni, for allowing me to spend three wonderful months in their laboratory.

I thank my fellow lab and Ph.D. colleagues at the University of Padua for the stimulating and funny discussions we have had in the last three years.

I would like to thanks my family and my friends for all their support and sustain.

Abstract

The advent of the Internet of Things (IoT) has laid the foundations for new possibilities in our life. The ability to communicate with any electronic device has become a fascinating opportunity. Particularly interesting are UAVs (Unmanned Airborne Vehicles), autonomous or remotely controlled flying devices able to operate in many contexts thanks to their mobility, sensors, and communication capabilities. Recently, the use of UAVs has become an important asset in many critical and common scenarios; thereby, various research groups have started to consider UAVs' potentiality applied on smart environments. UAVs can communicate with each other forming a Flying Ad-hoc Network (FANET), in order to provide complex services that requires the coordination of several UAVs; yet, this also generates challenging communication issues. This dissertation starts from this standpoint, firstly focusing on networking issues and potential solutions already present in the state-of-the-art. To this aim, the peculiar issues of routing in mobile, 3D shaped ad-hoc networks have been investigated through a set of simulations to compare different ad-hoc routing protocols and understand their limits. From this knowledge, our work takes into consideration the differences between classic MANETs and FANETs, highlighting the specific communication performance of UAVs and their specific mobility models. Based on these assumptions, we propose refinements and improvements of routing protocols, as well as their linkage with actual UAV-based applications to support smart services. Particular consideration is devoted to Delay/Disruption Tolerant Networks (DTNs), characterized by long packet delays and intermittent connectivity, a critical aspect when UAVs are involved. The goal is to leverage on context-aware strategies in order to design more efficient routing solutions. The outcome of this work includes the design and analysis of new routing protocols supporting efficient UAVs' communication with heterogeneous smart objects in smart environments. Finally, we discuss about how the presence of UAV swarms may affect the perception of population, providing a critical analysis of how the consideration of these aspects could change a FANET communication infrastructure.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The exponential growth of communication technology led to the possibility to deploy any sort of ICT application in our life. Only a few years ago we just communicated vocally via wire; then the Internet was created, allowing the transmission of bits. Some time later, the wireless revolution led the possibility to create mobile devices that can be remotely controlled and can communicate with other devices without a fixed infrastructure or centralized administration. The result is called *Mobile Ad-hoc Network* (MANET), an autonomous, distributed and self-configuring network comprised of mobile wireless nodes [1, 2].

Among mobile nodes, particularly interesting are the devices named *Unmanned Aerial/Airborne Vehicles* (UAVs), also called micro aerial vehicles or drones [3]. UAVs are autonomous and flying devices able to operate in many diverse contexts thanks to their mobility, possibility to equip sensors, and communication capabilities. Recently, the use of UAVs has become an important opportunity in many critical and common application scenarios, and their ability to exchange information and cooperate each other has strengthened their utility. UAVs are easy to deploy, flexible and cover many areas. The arrival of *Global Positioning System* (GPS) technology and other localization systems permitted UAVs to be independent, without any type of human control. Examples of UAV are illustrated in Fig. 1.1.

Thereby, various research groups have started to consider UAVs' potentiality applied on *Smart Environments* (SE). The SE paradigm is designed to provide infrastructures and services in an efficient way at low costs, utilizing Information and Communication Technology (ICT) solutions. SE is a part of the *Internet of Things* (IoT) paradigm [4, 5, 6] which laid the foundations for new possibilities and its ability to communicate with any electronic device has become a fascinating opportunity. This implies that such a SE requires huge integration of ICT, including any sort of mobile node equipped with sensors and communication capability. The main example of the SE paradigm is embodied in *smart cities* [7, 8]. With the advent IoT and the Cloud platforms, smart cities are becoming an important asset to facilitate and improve everyone's life. Smart cities can extend into many domains: transportation, health, tourism, home energy management, safety and security, as illustrated in Fig. 1.2. The UAV usage may provide a network infrastructure able to support monitoring, analysis, and other operations. This added value must deal with the law regulations and the fact that in many cities in Europe and Asia UAVs are almost banned. There are numerous motivations for these restrictions, but with appropriate collaboration between institutional departments, the UAVs adoption for public applications could become reality.

(a) (b) (c)

Figure 1.1: Examples of drones. a) DJI Mavic Pro, b) DJI Inspire1, c) DJI Pro M600.

Figure 1.2: The main domains of a smart city.

However, the mobile nature of UAVs and their effective employment in the context of smart cities and IoT passes through some tough technical challenges. Indeed, to guarantee a certain level of efficacy and efficiency, UAV systems must satisfy some requirements in terms of operation (integration of a middleware, flying precision, high resolution imaging and measurements), security (fail-safe system), intelligence (data compression algorithms, automated controls), and communication (wireless links, control traffic system, ad-hoc routing protocols).

UAVs can communicate with each other forming a *Flying Ad-hoc Network* (FANET) [9, 10, 11], an evolution of MANETs adapted for highly mobile flying devices. Recently, many application scenarios relied on swarm of UAVs to perform any distributed task. If UAVs have the capability to communicate with each other, they can collaborate for a specific purpose. In Fig. 1.3, a graphic representation of a FANET is drawn.

Figure 1.3: A FANET may be connected with heterogeneous devices, vehicles and buildings.

Considering these aspects, such UAV swarms fulfill the principles of a distributed processing system [12], in which UAVs are the nodes of an entire system, which is managed through internal mechanisms and ad-hoc communication among nodes.

Such cooperation could be very efficient for diverse applications, for example the Search and Rescue scenario [13]. Imagine a group of UAVs helping rescue teams to locate sensible targets (e.g., missing people, buried victims) that explores a large part of the considered area. Those UAVs are equipped with cameras, sensors and communication apparatus, in order to perform their task and communicate between them and with the team base. They could also provide a substitute network to the team, which relies on UAVs themselves to communicate. Also, the mission coverage is clearly enlarged, due to the UAVs distribution on the area. This is just a simple example; yet, it is effective to depict the potentiality of UAV swarm usage.

A FANET could be also used to help rural environments, where many operations are time consuming and labor-expensive. The so called *smart farm* represents the application of ICT into agriculture. A FANET may be a potential solution in this context, providing a backbone network that offers connectivity to the diverse devices (sensors, actuators, databases, etc.), or be an active part in the agricultural process [14]. An example of such scenario is depicted in Fig. 1.4.

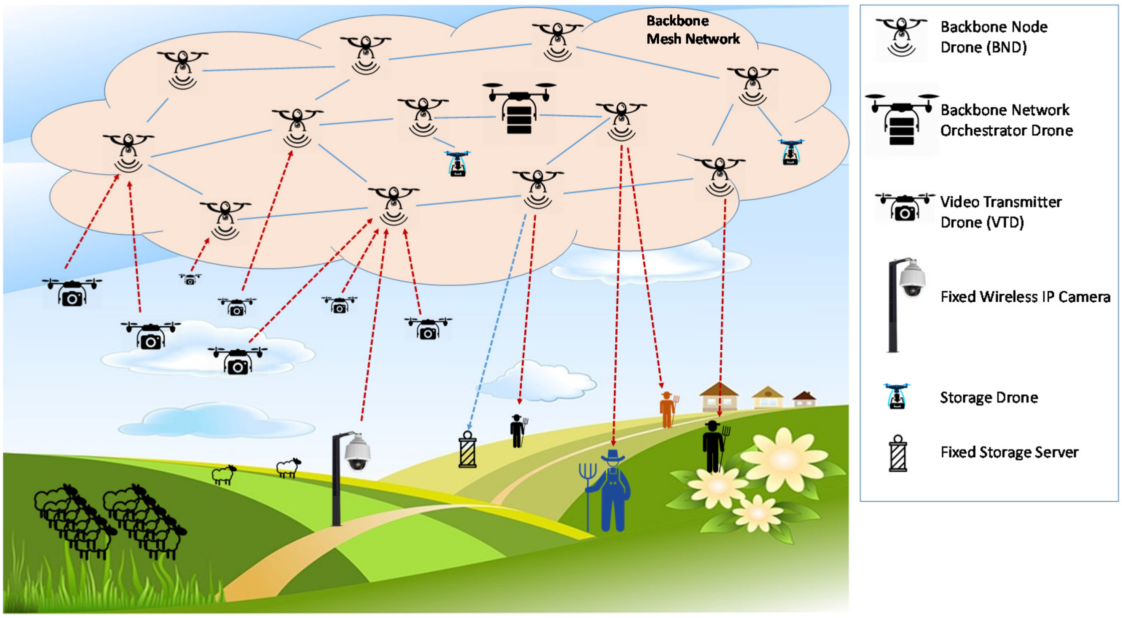


Figure 1.4: An illustration of a possible FANET in the context of a smart farm [15].

A FANET includes some important communication challenges, one of which is routing. Routing in FANETs is problematic, considering the different network characteristics, such as mobility, node density, node speed, topology. Topology, in particular, marks the difference between MANET and FANET, as a FANET is more likely to assume a three-dimensional (3D) topology.

This dissertation starts from this point, firstly focusing on a study about networking issues and potential solutions already present in the state-of-the-art related to FANETs. To this aim, the peculiar issues of routing in mobile, 3D shaped ad-hoc networks are investigated through a set of simulations to compare different ad-hoc routing protocols and understand their limits. Particular attention is paid to those protocols that rely on node location information to process the packet forwarding decision. Location information is a significant resource to find the correct route to a destination (like a navigator) and it is easy to know and communicate. Hence, we intend to exploit the advantages of position-based protocols that could be a more adequate strategies in FANETs.

From this knowledge, the project takes into consideration the differences between classic MANETs and FANETs, highlighting the specific communication performance of UAVs and the specific mobility models of UAVs. Based on these assumptions, we design refinements and improvements of routing protocols, as well as their linkage with actual drone-based applications, to support smart services. The goal is to leverage on this knowledge and consider a realistic scenario also involving heterogeneous nodes with communication capabilities present in smart city IoT (drones, vehicles, sensors, etc.).

Realistic scenarios could also include cases in which UAVs could not be always connected with each other or with the base station, introducing huge delays in transmissions. This characteristic is present in *Delay/Disruption Tolerant Networks* (DTNs), where UAVs store the data packets until they have the possibility to forward them. The dissertation also explores this aspect, analyzing the challenges of existing DTN routing protocols and designing a new solution based on positional information of UAVs.

If we look at a FANET from application point of view, they could be adequate for numerous contexts. On the other hand, a misuse of UAVs may lead to a negative perception to the population, so that the deployment of a certain UAV-system application into a city is actually quite difficult: some people feel scared or are uncomfortable in the presence of UAVs for many reasons, like security and privacy. A little part of our work provides a critical illustration of the possible aspects of UAVs negative impact to the people and how a FANET could be affected by this.

1.1 Contributions

We now introduce the overall contribution of our work, focusing on the design and implementation of our solutions and application scenario involved. Summarizing, the main contributions of this dissertation are as follows:

- Analysis and experimental comparison of considered position-based routing protocols for FANETs. We focus on the state-of-the-art, stateless geographic packet routing protocols conceived and adapted for three-dimensional network scenarios, evaluating their performance over a common scenario through a comparative analysis.

- Analysis and experimental comparison of topology-based and position-based routing protocols on a smart IoV scenario (vehicles and drones). We asses the feasibility of current routing proposal for IoV, comparing classic solutions for MANETs and solutions that use geographic position information.

- Study and analysis of FANETs in several application scenarios, considering the issues and the adequate mobility models. We list the existing mobility models and provide guidance to understand whether they could be actually adopted depending on the specific FANET application scenarios, while discussing their advantages and disadvantages.

- Design and implementation of a new routing hybrid solution: Greedy-AODV (G-CR). we propose a hybrid position-topology based algorithm that tries to strike a balance between a greedy and topology-based forwarding process in a 3D environment. G-CR greedily routes data packet toward a destination by exploiting nodes coordinates and whenever a local minima is found shifts to a topology, reactive-based approach to recover the delivery process. We present some simulations and discuss the benefits of the approach.

- Design and implementation of a new routing solution for DTNs considering geographic location and future UAV planned path. We propose a delay-tolerant routing protocol for Airborne Networks in Search and Rescue scenarios. This protocol exploits the geographical information of UAVs to make more appropriate packet forwarding decisions.

- Critical analysis of UAVs impact on people and how relative restrictions and preferences may affects UAVs behavior and network topology. Our aim is to study the key aspects of people negative perceptions and how would change FANET performance if some countermeasures were made to satisfy the population.

Chapter 2

Background

This chapter introduces the background of FANETs and their routing issues. In particular, Sec. 2.1 gives an overview of FANETs, their main differences with classic MANETs and their application scenarios. In Sec. 2.2 an introduction of the IoT context is provided, including Smart Cities and IoV scenarios. The routing problem is discussed in Sec. 2.3, introducing the position-based routing protocols and critically analyzing their benefits and how they may improve routing compared to topology-based routing protocols. Finally, in Sec. 2.4 we briefly introduce the DTN paradigm and its relevant research work.

2.1 Flying Ad-hoc Networks

Mobile Ad-hoc Networks (MANETs) have received a lot of attention from practitioners and researchers due to their inherently autonomous and distributed nature, capable of sustaining a decentralized service model without strict infrastructure reliance [1, 2]. Their flexibility makes them suitable to a wide range of operational scenarios such as rescue, disaster or hard to reach environments, military, underwater networks etc. The main characteristic of the MANET paradigm, differentiating it from traditional wired networks, is the potential of nodes to move around by following (un)planned trajectories. MANET evolutions have considered vehicles (Vehicular Ad-hoc Networks - VANETs) [16, 17], sensors (Wireless Sensor Networks - WSNs) [18, 19] and other devices, like flying ones.

More recently, the advent of new technologies and the growing capability in the production chain to miniaturize complex electronic systems has produced a wide range of gadgets. Some of them are nowadays capable to move and fly autonomously according to its preprogrammed route or remotely controlled by a user. These devices are commonly referred to as unmanned airborne vehicles (UAVs) [20], or even micro-aerial vehicles (MAVs), and can generally have quadcopter, helicopter or swinglet designs. Already many years ago, UAVs were built, especially for war scenarios. Nowadays, given the rapidly growing small unmanned aircraft industry and their cost reduction, their trade has been extended also in civil applications, such as policing and firefighting, and non-military security work, but often preferred in tactical and battlefield scenarios operated without a human presence, dirty or dangerous for manned aircraft. Interesting civil applications focuses on Search and Rescue missions, in which small-scale UAVs can be equipped with imaging and video sensors to transmit real-time audio/video streaming supporting rescue people [21].

There is a slight and still not clear difference between the terms "UAV" and "drone": a *drone* is generally defined as any aircraft that does not have a pilot aboard, whether it is operated by software or by a remote user. The term *UAV* generally refers to any aircraft operated without a pilot that can be reused. This also makes them drones, as they do not have onboard pilots. People uses the word "drone" to refer to both hobby quadcopters and military aircraft, but if someone says "UAV" they're probably referring to an aircraft used for applications that do not include hobby. For sake of clarity, in this thesis we will always use the term UAV to refer to any flying vehicle without a pilot in literature.

The use of multiple UAVs rather than one UAV plays an important role, since the limited fight time of such UAVs and the fact that some missions are time critical. Authors in [22] propose the problem of deploying a high number of low-cost, low-complexity robots inside a known environment with the objective that at least one robotic platform reaches each of n preassigned goal locations. In [9], an ambitious proposed project tries to design and build a control system of heterogeneous multi-agents, working in cooperation to solve distributed tasks that require different technical, physical and cognitive skills. This new scenario requires that those agents could communicate with each other, to exchange relevant information. These contexts have also introduced the need for a network that extends beyond the two dimensions, since the altitude is regarded as the third dimension.

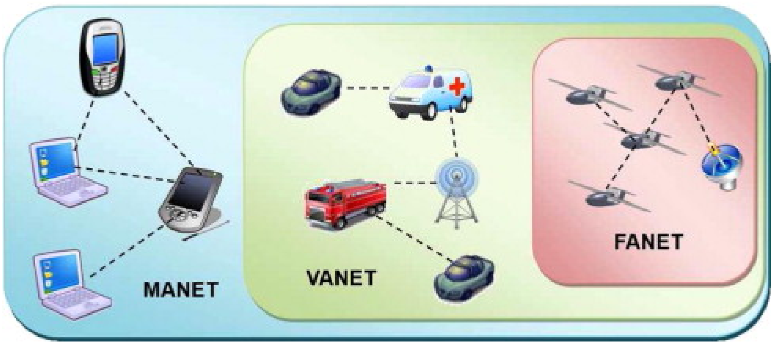


Figure 2.1: MANET, VANET and FANET.

All these aspects have paved the way to new and innovative application scenarios, defining a new kind of networking paradigm that extends the simple MANET, called *Flying Ad-hoc Network* (FANET) [9, 10, 11]. More precisely, a FANET could be defined as a subclass of VANETS, in turn included in the MANET class (see Fig. 2.1). Other different names used to refer this kind of network are [23]:

- Drone Ad-hoc Network (DANET);
- UAV network or Multi-UAV system;
- Unmanned Aerial Vehicles Ad hoc Network (UAANET);
- Airborne network;
- Networked Aerial Robots;
- Aerial Communication Network;
- Unmanned Aerial System (UAS).

A FANET differs from traditional MANETs in terms of degree of mobility, connectivity, applications areas etc. UAV connectivity is one of the major issue, because of the peculiar characteristic of flying device communication apparatus and physic motion.

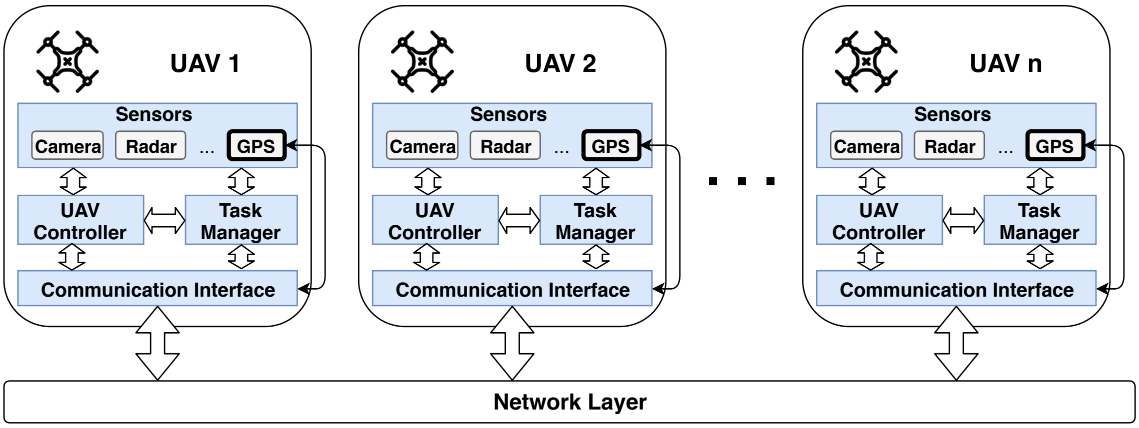


Figure 2.2: A FANET system with UAVs' and communication adopted modules [23].

In Fig. 2.2, a FANET system is illustrated. Each UAV that wants to be part of a multi-UAV system needs some required hardware and a well-defined networking model. The network layer is the fundamental part of that system. Sensors, Controller and Task Manager are other modules required to make the system working. In our work the use of GPS is crucial for its relevance in the position-base routing protocols; for this reason it is highlighted and directly connected to the Communication Interface.

The main motivations to use a multi-UAV system can be summarized in three points [10]:

1. **Extend the area coverage for a mission.** With an infrastructure-based UAV system, the operation area is limited to the communication coverage of the infrastructure. If a UAV cannot communicate with the infrastructure, it cannot operate. A FANET is based on ad-hoc communication, where UAVs can communicate with each other, extending the coverage.

2. **Bypass several obstacles.** Generally, a swarm of UAVs operates in dynamic environments, where conditions may change during the mission. This changes (e.g., weather condition) may introduce several obstacles that an infrastructure-based UAV system cannot bypass, due to the direct link between the UAV and the base station. A FANET can maintain connectivity through other UAVs, which assume specific positions to bypass the obstacle.

3. **Reduction of mission completion time.** A multi-UAV system may obtain a potential performance increase in terms of task completion time, compared to a single UAV. Their ability to communicate and cooperate can be exploited to complete complex tasks.

2.1.1 Differences between MANET and FANET

A FANET can be defined as a specialized form of MANET comprised of UAVs, differing from traditional MANETs in several key aspects [10]. In this respect, a FANET generalizes the topology from a two-dimensional (2D) one to a three-dimensional (3D) one. This trend is attracting researchers and practitioners, as well as becoming increasingly present in real life applications. In the following, the differences are detailed.

Mobility

UAVs are relatively faster than a typical MANET node. Mobility causes network partitioning, which causes links outages and changes in link quality. The movement in MANETs is generally constrained by ground artifacts while in FANETs nodes can potentially move freely in the sky. This additional degree of freedom in FANETs demands for efficient routing algorithms capable of counteracting mobility effects while preserving resources. An important consequence of mobility is the inter-UAV collision, which could prove to be very critical in some context specific missions considering self-driving UAVs. The problem could be tackled with appropriate cooperative protocols that, for example, allow the UAVs to communicate their position, speed and direction to avoid collisions.

Topology

A FANET generalizes the topology from a 2D topology to a 3D one with a certain movement scheme. Indeed, the capability of drones to fly generates a scenario where nodes are not just distributed on a plain surface, but they can assume different altitudes. Since the current commercial popularity of drones, this is the first time we can envision a real practical employment for such 3D MANETs. On the other hand, this is a very interesting and challenging scenario, especially when considering the packet routing process (from here on simply referred to as routing).

Radio Propagation

Radio signals are affected by the vicinity to the terrain. MANET nodes are very close to the ground, reducing radio propagation reliability, as in many cases there might be no line-of-sight between a source and a destination. Instead, UAVs can reach certain altitudes, reducing or even eliminating the presence of terrain artifacts, thus ensuring the line-of-sight. However, it has been evidenced that 802.11 standard performs poorly in aerial scenarios [24]. This is partially caused by channel interferences and by the characteristic of the network dynamics of UAV-to-UAV and UAV-to-ground links, which leads to restrictions on network connectivity. In fact, the radiation pattern of antennas is not a sphere, but a torus-shape, with difficult communication among UAVs placed one above the other.

Power Consumption

The nodes comprising the system are battery-powered devices, hence energy management is a key issue impacting network lifetime. Furthermore, differently from classic MANETs, nodes in a FANET have an additional power consumption, due to the actuating propellers/rotors. Hence, there is a demand for proficient protocols, increasing the network lifetime while guaranteeing system operations.

Localization

Localization of devices in an ad-hoc network is achieved by several existing localization methods, such as GPS, beaconing with anchor nodes and proximity-based techniques. Typically, MANETs use GPS to detect coordinate devices, obtaining an accuracy of 10-15 m, but the signal strength of GPS satellites could be not enough in some MANET scenarios (indoor, underwater, urban, etc.). However, in FANETs, because of their nature, we can rely on a good GPS signal.

2.1.2 Application Scenarios

In this section, we explore and categorize possible application scenarios, in which a potential FANET could be deployed to provide communication support. This description is needed since in Sec. 4.4, we describe the requirements of these application scenarios in terms of mobility, and analyze the most feasible mobility models for any scenario.

Search and Rescue Operations

In the search and rescue missions, UAVs are looking/sensing for a target, typically on the ground. Typical contexts are post catastrophic events (earthquake, hurricane, tsunami, etc.), where victims in dangerous or inaccessible areas are rescued. A FANET could be deployed to speed up the search operations providing, among other things, an autonomous and distributed coordination system. In this context, the area of interest could be divided into regions, and the system adopts a *divide et impera* approach speeding up the operations. This distributed UAV platform is capable of autonomous coordination through information exchange. UAV networks were used for search and rescue missions for the first time during Hurricane Katrina in 2005, and later in the 2011 Fukushima disaster, and in the April 2015 Nepal earthquake [13]. In [25], the authors propose Krypto, a system which deploys a group of UAVs on the disaster area, inaccessible or dangerous for the rescue team, detecting wireless signals from victims' mobile phones to locate possible survivors. In [21], a modular architecture of an autonomous UAV system for search and rescue missions is proposed and evaluated via a test in an outdoor mission. Several search strategies are explored in [26], using UAVs to minimize the time to find some victims. These strategies are discussed considering important factors that have an impact on them, such as sensors' quality, energy limitation, the presence of obstacles and communication between UAVs.

Relaying Network

Autonomously operated UAVs are being used as airborne communication relays to efficiently and securely transmit the information collected by ground devices to distant control centres, for example the delivery of data produced by WSN nodes on the ground to the user [27]. UAVs can be also used for increasing the communications range of ground relaying nodes [28], including IoV scenarios. In disaster contexts, during or after a catastrophic event, the traditional network infrastructure may suffer damages. A FANET could be deployed to restore or provide a self-sufficient communication network in isolated areas [21, 29, 30, 31, 32].

Forest Fire Detection

Forest fire detection applications cover all those cases where the monitoring of heat and fire risk is needed to prevent any type of disaster. Wooded areas are primarily affected by this type of danger. In [33], a multi-UAV cooperative perception system for the monitoring and measuring of forest fire is proposed. In [34], authors propose a fault-tolerant cooperative control (FTCC) strategy for cooperative UAVs, explaining the mobility pattern of UAVs as a certain shape that keeps the desired formation during the monitoring.

Traffic and Urban Monitoring

Roadway traffic monitoring is also a potential application in which FANETs can replace intensive labour and complicated observational infrastructure. Traditionally, these operations are achieved by fixed cameras or specialized mobile vehicles. However, UAVs seem to be a perfect match for the context, enabling information exchange amongst UAVs from/to infrastructure to have an accurate picture about urban conditions in a specific area. UAVs can detect and report traffic crashes easily. Similar, UAVs could be equipped with cameras and a variety of sensors, providing a bird's eye view for the ground rescue teams in that region(s), and can capture in real time visuals of different security situations and scenarios in both road and train networks. Monitoring urban areas can be also performed by military assets, moving on the context of military operations in urban terrain (MOUT) [35]. In [36], a multi-UAV system was designed to extend the range of surveillance operations by forming a link chain of UAVs using multi-hop communication. [37] proposes a formation scheme in which UAVs are placed in the street junctions. A discussion of multi-UAV systems for military reconnaissance and surveillance in urban scenarios is made in [35].

Reconnaissance and Patrolling

First line of defence patrol is a typical use of UAVs that fly in a stationary mode to oversee a specific area. Surveillance tasks may include collecting images of objects and sites of interest spread over wide areas. Sometimes, UAVs must monitor a specific target/area, like ground units that usually patrol periodically around a given region for observation, inspection, or security. For instance, in border policing, a swarm of UAVs can detect not only unplanned human disturbances, including those involving weapons and drugs, but also illegal border crossings [38]. Another context is the reconnaissance over an area to detect ground units in which UAVs cooperate for the mission with unpredictable paths [39]. Differently from the urban or civilian scenario, UAV usage in tactical contexts has strict requirements with respect to communication and coordination delay. This also affects the complexity needed for FANETs, like Tactical Edge Networks [40, 41].

Agricultural Management

Agriculture production management requires crop plant health to be monitored, entering in the context of Precision Agriculture (PA). PA includes all the techniques and methods that use information technology to perform any agricultural study (crop conditions, soil properties, water content, etc.) [42]. Although manned aircraft are used in this sector, the emerging concept of small networks of autonomous UAV swarms is believed to overcome the difficulties related to spatial and temporal resolutions, and with greater precision. Information related to the quality of crops, growth, and other possible issues can be retrieved within a matter of minutes, and adequate measures can be taken. In [42], the configuration of an UAV system for image acquisition is described and tested. The UAV flies on a specific area, automatically following a pre-determined path, and periodically capturing the terrain image. In [43], authors provide an overview of cooperative remote sensing for water management and irrigation control using a multi-UAV system. In [44], a flight path optimization for multiple UAVs in agricultural applications is proposed.

Environmental Sensing

A sensor network is a set of sensors arranged in proximity or within the phenomenon to be observed. These sensors are characterized by limited energy consumption, small sizes and low costs. Temperature, humidity, pressure, light intensity, or pollution levels are typical physical quantities that a sensor network can analyse. Recently, the adoption of UAVs for sensor network is growing, being such growth mostly associated to the heterogeneity nature of flying devices. In [45], the authors study the usage of UAVs for wireless sensor network data collection and implement an algorithm for efficient UAV deployment. The authors in [46] propose the possibility to equip UAVs with pollution sensors, to perform pollution measurements. The UAVs flies autonomously throughout the target area to gather information about the peak levels registered and surrounding values, returning home with the measured data to then characterize the pollution in the target area through heat maps. Clearly, this concept could be extended to the case of smart city sensing, combining UAVs with heterogeneous sensors [47, 48].

2.2 Smart Environments and IoT

A FANET could fulfill all its potential if being part of a more complex system. The possibility to remotely control the UAV swarm or to make it a context-aware system is one of the most ambitious objective. In this matter, we move to the *Smart Environment* (SE) characteristics. In general, the SE paradigm is an evolution of *ubiquitous computing*, an idea to promote an involvement of sensors, actuators, displays and computational elements, embedded in the everyday objects of our lives, and connected through a network. Smart environments are defined as little worlds where different smart devices are working to satisfy an objective. Typically, the work is done by cheap computing power devices and human interaction with the system. In [49], the author differentiates three kinds of environment for a smart system:

- Physical environment context: concerning the physical world and its phenomena (location, time, temperature, light level, pollution, etc.).
- Virtual environment context: concerning the awareness of distribute system about the available services, locally or remotely.
- Human context: concerning the interaction with human and its characteristics (identity, preferences, activities, etc.).

Smart environments are envisioned to be context-aware in this sense, considering part or all of these three elements.

IoT is the paradigm in which Internet extends into the real world comprising all the objects [5, 6, 50]. Physical items are connected to the real world and can be controlled remotely. Whit this, it is hence easy to understand that IoT and SE are in symbiosis with one another. In this dissertation we also analyze the network performance in IoT scenarios where the *things* (in our case, UAVs and vehicles) are characterized by high mobility.

2.2.1 Internet of Vehicles

In this section, we intend to focus on ad-hoc networks composed of any sort of vehicle (e.g., cars) that could be present in our towns and could be part of the IoT. These vehicles could be used to gather, elaborate and disseminate a lot of information having a local scope: traffic condition, local message exchange, pollution sensing, coordinated movements, warnings, etc. In other words, they would form an *Internet of Vehicles* (IoV), with multihop communications flowing among its nodes. Communication among vehicles has been categorized in recent years into various declinations, e.g., Vehicle-to-Vehicle (V2V), Vehicle-to-Road (V2R), Vehicle-to-Human (V2H), Vehicle-to-Sensor and Vehicle-to-Drone (V2D); it is hence clear how an IoV could be composed by much more than just cars.

Thanks to the IEEE 802.11p standard, a lot of research has now been devoted to Vehicular Ad-hoc Networks (VANETs) [51]. There are many applications that could exploit ad-hoc communication in the context of IoV, ranging from environmental monitoring to safety and entertainment [52]. For instance, one might envisage VANETs being employed to disseminate information regarding vehicular movements, or even advertisement toward the Internet (e.g., to web services such as Google Maps) and vice versa [53]. A timely and very challenging application regards the distributed control over wireless links to enable autonomous driving. Autonomous vehicles would in fact need to get as much information as possible from the IoV to determine their best course of action to ensure efficiency, reliability, and safety [54, 55]. This means obtaining information, even through multihop, from surrounding vehicles, from cameras placed on top of lamps and buildings, from sensors around, from hovering UAVs, etc. Automated driving or flying, and the need for IoV-based, distributed control could be used even in case of rescue operation based on autonomous terrestrial and aerial vehicles [25].

Even in case some information must go through the Internet, we can still consider the possibility of having some node acting as Internet gateway and other nodes in our ad-hoc network resorting to multihop connectivity to communicate with the Internet gateway [53, 56]. It is hence easy to see how multihop, ad-hoc routing represents a fundamental task in the envisioned IoV. Note that we consider the case in which UAVs may become more and more popular and able to perform smart autonomous purposes; i.e., we assume that drones will have an increment of popularity and functions like what happened with cellular phones (beside being interesting, this currently seems to be a plausible trend [57]). Indeed, flying drones are becoming frequently seen vehicles with communication and sensing capabilities. We can expect them to evolve in terms of functionalities and reach similar popularity peaks as smartphones. It is even possible that, in future, each person will have a personal drone helping her/him with creating material to populate a social account (e.g. automatically created logs composed by pictures, videos, etc. [58]. Drones may include any unmanned aircraft or self-driving vehicles, ranging in size from a palm-sized to several meters; they may also carry small amounts of cargo. Another possible application is the traffic monitoring, safety and law enforcement over the streets, in which drones can communicate among themselves, or to a specific car, or to a group of vehicles. IoV communication could be exploited even for local message exchange among passengers of cars in a certain area and people nearby. They might share text, voice, images, videos, online gaming, music, news and advertisement, even resorting to data generated elsewhere (e.g., a drone in the sky above them). Regarding local news and advertisement dissemination, data floating solutions could be adopted by having an IoV supporting them [59].

2.3 The Routing Problem

In ad-hoc networks two nodes can communicate with each other (generally, in both directions) if and only if the distance between them is less than the minimum of their transmission range. If a node is placed outside the transmission range of a node that wants to transmit, forward or receive a data packet to or from it, multi-hop routing is used through intermediate communicating nodes. The utility of routing mechanism makes it an integral part of any network and there exists a multitude of routing mechanisms (IP routing for internet, communication protocols that connect machine, robots, drones, and protocols for ad hoc networks). There are various challenges in MANETs, such as energy consumption, bandwidth optimization, shadowing propagation, etc. However, one of the most interesting and challenging issue is the packet routing. In a generic MANET, routing is mostly affected by frequent topology changes (which causes link failures) due to the nodes' movements. All the cited challenges could affect each other in different ways, making complex their study; in fact, MANET's characteristics such as its dynamicity, mobile nature, energy constraints and limited bandwidth, could differently affect the routing process. Different needs and characteristics of various networks present specific challenges, requiring appropriate routing techniques. One way of communication in such networks might seem to simply flood the entire network. However, the fact that power and bandwidth are scarce resources in such networks of low powered wireless devices needs more efficient routing protocols. Moreover, in high dense networks, the flooding method would cause many collisions, due to contemporary sending of the same data packet from multiple nodes and therefore an increase of the packet delivery time. Finally, flooding causes buffers filling, which translates in packets being dropped when buffers full or augmented delays due to long packet queuing times. These factors classify this method as not scalable. A scalable solution is one that performs well even in a large network. In the following section we further explore the position-based routing protocols.

2.3.1 Position-based Routing Protocols

Generally, highly mobile and dense networks pose various challenges to the routing process, since their characteristics such as frequent topology changes, energy constraints and limited bandwidth. To this end, various routing protocols have been proposed during the years [60, 61]. Mauve et al. [62], Stojmenovic [63] and Iche et al. [64] classify these protocols into two main categories: topology and position-based.

Topology-based routing protocols exploit link information to route the packets and maintain up-to-date routing tables, specifying the path or the next-hop where to route a packet from a source to a destination. In this context, there are three strategies that exploit topology information: proactive, reactive and hybrid. A protocol is considered proactive when each node keeps an up-to-date information reflecting the state of the network and this information is used when a message should be sent.

Position-based (or *geographic*) routing protocols use the position of the nodes in the network to make the forwarding decision. The first proposed protocols using geographic information were intended to be adopted in support to the topology-based protocols, limiting the propagation of route request packets into a determined area. Position-based routing protocols exploit a local knowledge; to make the forwarding decision, a node relies on its position information, the position of the destination and the position of its neighbors. To obtain the location information, the nodes use a location service such as a GPS receiver, or other types of location services. To receive the neighbors' position, the nodes make use of a beaconing mechanism in which each node transmits a beacon to its neighbour nodes, containing its position [65]. In general, position-based protocols have the following characteristics:

- each node can determine its position (longitude, latitude and altitude), the position of its neighbors (usually 1 hop);
- the destinations location information is assumed known a-priori (e.g., acquired through external means);
- nodes store the information about their neighbors (e.g., position, speed, direction) in a up-to-date neighbour table;
- the next-hop decision can be made based on the location of the current node (the node holding the data packet), its neighbouring nodes and the destination node.

Nowadays, there are several routing protocols, belonging to both the classes, proposed for MANETs to address the multi-hop routing problem. In a generic MANET, topological information is dynamic and transient in time, and traditional topology-based protocols may not always be suitable in these settings, also considering the possible limited energy supply. On the other side, position-based protocols try to address the issues by employing a different mechanism for path discovery. In this context, each node determines its own geographic position, and makes the forwarding decision relying solely on the destination's position and those of neighbors' nodes. This local decision making, makes position-based routing protocols suitable in scenarios of large and highly mobile networks; nodes do not have to explore the status of the whole network, store routing tables or exchange a huge number of control messages through the network.

Stateless Routing Protocols

A particular and interesting subclass of packet routing consists of the *stateless* routing. A stateless routing protocol handles each forwarding decision as an independent transaction, unrelated to any previous one. Therefore, the stateless design does not need any memory, which would be used to store past decision/forwarding actions. This is a promising class of protocols, especially when considering mobile ad-hoc networks such as FANETs. In this dissertation, we mostly focus on this subclass.

2.3.2 Routing Problem in FANETs and IoV

In MANETs, a lot of research effort has been devoted to devise efficient and reliable routing protocols; however, in case of FANETs, there are aspects in which the classic solutions fall into difficult. As illustrated in 2.1.1, a FANET differs from a MANET in several aspects. In this section we discuss the issues of FANETs from the routing point of view.

First, in FANETs, nodes may be distributed in a 3D space. While topology-based routing protocols are not generally affected by the third dimension because of their relying on a link-state system knowledge, the geometric concepts defined in the most of position-based protocols, must be extended in 3D. Many research focused on studying geographic routing protocols only for classic 2D networks, and being based on spatial position and geometric characteristics, these protocols need to be adopted for the 3D case. It is not trivial, because some assumptions made in the 2D context, such as the ability to extract planar subgraphs in the case of Face, break down in 3D. The transition from 2D to 3D topologies is not well explored and brings new difficulties. For instance, geographic routing following a greedy approach in 3D topologies is intrinsically harder than the same approach employed in 2D topologies since, generally, the number of local minima in 3D topologies is higher than the number of local minima in 2D ones under similar network scenarios (in terms of network density and network field size). Furthermore, the heterogeneity of the methodology used to assess the various proposals has led to the absence of a coherent performance benchmark. Indeed, many proposals have been assessed through numerical simulations and each proposed protocol was tested with its own simulation scenario(s) and settings. Durocher et al. [66] shows the impossibility of routing protocols to guarantee delivery in 3D ad-hoc networks, when nodes are constrained to have information only about their k-hop neighbourhood. This leads to the problem of finding other solutions that can guarantee the delivery of packets, with the least use of resources.

Second, due to mobility related changes in the topology, network routing maintenance imposes additional overhead which may hinder the performance. Furthermore, in large scale networks this situation is worsened, and routing tables size may become a burden. All these characteristics are exacerbated in FANETs. In particular, due to the high mobility of UAVs and the rapid change of link quality, routing table maintenance could result superfluous in terms of reliability and overhead [10].

As a consequence, most of the existing topology-based routing protocols might become almost useless in dynamic, large networks and hence not ideal for FANETs. Stateless, position-based approaches have been proposed to address some of these limitations as they do not need to establish and maintain routes, thereby eliminating the overhead due to frequent updates. This makes position-based protocols more scalable than the topology-based ones [67] and more suitable to satisfy the requirements of a FANET.

Another aspect, also depicted in the IoV context, is the assumption that mobility conditions may change greatly from case to case; we may have scenarios where all nodes are moving, as well as scenarios where just a small percentage of nodes are moving whereas the majority is static or nomadically moving by now and then (and hence can be assumed as static during the duration of a data flow). Furthermore, the scenario where most nodes are static also includes those cases where the IoV is interconnected with static sensors with communication capabilities (e.g., some sensors on top of roadside lamps or buildings' roofs), thus making the considered scenarios even more representative. Several researches explored the routing issues in IoV, where vehicles and UAVs could communicate with each other [68].

2.3.3 Neighbor Discovery

When position-based routing protocols are considered, the general way for a node to announce its presence to the neighborhood is by periodical *beacon* broadcasting. Each node locally transmits in broadcast a short *hello message* (beacon) in order to announce its presence and position, generally expressed as coordinates (x, y, z). Each node, upon receiving a beacon, stores the information of the neighbor node in a *neighbor table*. The beacons are periodically transmitted by each node. If a node does not receive any beacon from its neighbors within a certain time interval, the corresponding node is considered to have left the transmission range, or to be unreachable for some reason, and is deleted from the neighbor table. The beacon time period can be fixed or based on some current node's characteristics (speed, position, available energy, node type, etc.).

The problem of neighborhood discovery can be identified when the network nodes move. In [65], the authors analyze the impact of diverse beacon settings (beacon timing, node density, node speed, etc.) on the routing performance. In this work, it is shown that if a periodic beacon between nodes is started, the speed of the nodes can affect the correctness of position information. Inaccurate or out-dated neighborhood information may severely affect position-based routing protocols. If a periodic beaconing is used, there are three different situations that can occur when nodes moves:

1. Nodes are listed in the neighbor table with an inaccurate position, but are still accessible.

2. A node moves within the transmission range of another node, but was not visible before (because it had not received the beacon).

3. A node moves out of transmission range of another node. So, the routing has still that node in its neighbor table and may send a message to it, but the MAC layer is unable to reach the node. After some retransmission, the MAC layer drops the message or notify that it was not able to send and send back the packet.

Due to these conditions, the routing protocol may make errors in selecting the next node or transmit to a node no more reachable, causing unnecessary use of network resources and energy. No much research has investigated this problem, which could become important for the effectiveness and efficiency of position-based routing.

To improve the accuracy of the beaconing, several strategies are proposed [65]:

- Time-based beacon: each node sends a beacon periodically (every t seconds).

- Distance-based beacon: each node sends a beacon only when it is moved of d meters from its original last sending position.

- Speed-based: the frequency of the beacons is proportional to the speed of the nodes.

Clearly, the use of beaconing is closer to a more realistic scenario, where nodes do not know a priori their neighbors. This assumption leads to the conclusion that the delivery ratio can be reduced, compared to the ideal case. This fact is then confirmed by our tests.

2.3.4 Taxonomy of Position-based Routing Protocols

Several taxonomies have been proposed for MANETs routing protocols [62]. Our focus is on position-based approaches proposed for i) 3D topologies or ii) 2D topologies but applicable even to the 3D ones. To this end, we propose a taxonomy differentiating the proposals along *path* and *forwarding* strategy axes.

The path strategy represents how a data packet traverses a network. The packet can use either a single path to reach the destination or multiple paths. A single path strategy requires that each node forwards the packet to only one of its neighbors. So, there is only one copy of a packet in the whole network. Algorithms that employ a single path strategy generate less communication overhead. On the other side, the multipath strategy allows a node to forward multiple copies of the same packet to several neighbors, following specific forwarding strategy, or even to split traffic and distribute it along multiple disjoint paths [69].

The forwarding strategy describes the forwarding criteria used to select the candidate next (neighbour) node(s). In Fig. 2.3, a taxonomy differentiating the main approaches is presented, with relative classes subdivision.

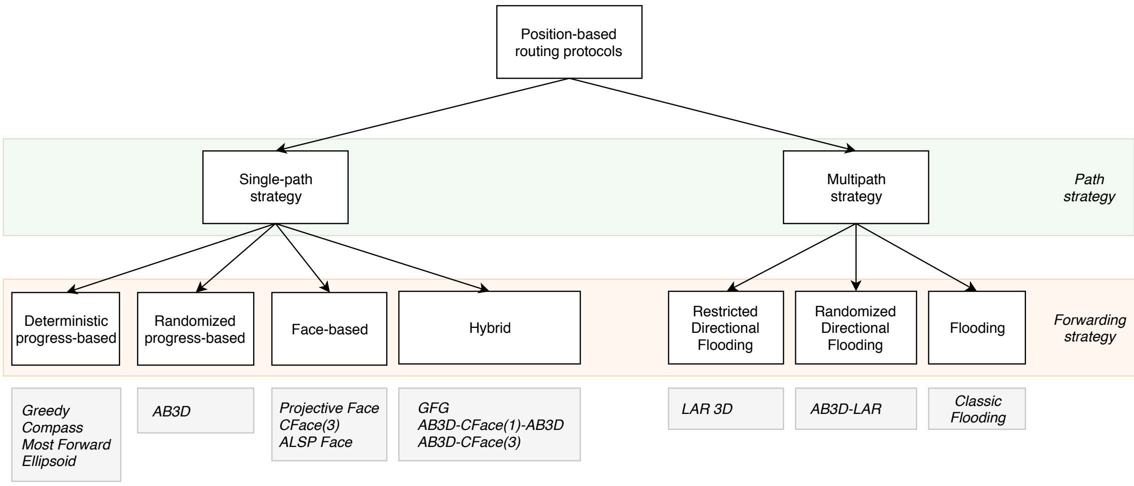


Figure 2.3: Taxonomy of Stateless 3D Position-based Packet Routing Algorithms for FANETs.

Every strategy exploits different methods and geometric models. Considering the single-path subclass, the three main approaches are categorized as deterministic progress-based, randomized progress-based and face-based. In deterministic progress-based routing algorithms, the current node holding the packet forwards the packet at every step to one of its neighbors that makes progress towards a destination. Randomized progress-based strategy is like deterministic progress-based method but in this case the next node is chosen randomly or according to a probability distribution, from the set of candidate nodes. Face-based strategy uses an algorithm, called Face, that advances the packet between the faces by considering the right-hand rule, always guaranteeing the packet delivery to the destination, in the context of planar (2D) networks.

Considering the multipath subclass, the first variant is Flooding, in which a node sends a copy of the same packet to all its neighbors. Flooding introduces an overhead, as packets get duplicated in the network consuming network's available resources. Limited or controlled-flooding, by which a node sends copies of the packet to a subset of its neighbors, tries to restrict the propagation of the duplicate packets within a defined area; examples of limited flooding protocols are proposed in [70, 71]. Limited flooding techniques are shown in Fig. 2.3 as restricted directional flooding (deterministic decision) and randomized directional flooding (randomized decision).

In Chapter 3, an accurate description of all the considered position-based protocols is given.

2.4 Delay Tolerant Networks

In particular scenarios, infrastructure-based services access might not always be available or convenient. Examples are disaster scenarios, where the infrastructure networks could have fallen down, or in Search end Rescue missions that could need the deployment of a network in areas not covered by infrastructure. On the other hand, cellular networks could become a not-optimal solution for several reasons, like coverage gaps, interoperability issues between carriers, and the service model. Ad-hoc networks seems to be a potential solution for Search and Rescue scenarios, but the dynamic nature of such context could affect the routing performance, due to frequent link failures, delays, packet route errors, etc. [72].

The use of UAVs to deploy a mobile network mostly derives from the inability of infrastructure-based networks to work in these situations, due to damaged or lack of infrastructures [73]. Their ability to easily and quickly reach a point of interest makes them an excellent solution for diverse tasks. UAVs have recently been coupled with an interesting network paradigm, called *Delay-tolerant network* (DTN) or *disruption-tolerant network.*

A DTN is an approach that seek to address the technical issues in networks that may lack continuous network connectivity, that does not guarantee an end-to-end path. The DTN paradigm is best suited for the mentioned scenario due to their inherent tolerance to intermittent connectivity, providing buffering and carrying data [74, 75]. DTNs use the store-and-forward policy, where each node holds the entire packet/message or chunks of it (bundle) to forward it to any other node at a later stage. An ideal solution comes through the integration of DTNs into FANET infrastructure, enabling and maintaining network connectivity under high delays and intermittent links [76, 77].

In [78], a detailed overview of DTN technology applied to the FANET infrastructure is provided, with special focus on tactical scenarios and DoD (Department of Defence) programs. UAVs as DTN nodes used for carrying data from and to isolated buildings is considered in [79], where a smart city is the context scenario. Other works propose the implementation of an UAV architecture to carry the data within a building, creating an indoor DTN system [80].

One objective of our work is to also consider the DTN architecture in FANET applications, se well as to propose and analyze a routing solution for DTNs that takes into account geographic information of the nodes and, in particular, waypoints' location. When we consider a FANET application, it is very common for UAVs to already have a planned movement path, or that, at least, known target coordinates to be reached. The UAVs, through *hello messages*, could communicate their scheduled plan. Thanks to this information, it is possible to devise a routing protocol based on reasonable assumptions and predictions of node encounters, in order to calculate the time to reach a certain location, which could be the base station. In essence, each node may select the best available carrier to send a packet to the destination, according to the waypoints' location.

Chapter 3

Related Work

In this chapter, we illustrate the different proposed solutions to face the different issues in the routing problem. As said in Sec. 2.3.1, all the protocols can be divided into two main categories: topology and position-based. In this chapter we illustrate the stat-of-the-art literature relevant to the position-based routing protocols, the focus of our work. To this end, in Sec. 3.2, we provide a survey of position-based routing proposals, and how these solution are extended in 3D. Next, in Sec. 3.3, we illustrate some proposals that use geographic information for packet forwarding in the context of DTNs.

3.1 Topology-based routing Protocols for Ad-hoc Networks

In this section, we discuss the most relevant work on topology-based routing protocols in MANETs. These routing protocols are adopted as a comparison metric for our performance evaluation.

Destination-Sequenced Distance-Vector (DSDV [81]) is based on the Bellman Ford algorithm. DSDV is a proactive protocol that is enhanced by the use of sequence numbers in the routing tables to avoid the loop problem. In this way, the most recently updated paths have a higher sequence number. Each node updates its sequence number every time that it sends an update and maintains a routing table with an entry for each other network node. Each entry holds a sequence number, which is updated with each change, used to avoid cycles and discriminate between old routes and new ones. Updates are transmitted by nodes periodically or as soon as major changes take place. When a node receives two different paths to the same destination, it chooses the one with the greater sequence number, or the one with less hops in case of equal sequence number. To reduce the overhead of network traffic, this routing protocol uses two types of update packets:

- Full dump: all complete routing information are sent.
- Incremental dump: only updates are sent.

Ad-Hoc On Demand Distance Vector (AODV [82]) is a reactive protocol whereby routes are established on-demand, as they are needed. In AODV, the network is silent until a connection is needed. When a node needs to find the path toward a certain destination, a Route Request packet (rreq) is sent in broadcast over the network. Other nodes that receive this rreq packet forward it and record the node from which they have received it by creating or updating the temporary route to reach the source node in their routing table. When a node possessing the information about the route to the destination receives a rreq, it answers by sending a rrep packet through a temporary route to the requesting node. The requesting node then begins to use the route that has the least number of hops through other nodes. When a link fails, a routing error is passed back to a transmitting node sending a Route Error packet (rerr), and the process repeats. Nodes use a sequence number so that they do not repeat route requests that they have already forwarded. The advantage of AODV is that it does not create extra traffic in maintaining the routing tables if cases they are not used. On the other hand, it requires more time to establish a route when compared to DSDV.

Dynamic Source Routing (DSR [83]), like in AODV, the source node initiates a route discovery process generating a rreq packet which is flooded into the network. The rreq packet contains a list of hops which are incrementally added into the route request packet header as it is propagated through the network. Once the rreq reaches the destination or a node that has a path toward the destination, a rrep is sent back along the reverse path collected in the rreq. The main difference between DSR and AODV is in the way the route information is kept: in DSR it is stored at the source and in the header of the transmitted control packet, while in AODV it is stored at the intermediate nodes.

Other topology-based routing protocols that deserve to be mentioned are *Topology Broadcast based on Reverse Path Forwarding* (TBRPF) [84] and *Optimized Link State Routing Protocol* (OLSR) [85].

3.2 Position-based routing Protocols for Ad-hoc Networks

In this section, we discuss the state-of-the-art of the position-based routing protocols in MANETs, specifically discussing their extension in 3D topology. We describe the forwarding algorithms employed by the routing protocols. For each forwarding strategy it is explained how it is extended in 3D.

3.2.1 Notation and Models

In the following we provide a list of conventions and notations needed to better comprehend this dissertation. In order to provide a uniform and fair treatment of all the routing algorithms, a common terminology is introduced.

- A MANET model is represented, in R^2 (2D MANET) and R^3 (3D MANET, FANET) spaces, by a geometric graph $G = (V, E)$, consisting of a finite set $V = v_1, v_2, ..., v_N$ of nodes and a subset E of Cartesian product $V \times V$, the element called edges (links from a node to another).

- In this thesis there is the convention that all nodes have the same communication range r. The area within the communication range of a node u is indicated as $R(u)$ and it is represented as a disc in 2D space and as a sphere in 3D space; the graphs thus obtained are called *Unit Disk Graph, UDG(V, r)* and *Unit Ball Graph, UBG(V, r)* respectively.

- We define $dist(u, v)$ as the geometric distance between two nodes u and v, given by the formula of the Euclidean distance:

$$dist(u, v) = \sqrt{(u_x - v_x)^2 + (u_y - v_y)^2 + (u_z - v_z)^2}.$$

- Two nodes u and v are connected by a link if $dist(u, v) \leq r$; two connected nodes are called neighbors.

- For a node v, we define its *neighborhood* as the set of nodes $u_1, u_2, ..., u_n$ directly connected to v; The neighborhood of v can be represented with $N(v)$; the size of $N(v)$ is indicated with n_v.

- A path from a node u to a node v is a sequence of nodes $u = n_1, n_2, ..., n_k = v$, such that n_i and n_{i+1}, $1 \leq k - 1$, are neighbors.

- The *source node*, s, is a node that produces a generic data packet to be sent to a receiver node, called *destination node*, d.

- The *current node*, c, is the node that applies the routing algorithms at the current time and holds the data packet to be forwarded.

- The *previous node*, p, is the node that sent the data packet to c in the previous step.

- A node in which the data packet gets stuck because of lack of a neighbor node that makes progress with the considered heuristic, is called *local minimum*, indicated as l.

- A line passing through two nodes u and v is referred as (uv).
- A segment that starts from a node u to a node v is referred as $\overline{uv}$.

3.2.2 Deterministic Progress Forwarding

Progress Forwarding (also called *greedy forwarding*) is a single-path forwarding strategy, based on the notion of *greedy* paradigm. In each step, a node forwards the packet to the neighbor node that performs the best *progress* to the destination node. In general, the algorithms that follow this strategy use only local information; in the case of routing, the local information is the neighborhood. Thus, each node forwards the data packet to the neighbor that is most suitable from a local point of view. In Fig. 3.1, a graphical illustration of the different progress forwarding strategies is provided.

The most simple forwarding algorithm is *Greedy*, that tries to forward the data packet closer to the destination. In each step, c transmits the data packet to the node u_i in $N(c)$ that minimizes the distance to d, i.e., to the u_i which distance is:

$$min\{dist(u_1, d), ..., dist(u_n, d)\}.$$

If c does not find any neighbor node that offers a progress to d (i.e., there is a void), Greedy fails and c becomes a local minimum l. More specifically, there are two classes of forwarding algorithms that adopt the concept of distance to d, which differ from one another based upon the nodes considered as part of the neighboring set:

- Greedy [86]: c forwards the packet to the closest u_i to d in $N(c)$, which is also closer respect to c.
- GEDIR [87]: c forwards the packet to the closest u_i to d in $N(c)$, but not necessarily closer respect to c.

In Greedy, only the neighbors that are closer to d than c are considered. If no one is closer to d, the algorithm fails. In GEDIR, all neighbors are considered. So, even the nodes that are in *backward* direction can be chosen, and the only kind of loop that may be formed using this algorithm is the local loop between c and the previous node p that sent the packet to c in the previous step [87]; thus, the packet has reached a local minimum c and the algorithm fails. Fig. 3.1 shows a step's example of the progress-based algorithms. c is chosen among five possible candidates $(n_3, n_4, n_5, n_6, n_7)$ which are in the direction of d. With *Greedy* the choice falls to candidate node n_5 (the closest toward d).

Other important algorithms that use the notion of progress are *Compass*, *Most Forward*, and *Ellipsoid*. Compass (or *Directional*, *DIR*) [88] uses the direction of neighbors to

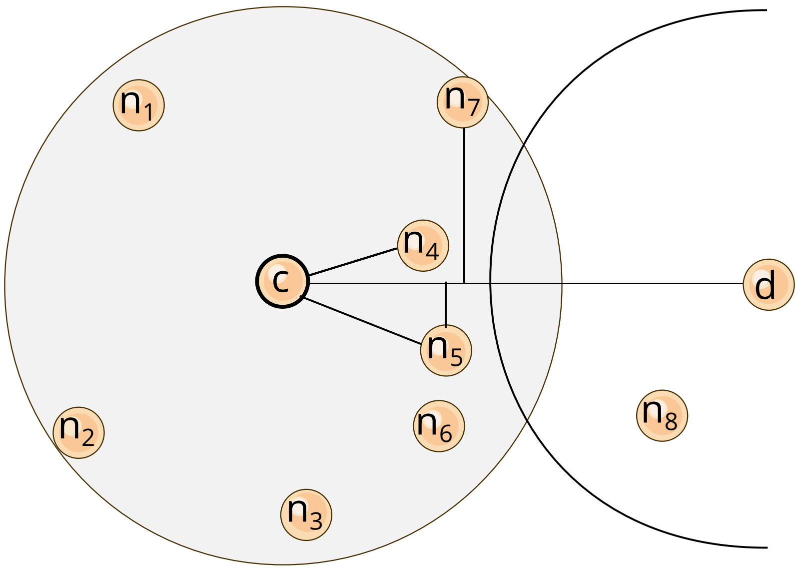


Figure 3.1: Illustration of several next nodes chosen by c using the progress-based forwarding strategies.

select the best forwarding node. In each step, c uses the location of d to calculate its direction. Then, it forwards the data packet to the node u_i in $N(c)$ that forms the minimum angle to d, i.e., to the u_i which angle is:

$$min\{\angle u_1cd, ..., \angle u_ncd\}.$$

With reference to Fig. 3.1, c chooses n_4 as the next node, since the angle $\angle n_4cd$ is the smallest among all. Compass is not a loop-free algorithm and it is demonstrated in Fig. 3.2, where a loop consists of four nodes, n_1, n_2, n_3 and n_4. n_1 and n_3 are not connected (because they are outside their own transmission range). n_2 receives a packet from n_0 and selects n_1 to forward the packet because the direction of n_3 is closer to d than the direction of its other neighbor n_4. Similarly, n_1 selects n_3, n_3 selects n_4, and n_4 selects n_2. The travel continues with this loop. If this situation occurs, the nodes in the loop are not able to recognize the loop itself, unless the packet id is memorized; in this case a node may recognize if the same packet arrives and take appropriate countermeasures (e.g, packet drop, change of the next node, etc.).

Most Forward, (MFR - Most Forward Routing) [89] is very similar to Greedy, but, in this case, c forwards the packet to the node u_i in $N(c)$ whose projection on the line (cd) is closer to d. If the packet reaches a local minimum (there is no neighbor projection that makes progress towards d), the algorithm fails. In Fig. 3.1, c selects n_7 as the next node, since the latter has the smallest projected distance to d on (cd). MFR is a loop-free

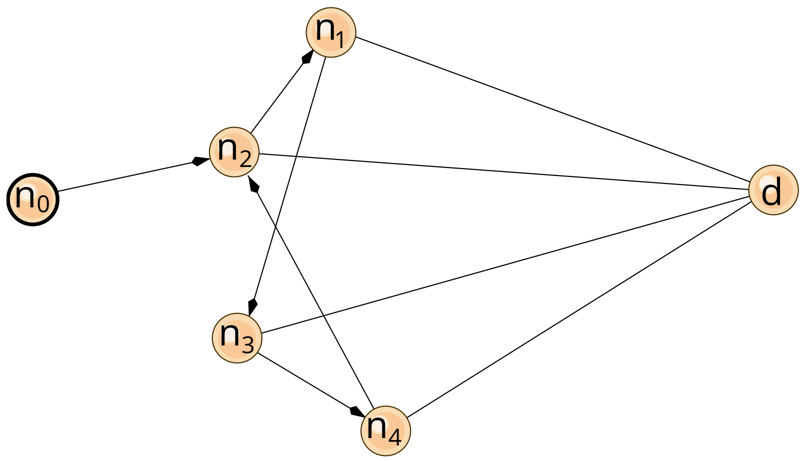


Figure 3.2: A loop with Compass.

algorithm, for the same reason as Greedy.

In the Ellipsoid algorithm [90], c forwards the packet to the node u_i in $N(c)$ that minimizes the sum of the distance from c to u_i and the distance from u_i and d. If the packet reaches a local minimum, the algorithm fails. Referring to Fig. 3.1, the algorithm chooses n_4 as the next node to handle the packet. Ellipsoid aims to select the next hop node as the node that lies closer to the straight line joining the forwarding node and the destination.

3D extension. The extension of deterministic progress algorithms in 3D space, illustrated in [91], is relatively simple, as we need only the definition of the Euclidean distance between two points u and v in three dimensions (see Sec. 3.2.1).

3.2.3 Randomized Forwarding

Randomized strategies try to solve the local minimum problem described previously, by randomly choosing the next node from a subset of the current neighboring nodes to make progress toward a destination. Already since 1984, some authors proposed the idea of randomized algorithms by exploiting a random walk approach [92]. Typically, randomization is performed choosing a number of candidate neighbor nodes, and forwarding the packet to one of them with a certain probability. In [93], authors present examples of several situations where Greedy and Compass fail, proposing the *Randomized Compass Routing* algorithm that solves the problem, and works as follows; let u_{cw} (clockwise neighbor) be the node in $N(c)$ over the line (cd) that minimizes one of the progress values defined in previous progress-based algorithms (typically, the distance) and u_{ccw} (counterclockwise

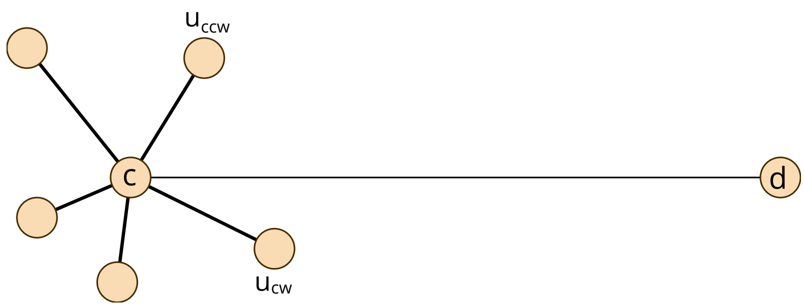


Figure 3.3: Definition of u_{cw} and u_{ccw}.

neighbor) be the node in $N(c)$ under (cd) that minimizes the same one (see Fig. 3.3); the algorithm moves the packet to one of u_{cw}, u_{ccw} with equal probability. In [94], variations of randomized Greedy and Compass are proposed to enhance the packet delivery, using diverse neighboring sets and heuristics.

Since, for randomized algorithms, the definition of failure is not defined (once the local minimum is reached, the packet can still be forwarded), in this work we use the convention to consider them failed when the number of hops in the path computed so far exceeds a threshold value, called *Local Threshold Random* (LT-R).

3D extension. The extension of randomized algorithms in 3D environments is not trivial, as it is not obvious how to best determine the candidate neighbor nodes. The reason is that, considering the proposal of [93], in a 3D graph there is no concept of *above* and *below* a line passing from source to destination. In [95], authors propose a 3D extension for the randomization-based algorithm that uses planes. The proposed algorithm is called *Above/Below 3D* (AB3D), and uses a plane to divide the 3D space in two regions. It uses three parameters: m indicates the number of possible candidates neighbors chosen in $N(c)$, and can assume the value 3 or 5; R represents the selected strategy for choosing candidate nodes and can be one of CM (as Compass), GR (as Greedy) or MF (as Most Forward). Similarly, S represents the probability weight when randomly choosing between more than one candidate neighbors and can be a uniform probability (U), or according to the candidate node's angle with the destination (A), or according to the candidate node's distance to the destination (D).

The plane is chosen selecting three points: c, d and the first candidate neighbor node n_1 in $N(c)$ chosen according to R. If $m = 3$, other two candidate nodes are selected: one neighbor n_2 is chosen above the plane according to R, and one n_3 is chosen below the same plane according to R. If $m = 5$, a second plane, perpendicular to the first one, is defined. It passes through c and d, such that the intersection line between the two planes

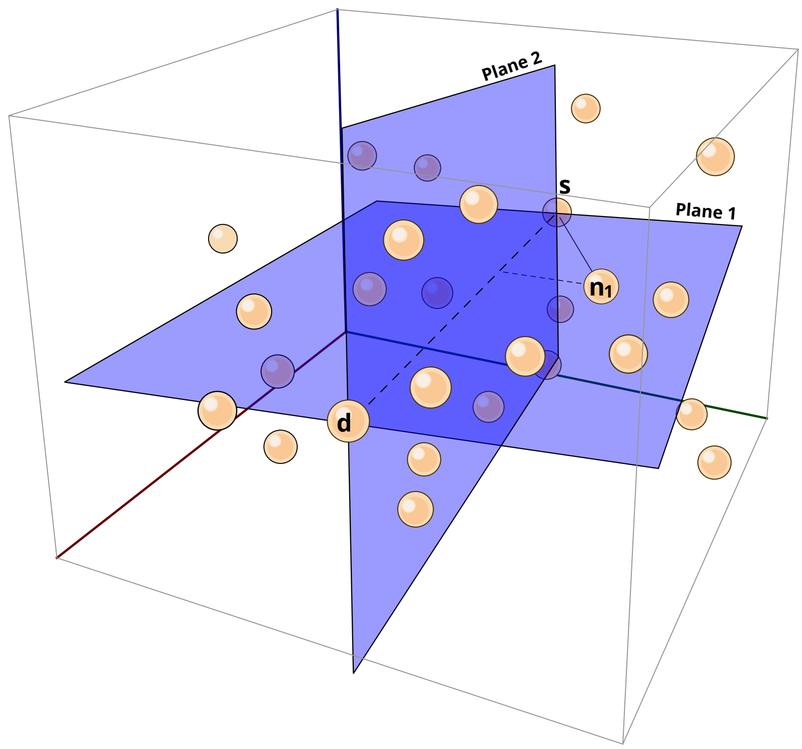


Figure 3.4: In AB3D, Plane 1 passes through s, d and n_1, and Plane 2 is orthogonal to Plane 1. Both planes contain the line (sd).

is (cd). In this case, in addition to n_1, the algorithm chooses four neighbors n_2, n_3, n_4, n_5 from $N(c)$, each one in one side of the four regions that result from the intersection between the two planes. Once these candidates are determined, c selects one of these nodes randomly, according to the probability weighting determined by S, and forwards the packet to the chosen node. In Fig. 3.4, the planes representation and relative regions subdivision is illustrated.

In [96], authors proposed an hybrid algorithm, called *Greedy-Random-Greedy* (GRG), that uses Greedy as the primary stage and a random walk approach as recovery stage. It starts with Greedy until finds a local minimum m. At this point, GRG stores the distance $dist(m, d)$ on the packet, and switches to the random phase, where m randomly selects one u_i from $N(m)$. If $dist(u_i, d) < dist(m, d)$, the algorithm resumes the greedy forwarding, otherwise it continues randomly. Note that we can adopt AB3D as the random phase.

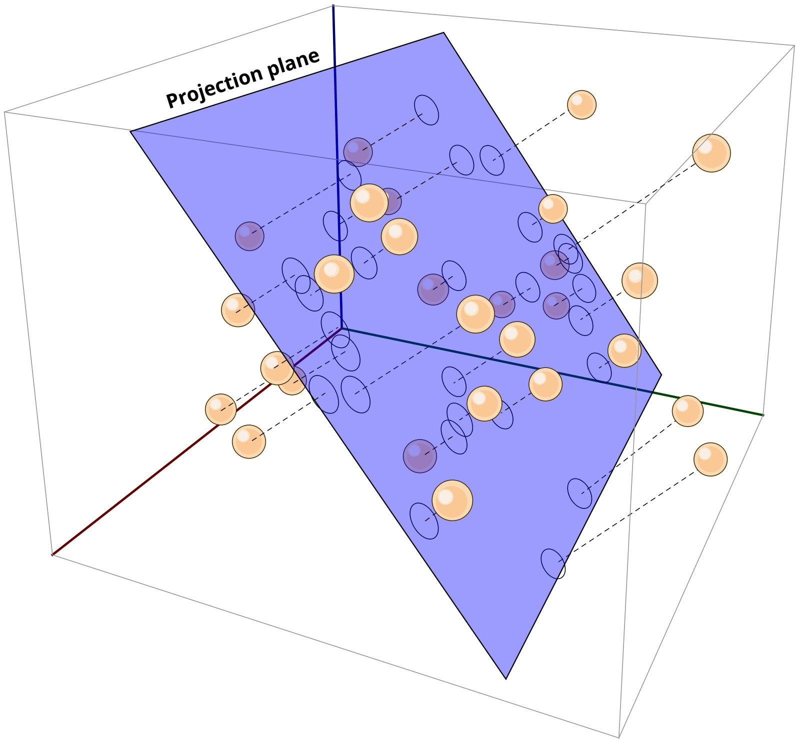


Figure 3.5: Nodes' projection onto a plane.

3.2.4 Face Forwarding

Face, a.k.a, *Perimeter* forwarding method has been proposed for the first time in [88] and optimized in [97]. This method is a typical planar graph technique which involves walking adjacent faces using the right-hand rule and it is the first geographic routing algorithm that guarantee a delivery rate of 100% in 2D network graphs. Position information is used to extract a connected planar subgraph, called *Grabriel Graph* (GG) [98], which contains faces whose corners are the terminal node and does not contain crossing edges. In [97], two algorithms called *Face1* and *Face2* are discussed, with the latter being more optimized than the former and hence considered here. With Face2, the packet is routed over the faces of the GG that are intersected by the segment $\overline{sd}$. Each face is traversed using the right(left)-hand rule, unless the current edge crosses $\overline{sd}$ at an intersection point p. At this point, the algorithm switches to the next face sharing that edge. This process is repeated until the packet arrives to d.

3D extension. Since the face strategy cannot be performed directly on a 3D graph, a proposed solution is to project the nodes onto a plane, as seen in Fig. 3.5, to perform Face.

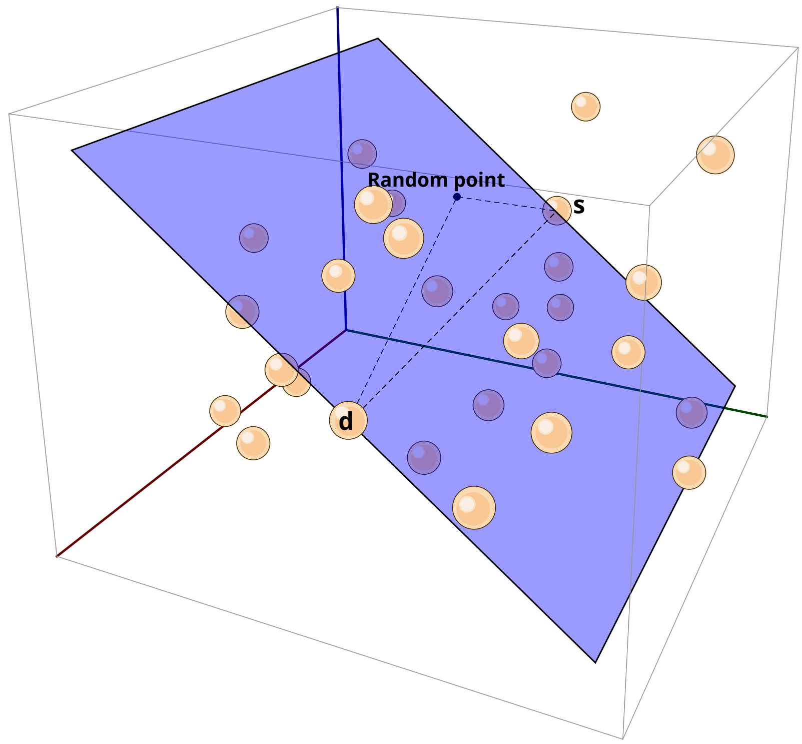


Figure 3.6: Computing a plane with Projective Face.

The first extension of face-based strategy in 3D space uses two orthogonal planes intersecting at the line connecting source and destination. In [99], authors propose *Projective Face*, in which the nodes of the network are projected onto the plane that contains the segment $\overline{sd}$ and a third point chosen randomly. Then, Face is performed on this projected graph. If the routing fails, the nodes are then projected onto a second plane, orthogonal to the first one which contains the same $\overline{sd}$. Then, Face is again performed. Fig. 3.6 shows an example of plane configuration in Projective Face. Similarly to the random strategies, since the delivery rate is not guaranteed, and although the packet reaches a local minimum, it can still be forwarded to the node with the least backward progress. Hence, the algorithm makes use of a local threshold value, that we call *Local Threshold Face* (LT-F), in order to terminate the algorithm in case it does not reach the destination within LT-F hops. This is necessary because the algorithm may get stuck in a loop. More precisely, in this version of the algorithm, the LT-F counter is started twice, once for the first plane and then for the orthogonal plane, obtaining a global threshold value, GT = 2 LT-F.

CoordinateFace(3) (CFace(3)), proposed in [95], uses another set of projection planes, which is composed by the planes xy, xz and yz for the projection of the nodes. With CFace(3) all the nodes are projected on the first xy plane ($z = 0$), and Face is started on

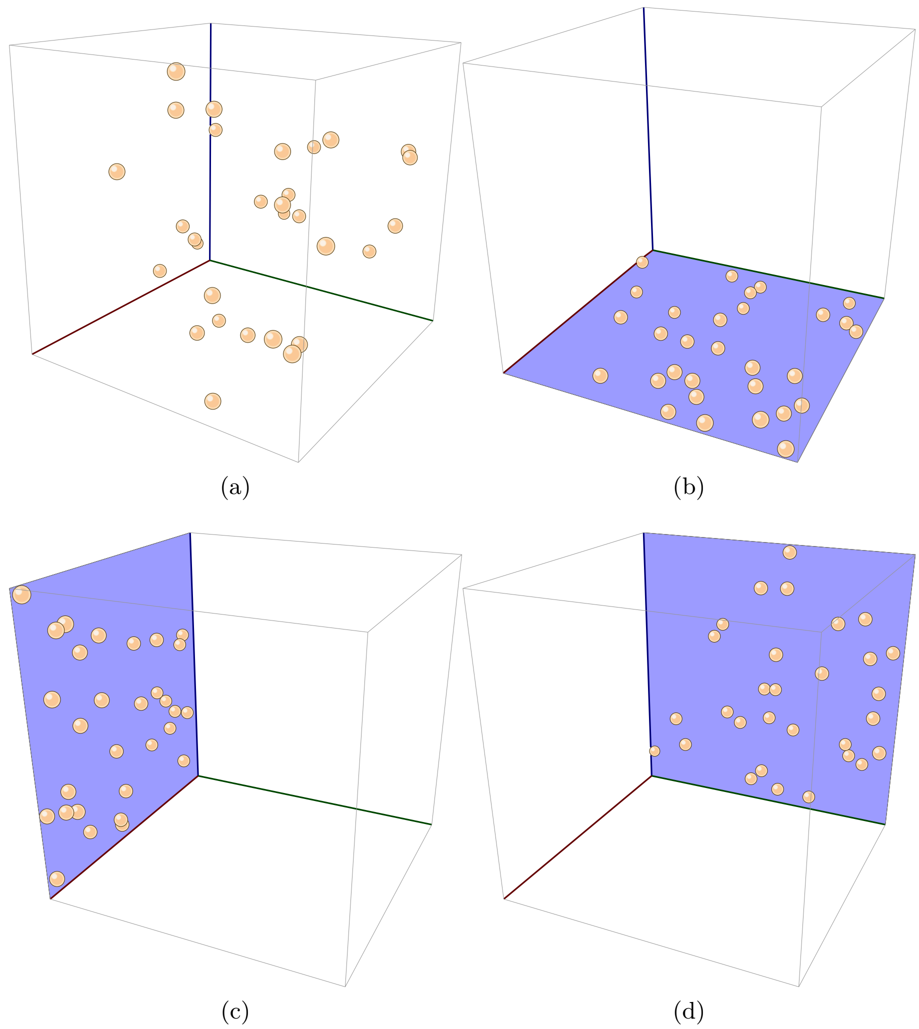


Figure 3.7: Projection of graph nodes (a) on the three planes xy (b), xz (c) and yz (d) in *CFace(3)*.

the projected graph. If the packet does not arrive at the destination (LT-F has expired), the original coordinates of all nodes are projected on the xz plane ($y = 0$) and Face is performed again. If the packet does not reach the destination, the original coordinates of all nodes are projected on the yz plane ($x = 0$) and Face starts again. If the packet does not arrive even through this last plane, the algorithm fails. Fig. 3.7 shows all the three projections on the coordinate planes. Note that, in this algorithm, GT is at most 3 LT-F.

Authors of [100] proposed three heuristics to modify and improve the Projective Face algorithm. The new obtained algorithm is called *Adaptive Least-Squares Projective Face* (ALSP Face). The three heuristics are:

- Least-Squares Projection (LSP) Plane;
- Adaptive Behavior Scale (ABS);
- Multi-Projection-Plane Strategy.

In Projective Face, a third point is chosen randomly, together with the source and the destination points, to compute the first projection plane. Instead, ALSP Face chooses the third point adopting a mathematical optimization technique (first heuristic) for finding the best fitting plane to the set of neighbor nodes. Using the set composed by c, its $N(c)$ up to one (or two) hop(s) away and d, the initial projection plane is determined by using the *least-squares* error minimization on the perpendicular distance of these nodes to the plane, called *least-squares projection plane* (LSP plane). To maintain the local characteristic of the routing algorithm, authors propose that only c, d and $N(c)$ within the 1(2)-hops scope are selected as the set of nodes for computing the least projection plane. Then, nodes are projected on this plane and Face is performed. This LSP plane aims to have a less distorted projected graph so that the number of crossing edges can potentially be reduced. The second heuristic defines a parameter called *Adaptive Behavior Scale* (ABS), used to determine when recalculate the LSP plane, in order to ensure that the plan is always appropriate for c. The third heuristic uses a set of projection planes arranged in a fixed order around an axis. The algorithm switches between these planes, following the order, to disrupt any looping that may occur during routing. In [100], it is said that performing the face routing on the additional projection plane, significantly increases the delivery rate. Therefore, the third heuristic tries to increase the number of projection planes. But this is true only up to a certain point; even if it is true that the delivery rate is slightly increased, it is also true that the path followed by the packet becomes enormously long, because, for each projection plane, the threshold value (here, LT-F) is reset to its pre-set value. Such high *path dilation* value might not be acceptable in a real deployment. In this work we consider the third heuristic using only two additional planes, chosen as in Projective Face.

The *Greedy-Face-Greedy* algorithm (GFG) [97], also referred to as *Greedy Perimeter Stateless Routing* (GPSR) [101] for 2D networks, uses a combination of greedy and face methods. With GFG, a flag is stored in each data packet. This flag can be set into greedy-mode and face-mode, indicating whether the packet is forwarded with either Greedy or Face. The algorithm starts from s with Greedy, setting the packet into greedy-mode and forwarding it. Each node that receives a greedy-mode packet searches among its neighbors the node that is closest to d. If this node exists, the packet is forwarded to it, otherwise it is marked to face-mode. GFG forwards the face-mode packets performing the same planar graph as Face2. Moreover, when a packet enters in face-mode at node x, GFG sets the

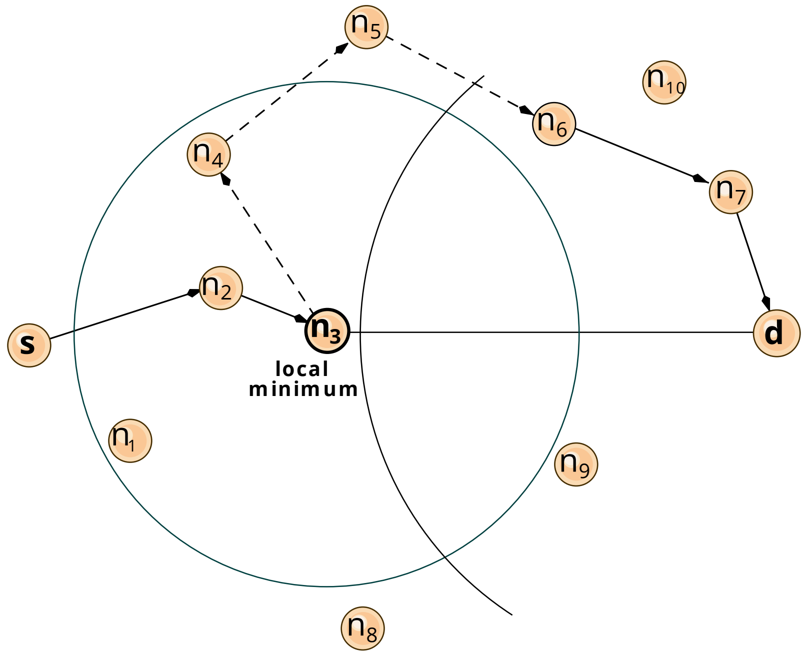


Figure 3.8: Performing of *GFG* on a 2D graph. Solid arrows represent greedy-mode forwarding, dashes arrows represent face-mode forwarding.

segment $\overline{xd}$ as a reference segment to the crossing with the faces, and records in the packet the location of x as the node where Greedy failed. This information is used at the following hops to determine whether and when the packet can be returned to greedy-mode: upon receiving a face-mode packet, c first compares its location with the location of x. The packet returns in greedy-mode if the distance from c to d is less than the distance from x to d. In this case, the algorithm continues the greedy progress towards d. Otherwise, GFG continues with Face. Fig. 3.8 shows an example where the packet starts from s and reaches n_2 and n_3 in greedy-mode, stopping at n_3, that is the local minimum. A this point, from n_3, the face-mode is started, forwarding the packet on progressively closer faces of the planar graph, each of which is crossed by $\overline{sd}$. So, the packet traverses n_4, n_5 and n_6 in face-mode. n_6 is closer to d than n_3, the current local minimum, so the packet can be returned to greedy-mode, and delivered to d. In a 3D context, GFG uses ALSP Face as face-mode [102], which offers the best performance in terms of delivery rate.

The *AB3D-CFace(1)-AB3D* algorithm [95] is similar to GFG, but uses AB3D instead of Greedy. It starts with AB3D, until d is reached or the relative local threshold LT-R is reached. In this last case, the algorithm switches to CFace(1). CFace(1) traverses one projected plane, which is chosen randomly from xy, yz, or xz planes, starting from the node in which the algorithm is switched. At this point, LT-F is initialized to 0 and CFace(1) starts. If the destination is not reached during this phase and LT-F is passed (a looping occurs), the algorithm goes back to AB3D and LT-R restarts at 0.

The *AB3D-CFace(3)* algorithm [95] is a similar algorithm that starts as AF(1)A, but instead of going back to AB3D if the phase in a projective plane fails, it tries the other two projected planes, defined as in CFace(3). This algorithm starts with AB3D and if the destination is not reached and LT-R is passed, the algorithm switches to CFace(3) using the first xy plane. Again, if a loop occurred (LT-F is reached), it switches to the yz plane, and finally the same process is repeated for xz plane.

3.2.5 Multipath Forwarding

Multipath forwarding is a technique that use multiple copies of the same packet. Each node may forward the packet to more than one of the neighbors. *Flooding* is the main example, used by routers to forward an incoming packet on all lines except the one it came from. Multipath strategies use a *partial* flooding mechanism to deliver the packet to the destination and present the problem of packet redundancy: a node can receive a packet for the second time, or even several times, resulting in a very high traffic if it does not have a mechanism to recognize and eliminate these duplicate packets. Moreover, this situation, can lead to a considerable number of duplicated packets.

In this dissertation, a partial flooding strategy is considered, in particular those called *restricted directional ooding*, which use the positional information to send packets to more than one of the neighbors that are located closer to the destination, than the forwarding node itself. Examples of routing algorithms that use this strategy are *Distance Routing Effect Algorithm for Mobility* (DREAM) [103] and *Location-Aided Routing* (LAR) [71]. We consider only LAR algorithm for its 3D extension and comparison.

LAR uses the position information of nodes to restrict the flooding process. With the available information of source node and destination node positions, the flooding area is (in 2D) a minimum size rectangle with $disk(s, r)$ in one corner and the *expected zone* in the other corner. The expected zone, in the original definition [71], is a circle around d of radius equal to $v_{max} * (t_1 - t_0)$ where t_1 is the current time, t_0 is the timestamp of the position information that c has about d, and v_{max} is the maximum speed of the node in the network. In our case, since we analyze static scenarios ($v_{max} = 0$), the expected zone is $disk(d, r)$.

3D extension. In the extension of LAR in 3D space [104], s computes the expected zone for d, which is a sphere around d of radius equal to $v_{max} * (t_1 - t_0)$ where t_1 is the current time, t_0 is the timestamp of the position information that s has about d, and v_{max}

is the maximum speed of the node in the network. This zone is used to define the 3D flooding area, which is the minimum size rectangular box with $ball(s, r)$ in one corner and the expected zone in the opposite corner. In our case, since we consider a static scenarios ($v_{max} = 0$), the expected zone is $ball(d, r)$.

The hybrid *AB3D-LAR* algorithm was proposed in [104, 105], and it is a hybrid variant combining AB3D with LAR. This algorithm tries to reduce the high path dilation of LAR. All the combinations in AB3D-LAR use the same partitions as in AB3D. The difference is that, while AB3D selects only one of the m candidates chosen from the neighborhood, AB3D-LAR transmits the packet to all those selected candidates which are within the rectangular box defined as in LAR.

3.2.6 Memory Aware Forwarding

Memory Aware Forwarding strategies use the node's memory to support the routing process. Typical example is when the packet id is stored into the node in order to be considered later (for example to eliminate packet duplication). The perfect example is *Depth First Search* (DFS) [106], which is a distributed approximation of the classical depth-first search algorithm from where the name derives. The proposal follows a progress-based forwarding strategy like Greedy, and the forwarding strategy takes place as follows: each node has a list in its local memory that contains the id of received packets. If a packet arrives in a node, its id is stored in this list. If a packet id is not found in the local memory of a node, this node is marked as white; otherwise, if the packet id is present, the node is marked as gray (which means that it has received packet at least once). The process of visiting nodes coincides with sending packets between nodes. If a node receives a packet for the first time, it memorizes also the node that forwarded that packet. So, each node stores a list of tuples $id, from$, where id is the packet id and $from$ is the node from which the packet arrived.

The source node s starts DFS coloring itself as gray and storing the id packet in its list (the $from$ field is empty). Each DFS packet has one bit that indicates whether the packet is forwarded or returned. When the current node c receives a packet for the first time, it adds a tuple $(id, from)$ into its memory and orders its neighbors according to their distance to the destination d (hence following a greedy method). The only node not to be taken into account in the ordered list is node $from$ that sent the packet to c. The packet is then forwarded to the first choice u among the neighbors (the first node chosen is the node that is closest to the destination). If there is no choice, the packet is returned to $from$.

If receiving a packet forwarded from any node b, a gray node c will reject it immediately, returning it to b (returned packet). A gray node b, upon receiving a returned packet from node c, will forward the packet to the next choice e in its sorted list of neighbors, if such a neighbor exists. If b has no more neighbors in its list, the packet will be returned to the node $from$, which originally sent the packet to b (memorized in the list of packets). An index L is used to know which is the next node to forward the packet in the ordered list. L is the index, in the list, of the last neighbor u selected for packet forwarding. When a new node has to be chosen, L is increased.

3.2.7 Summarizing

Most of the described protocols are proposed by works that assessed their performance using their own network simulator. In this respect, it is not easy to compare network performance among protocols, because of different simulators, test-beds, settings, etc. One of the aim of this dissertation is to provide a complete and common simulation scenario able to have a fair comparison. Table 3.1 summarizes the main characteristics about the considered stateless routing algorithms, as well as the simulation aspects of their proposal. Note that DFS is not included in this summary, since it is not a stateless strategy. In the following, we describe the meaning of the columns.

- **Loop Freedom**. A data packet repeatedly traversing the same data path. This endless process is stopped by employing a threshold value, necessary to terminate the packet traversal in the network.
- **Forwarding Method**. It indicates the forwarding strategy exploited to elect the next forwarder(s). There are four main schemes: progress, randomized, face and hybrid-based, the latter employing a combinations of the first three ones.
- **Simulator**. Every proposal considered in this work has performed a simulation of their protocol(s). This column reports the adopted simulator name or whether an custom one was considered.
- **Compared with**. This column indicates the other protocols with which the proposed protocol was compared in its original paper.

3.3 Routing Solutions in DTNs

In this section we explore the past research in terms of DTN routing. A brief description of classic DTN routing protocols is provided, as well as some related works that use location

Table 3.1: Summary of the considered position-based stateless routing algorithms. Note that some algorithm names are abbreviated.

Routing Protocol	Loop Free	Forw. Method	Simulator	Compared with
Greedy [86]	Yes	Prog-based	Custom	Compass, Ellipsoid, Most Forw., Proj. Face
Compass [88]	No	Prog-based	Custom	Greedy, Ellipsoid, Most Forw., Proj. Face
Most Forward [89]	Yes	Prog-based	Custom	Greedy, Compass, Ellipsoid, Proj. Face
Ellipsoid [90]	Yes	Prog-based	Custom	Greedy, Compass, Most Forw., Proj. Face
AB3D [95]	Yes	Rand-based	Custom	Greedy, Compass, Most Forw., AB3D, AB3D-LAR, LAR, Proj. Face
GRG [96]	Yes	Hybrid-based	Sinalgo	Flooding
Projective Face [99]	No	Face-based	Custom	Greedy, Compass, Ellipsoid, Most Forw.
CFace(3) [95]	No	Face-based	Custom	Greedy, Compasss, AB3D, AB3D-CFace(3), AB-CFace(1)-AB
ALSP face [100]	No	Face-based	Custom	Projective Face
GFG [97, 101]	No	Hybrid-based	Custom	Greedy, ALSP Face
AB-CFace(1)-AB [95]	No	Hybrid-based	Custom	Greedy, Compasss, AB3D, AB3D-CFace(3)
AB3D-CFace(3) [95]	No	Hybrid-based	Custom	Greedy, Compasss, AB3D, AB-CFace(1)-AB
LAR [104],	No	Prog-based	Custom	Compass, AB3D-LAR
AB3D-LAR [105, 104]	No	Prog/Rand-based	Custom	Compass, LAR

information to support the packet delivering in DTNs. Next, we discuss several works focused on geographic information in order to help the packet delivering.

3.3.1 Classic DTN routing protocols

Routing protocols for DTNs are classified into two categories: *replication-based protocols*, which ensure better delivery ratios allowing the packets to be duplicated in the network, and *forwarding-based protocols*, which never replicate the packets. In this section we describe some of the state-of-the-art DTN routing protocols, which we will consider for our performance evaluation.

A first example is *Epidemic* [107], a replication-based protocol, in which nodes continuously replicate and transmit the packets to newly discovered contacts. Epidemic is resource hungry, as it does not limit replications, in order to improve the chance to deliver the packet. This strategy is effective when the opportunistic encounters are purely random.

The *MaxProp* protocol [108] is a flooding-based technique, in which if a contact is discovered, all the packets attempt to be replicated and transmitted to that contact. The difference with Epidemic comes in determining which packets should be transmitted first and which ones should be dropped first. In other words, MaxProp maintains an ordered-queue based on the destination of each packet, ordered by an estimated likelihood of the path.

First Contact [109] is forwarding-based routing protocol in which each node transmits the current owning packet(s) to the first encountered node. The node that receives the packet(s) acts the same process, waiting for the first available contact. The process goes on until the packet(s) arrive(s) to the destination.

Spray and Wait [110] is a routing protocol that attempts to gain the delivery ratio benefits of replication-based routing as well as the low resource utilization benefits of forwarding-based routing. Spray and Wait achieves resource efficiency by setting a strict upper bound on the number of copies per packet allowed in the network. The Spray and Wait protocol is composed of two phases: the spray phase and the wait phase. When a new packet is created in the system, a number L is attached to that packet indicating the maximum allowable copies of the packet in the network. During the spray phase, the source of the packet is responsible for *spraying*, or delivery, one copy to L distinct *relays*. When a relay receives the copy, it enters the wait phase, where the relay simply holds that particular packet until the destination is encountered directly.

3.3.2 Geographic Information in DTN

A first example is AeroRP [111], designed for airborne telemetry applications. It checks the location and the current trajectory of nodes to take a more precise forwarding decision. AeroRP predicts the available connection with the recipient (base station) using this information, increasing the delivery ratio and reducing the overhead with respect to classic routing protocols like AODV and DSDV.

In [112], a protocol performance evaluation is performed considering a search and rescue scenario in the simulation configuration. Classic DTN routing protocols are evaluated for the comparison, and the authors conclude that contact opportunities represent useful information. Therefore, it is crucial to exploit these existing contacts, also considering context information such as node locations, movement directions, etc., in order to make better forwarding decisions.

The authors in [113] propose a method to optimize the Public Unmanned Aerial Vehicles (UAV-P) trajectory for delivery time minimization. UAV-P act as delivery devices, storing the data and delivering it to a base station, making this system a DTN. The work faces two problems: (a) finding a minimum number of coordinates where the UAVs need to reach all the information sources, and (b) determining the optimal route that connects these coordinates. Our proposal is different, since we do not act on the UAVs' trajectory; rather, we analyze the trajectory to choose the current best delivery UAV.

A delay tolerant application platform deployed on the Public Transportation System is proposed in [114]. The authors analyze the possibility to provide opportunistic connectivity using a carrier-based approach based on bus routes. Buses keep the users' data requests and transmit them when Internet access is available. The solution exploits the joint points between different bus lines, thanks to the availability of the buses' predetermined path and their time schedule. Simulation outcome shows good performance in several scenarios, making the proposal a practicable non real-time solution. A related and more detailed work is done in [115].

Some works focus on routing strategies that merge existing ad-hoc routing protocols with DTN characteristics. It is the case of [116], which adapts the AODV routing protocol for the DTN architecture. Basically, a new DTN-aware routing protocol is implemented on top of the underlying and unmodified AODV.

Authors of [117] present the possibility to make the UAVs' movement dynamic, based on the connectivity level between neighbor nodes. The objective is to change the position of UAVs in a DTN in order to establish the best communication quality in terms of signal-to-noise (SNR) ratio and packet throughput. The optimization algorithm orders the UAVs to move in a specific direction, so that the two metrics are minimized.

Chapter 4

Mobility Models

In this chapter, we describe and critically analyze the available mobility models for simulations, with particular reference to the FANET application scenarios discussed in 2.1.2.

When researches design and propose innovative communication protocols, these proposals employ simulations as a validation tool to analyze their performance metrics. When FANETs are considered, the requirements for a simulator used in this context are a sound modeling of realistic UAV movement, size and communication. Simulations that involve mobile nodes require a mobility model to represent how the nodes change their position while communicating, in order to analyze the network performance under mobility [118]. When flying objects are considered, a mobility model specifically designed for UAVs is needed. Such models are not yet well explored, since many researches ground their simulations on simpler mobility models, like the Random Waypoint model, which have actually been designed for traditional Mobile Ad-hoc Networks (MANETs). Nevertheless, the mobility of flying devices is very different from ground vehicles or other devices, due to the aerodynamic constraints. Hence, these mobility models fail to accurately reproduce the realistic behavior of UAVs, and could severely mislead the simulation outcome [119].

Realistic and synthetic mobility models for diverse application scenarios are discussed in [120]. In our work, we aim to give a critical evaluation of the available mobility models and provide requirements for each application scenario that involves a multi-UAV system, for example the need for an always connected network. By modeling these requirements in terms of UAV movement and coordination specifications (type of movement, speed, direction, etc.), we are able to associate the most feasible mobility models for any scenario, or to conclude that certain mobility models could not replicate particular realistic cases.

In our work, all the considered mobility models are implemented and performed in order to obtain the motion graphic representations presented in the following.

4.1 Mobiliy models state-of-the-art

We can divide the existing mobility models of an UAV-based network into five classes:

- **Pure Randomized mobility models.** The mobility of UAVs is randomized in terms of direction, speed, and time of movement.
- **Time Dependent mobility models.** The mobility of UAVs depends on the previous speed and direction.
- **Path-planned mobility models.** Each UAV follows a pre-planned path, without taking any random direction.
- **Group mobility models.** UAVs movement is constrained by a reference point (or more reference points). The UAVs move randomly within a defined area centered at the reference point.
- **Topology-control based mobility models.** UAVs fly over an area being aware of their own topology, coordinating their position among themselves. This is needed, for example, when we have network connectivity constraints.

Pure Randomized Mobility Models

Randomized mobility models are the common models for network research. They represent multiple mobile nodes whose actions are completely independent of each other.

Random Walk. The *Random Walk* (RW) mobility model was designed considering the unpredictable movement of many entities in nature. It refers to the Brownian motion described mathematically by Einstein in 1926. In this mobility model, the mobile nodes simulate this irregular movement choosing, every time, a random direction between $[0, 2\pi]$, and a random speed between $[s_{min}, s_{max}]$. Each movement occurs in either a constant time interval or a constant traveled distance, at the end of which a new direction and speed are computed. If a node bounces to the border of simulation area, the new direction is calculated according to the incoming direction. RW is a memory-less mobility model, since it does not store the knowledge of its past locations and speeds. An example of a RW motion is shown in Fig. 4.1. The mobile node starts from the center of the 300m x 300m simulation area. Then, it randomly chooses a speed between $[10m/s, 30m/s]$ and a direction between $[0, 2\pi]$. The mobile node moves for 10 seconds, after which a new speed and direction are chosen. The figure also illustrates the mobile node's speed, represented by the point density on the trace: the less the density, the higher the speed (in fact, each point is the position of the mobile node at each time step).

RW is very simple to model and implement, due to its basic mathematical characteristics. However, it simulates unrealistic movement patterns due to ignoring many details of realistic flying movements. We can note frequent sudden stops, changes of speed, and changes of direction (see Fig. 4.1).

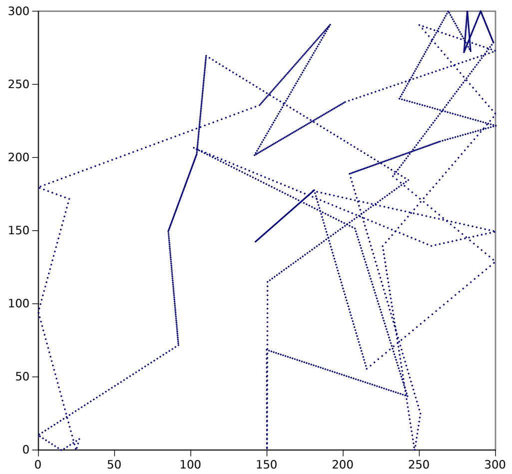


Figure 4.1: Pattern of a mobile node using the Random Walk mobility model.

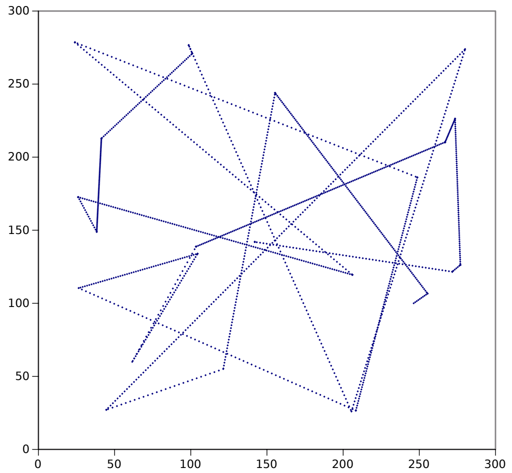


Figure 4.2: Pattern of a mobile node using the Random Waypoint mobility model.

Random Waypoint. The *Random Waypoint* (RWP) mobility model [121] behaves in the same way as RW, but with a few differences. A mobile node starts by being stationary for a certain period of time (i.e., a pause time). Once this time expires, the mobile node chooses a random destination in the simulation area, and a speed that is uniformly distributed between $[s_{min}, s_{max}]$. The mobile node then travels toward the newly chosen destination point at the selected speed. Upon arrival, the mobile node pauses for a specified time period before starting the process again. The difference from RW is the inclusion of a pause time and the random selection of a destination, rather than a direction selection. Fig. 4.2 shows an example of a mobility pattern using RWP, where the mobile node begins from the center of the simulation area. For each destination point reached, the mobile node chooses a new speed between $[10m/s, 30m/s]$. We can notice that the motion pattern is concentrated at the center of the simulation area.

Despite the presence of pause times helps at smoothing sudden changes of direction, both stops and movement starts remain sudden, with infinite acceleration. UAVs (like any other physical object) must have a progressive increase or decrease of speed.

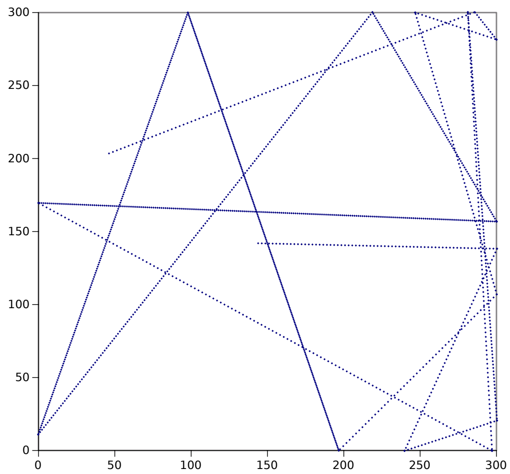


Figure 4.3: Pattern of a mobile node using the Random Direction mobility model.

Random Direction. The *Random Direction* (RD) mobility model [122] was created to face the issue of concentration of nodes in the center part of simulation area in the RWP mobility model, due to the high probability of moving towards a new destination near the middle of the simulation area. With RD, each mobile node selects a destination point on the edge of the simulation area. When it arrives at the edge, it pauses for a time and then again selects another random destination point at the edge. Fig. 4.3 shows this difference from RWP.

Even in RD we have the same unrealistic motion characteristics detected for RW and RWP.

Manhattan Grid. The *Manhattan Grid* (MG) mobility model [123] uses a grid road topology (see Fig. 4.4). This mobility model was mainly proposed for describing the movement in an urban area, where the street layout is very regular. In this mobility model, the mobile nodes move in horizontal or vertical directions on an urban map. The MG model employs a probabilistic approach in the selection of node movements, since, at each intersection, a vehicle chooses whether to keep moving in the same direction or to turn. The probability of going straight is 0.5, and taking a left or right turn is 0.25 each.

For MG it is very clear that, for several UAV scenarios, this model is not suitable. Although this model provides flexibility for the nodes to change the direction, it imposes geographic restrictions on nodes' mobility. The same considerations for the first three mobility models can also be taken in MG considering the sharp turns and speed.

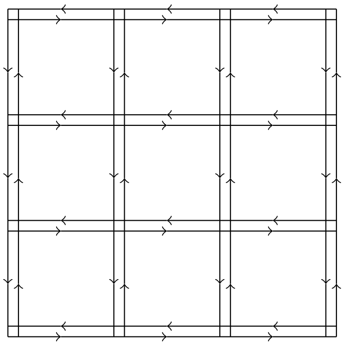


Figure 4.4: An example of a Manhattan Grid topology with double-way roads.

Time Dependent mobility model

This category of mobility models tries to avoid sharp speed and sharp direction changes. The smooth change of motion can be performed using different mathematical equations.

Boundless Simulation Area. The *Boundless Simulation Area* (BSA) mobility model [124] uses a relationship between the previous direction and speed and the current ones. For each mobile node, through a vector $v = (v, \theta)$, which represents the mobile node's speed and direction, the speed and direction values are updated at every time step Δt using the following equations:

$$v(t + \Delta t) = min[max(v(t) + \Delta v, 0), V_{max}];$$

$$\theta(t + \Delta t) = \theta(t) + \Delta\theta;$$

where V_{max} is the maximum speed allowed, Δv is the speed change uniformly chosen between $[-A_{max} * \Delta t, A_{max} * \Delta t]$, A_{max} is the maximum acceleration allowed, $\Delta\theta$ is the direction change uniformly chosen between $[-\alpha * \Delta t, \alpha * \Delta t]$, and α is the maximum angle permitted. With the new values, the mobile node's position can be also updated:

$$x(t + \Delta t) = x(t) + v(t) * cos\theta(t);$$

$$y(t + \Delta t) = y(t) + v(t) * sin\theta(t);$$

The movement into the simulation area boundary is handled differently from the other mobility models: when a mobile node reaches one side of the simulation area, it continues to travel and reappears on the opposite side of the simulation area. An example of the BSA movement is shown in Fig. 4.5, where V_{max} is 30 m/s, A_{max} is $10m/s^2$, α is 90 degrees, and Δt is 0.1 seconds. As we can see, the point density changes smoothly, which is due to the the smooth change of speed.

BSA allows the mobile nodes to move unobstructed in the simulation area, removing any edge effects on the simulation evaluation. However, several simulation contexts could find undesirable the side effects that occur from the moving around a torus. In applications where the mission area is simply a 2D shape and the study of relationship between mobile nodes is prominent, this model would not meet the necessary conditions because of the teleportation effect inherent to the model [125].

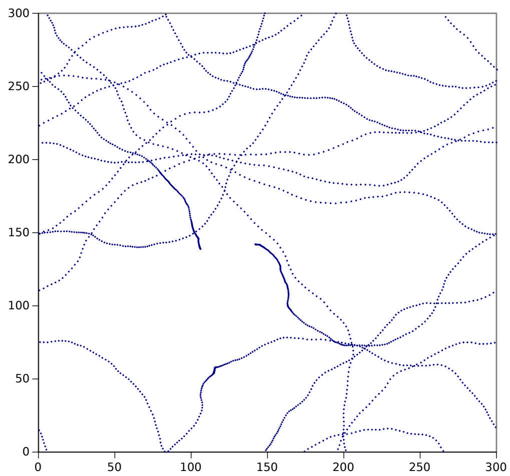


Figure 4.5: Pattern of a mobile node using the Boundless Simulation Area mobility model.

Gauss-Markov. The *Gauss-Markov* (GM) mobility model [126] uses several parameters that are tuned to adapt the model to different levels of randomness. At start, each mobile node is set with a current speed and direction. Then, at each time instance n^{th}, the new speed $s(n)$ and direction $d(n)$ are updated using the speed and direction values at the $(n-1)^{th}$ instance, according to the following equations:

$$s(n) = \alpha s_{n-1} + (1-\alpha)\bar{s} + \sqrt{(1-\alpha^2)} s_{x_{n-1}}$$

$$d(n) = \alpha d_{n-1} + (1-\alpha)\bar{d} + \sqrt{(1-\alpha^2)} d_{x_{n-1}}$$

where α, which can assume a value between $[0,1]$, is the tuning parameter used to change the randomness, $\bar{s}$ and $\bar{d}$ are the mean value of speed and direction respectively, and $s_{x_{n-1}}$ and $d_{x_{n-1}}$ are random variables from a Gaussian distribution. When $\alpha = 0$, we obtain a completely random motion of the mobile node, and when $\alpha = 1$ we have a linear motion. By accurately setting the value of α between 0 and 1, we can obtain the specific desired motion behavior. Furthermore, the location (x_n, y_n) at the n^{th} time instance is calculated based on the previous location (x_{n-1}, y_{n-1}), speed s_{n-1} and direction d_{n-1} of movement:

$$x_n = x_{n-1} + s_{n-1} \cos d_{n-1}$$

$$y_n = y_{n-1} + s_{n-1} \sin d_{n-1}$$

Fig. 4.6 illustrates an instance of the GM mobility model, with $n = 1$ second, $\alpha = 0.8$, $\bar{s} = 10m/s$, $\bar{d} = 0$ degrees, and where $s_{x_{n-1}}$ is chosen from a random Gaussian distribution with mean equal to 0 and standard deviation equal to $1m/s$, and $d_{x_{n-1}}$ is chosen from a random Gaussian distribution with mean equal to 0 and standard deviation equal to 45 degrees.

Other versions of the Gauss-Markov mobility model were proposed; they modify the original equations with other variables, functions and constants. In addition, a 3D version of GM is defined in [127].

GM is able to reduce the sudden stops and sharp turns that we can notice in RW, RWP and RD. This is possible thanks to the equations system that relates the past speed and direction with the future ones, allowing a smooth update if appropriate parameters are chosen. However, this model cannot reproduce a typical behavior of UAVs (especially turns), and it is difficult to choose good parameters to make their movement more realistic.

Smooth-Turn. The *Smooth Turn* (ST) mobility model [128] allows the mobile nodes to move in flexible trajectories, correlating the acceleration of the mobile node across temporal and spatial coordinates. With this mobility model, each UAV chooses a point

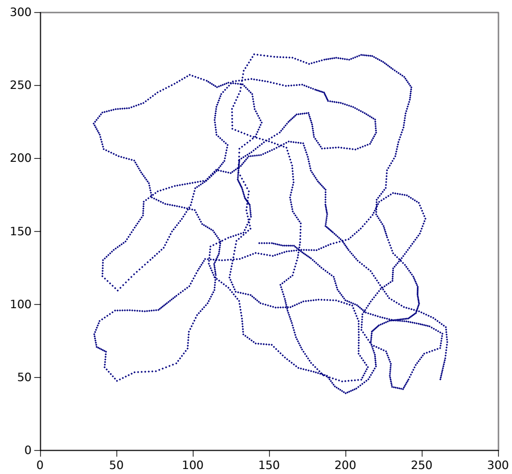


Figure 4.6: Pattern of a mobile node using the Gauss Markov mobility model.

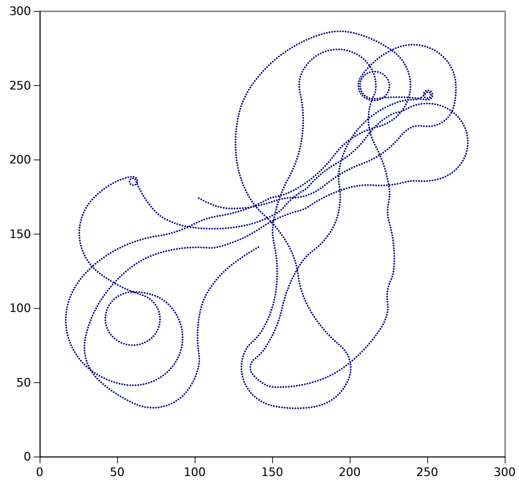


Figure 4.7: Pattern of a mobile node using the Smooth Turn mobility model.

in the space and then circles around it until the UAV selects another turning point. The chosen point must be perpendicular to the UAV direction, to ensure a smooth trajectory. The waiting time interval (i.e., the duration for the aircraft to circle around the current turn center) is modeled to be exponential distributed. The pattern performed with this mobility model can be seen in Fig. 4.7, where an average speed of $15m/s$ is assumed, and the turn radius is uniformly chosen between $10m$ and $100m$.

Unlike traditional mobility models which force vehicles to make sharp turns, ST realistically captures the smooth movement patterns of airborne vehicles without placing additional constraints. We have noticed some drawbacks: the lack of any collision avoidance mechanism, and boundary reflection effects due to the impact of the mobile node on the area boundaries, which forces the node to suddenly change its direction.

Path-planned mobility models

These mobility models deploy a certain predefined path in order to force the UAVs to follow it. Especially, each UAV follows a specific pattern until it arrives at the end; at this point the UAV randomly changes the pattern or repeats the same one.

Semi-Random Circular Movement. The *Semi-Random Circular Movement* (SRCM) mobility model [129] is designed for the curved movement scenarios of UAVs. It is suitable for simulating UAVs turning around a specific position in order to gather some information. In the SRCM model, each node n_i starts from an initial point $P_i^{(0)}$ on the circle $C_i^{(0)}$, rotating with a speed v_i uniformly distributed in $[v_{min}, v_{max}]$ about the center of the circle. The mobile node moves towards the first destination point $P_i^{(1)}$ on the same circle. Then, the node remains stationary for a time pt_i, and then it starts moving to a second destination point $P_i^{(2)}$, and so on. When the node completes an entire round on the circle, it randomly chooses another circle, moving in a straight line on it, and repeating the process explained before. Fig. 4.8 illustrates an example of the SRCM mobility model behavior.

A particular strength of SRCM is the reduction of potential collisions between UAVs due to the circular nature of the movement, compared with any trivial random mobility model. However, the change of the turn radius leads to an orthogonal sudden change of direction, being clearly not realistic from the UAV perspective.

Paparazzi The *Paparazzi* mobility model (PPRZM) [130] is a stochastic mobility model that imitates Paparazzi UAV behavior in the Paparazzi autopilot system [131]. It is based on a state machine containing six movement pattern states: Stay-At, Eight, Oval, Scan and Way-Point. At the beginning, each UAV chooses a movement type, its start position and the speed. Then, it chooses and fixes a random altitude, which remains constant for the entire simulation. Fig. 4.9 shows the different movement patterns of PPRZM. In [132], a study of several routing protocols using PPRZM is provided.

PPRZM tries to face the problems of pure randomized mobility models providing a set of well defined pattern schemes that simulate some real UAV movements. Authors in [130] sustain that these pattern schemes are closer to the behavior of real application scenario traces, compared to a simple RWP. On the other hand, the simple state machine, through which the movement pattern for each UAV is chosen, is too casual for a specific

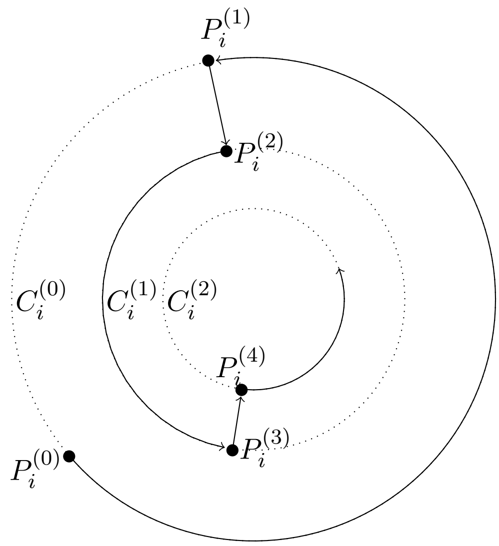


Figure 4.8: An example of pattern traveling performed by a node i with the SRCM mobility model.

mission planning. Initially, the UAVs' positions are chosen randomly, and lack a specific organization among their pattern movements. Furthermore, remains unclear how a node can switch from a pattern scheme to another one: each scheme change could cause a sudden change in the UAV's direction.

Group Mobility Models

Group mobility models include a spatial constraint among all the mobile nodes. In the previous sections, all the mobility models simulate the behavior of mobile nodes that are completely independent of each other. However, in FANETs, there are many situations where it is necessary that UAVs move together following a common point, introducing a spatial dependency between them. For example, a group of UAVs can follow a common line that moves along with time, in order to patrol a specific large area. Typically, this group formation is planned to accomplish a common goal. The Reference Point Group mobility (RPGM) [133] model simulates a random motion of mobile nodes within a reference group. The reference group, represented by its logical center, moves on the area by following a simple RWP model. Its motion consequently characterizes the speed and direction of its corresponding group of mobile nodes. In particular, at each time step, the location of each mobile node is updated according to the logical center of the reference group; then, the random vector is combined with the updated location, in order to diversify the behavior of all the group's nodes. In the following, three mobility models are described as special

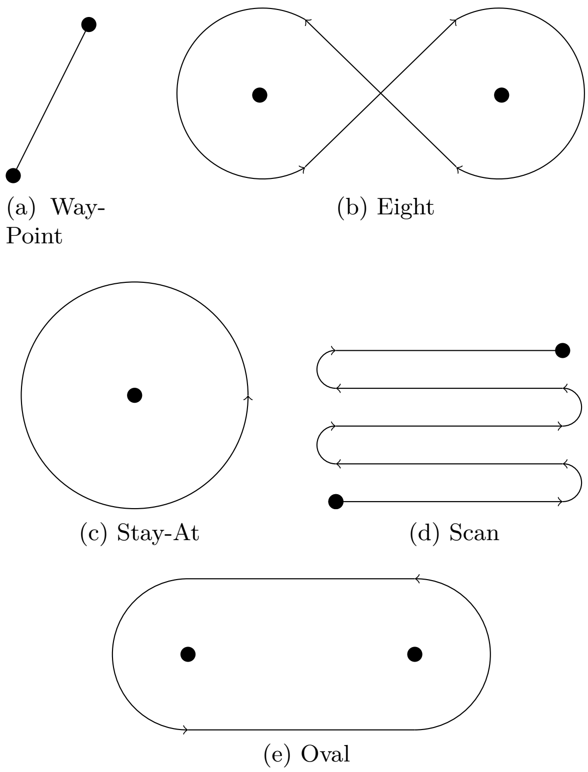


Figure 4.9: All the path schemes defined in the PPRZM mobility model.

cases of the general RPGM.

Column mobility model. In [134], the *Column* (CLMN) mobility model is proposed for scanning or target searching application scenarios. Each mobile node moves around a reference point placed on a given line, which is moving in a forward direction. In particular, each mobile node randomly turns around the reference point through an entity mobility model, for instance a simple RW. Fig. 4.10 illustrates an example of CLMN, where three mobile nodes randomly move around their own reference point placed on a line, which moves and turns. At each time interval, the new position of each reference point is calculated taking into account the old reference point's position and the advance vector of the line, identified by a random distance π and a random angle θ. In Fig. 4.11, an example of three nodes that move according to CLMN is shown.

The Group mobility models could represent many realistic scenarios involving UAVs. The spatial constraint defined by these models can allow a guaranteed connection among the UAVs within each group. CLMN can prevent collision between UAVs, since each UAV moves around a fixed point, being these points placed far from each other. However, smooth turns and speed changes are not present in this mobility model.

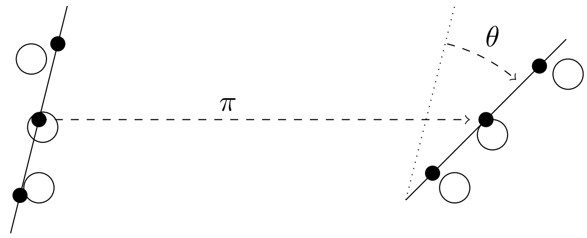


Figure 4.10: Example of a single step movement using CLMN, in which a set of three reference points are fixed to a line, which moves for π and turns for θ.

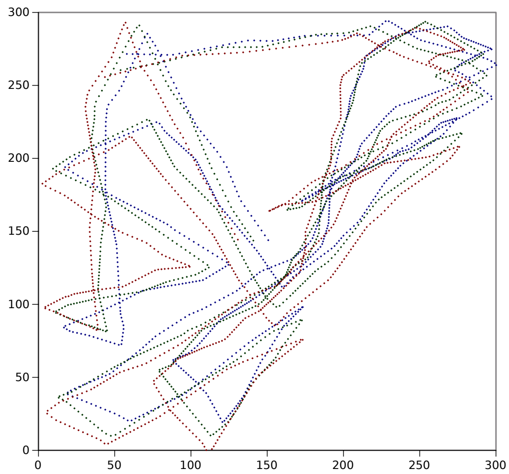


Figure 4.11: Pattern of three mobile nodes using the Column mobility model.

Nomadic Community mobility model. In the *Nomadic Community* (NC) mobility model, each mobile node moves randomly around a given reference point, without any constraint (as in CLMN). Fig. 4.12 gives an illustration of five nodes moving around a reference point, which move, in turn, in a certain direction. The mobile nodes can move within a maximum distance r_{max} from the reference point. At each time interval, the reference point also moves for a certain distance π using a random mobility model (e.g., RWP). Differently from CLMN, the nodes in the group share the same space defined by the unique reference point, rather than an individual reference point in a column. In Fig. 4.13, an example of three nodes that move according to NC is shown.

NC has nodes that share common spaces, producing collision events between UAVs. Modified versions of this mobility model can include additional constraints in order to initially divide the flying spaces, or to control the distance between the UAV pairs, in order to avoid collisions. Furthermore, even here we have sudden movements due to the changing direction and speed of mobile nodes and reference points. As in CLMN, smooth turns and speed changes are not taken into account.

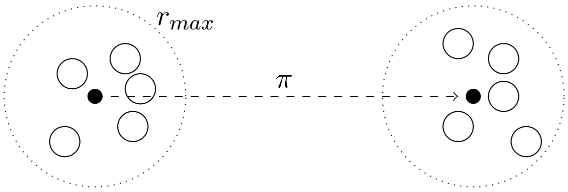


Figure 4.12: Example of a single step movement using NC, in which five mobile nodes move around a moving reference point.

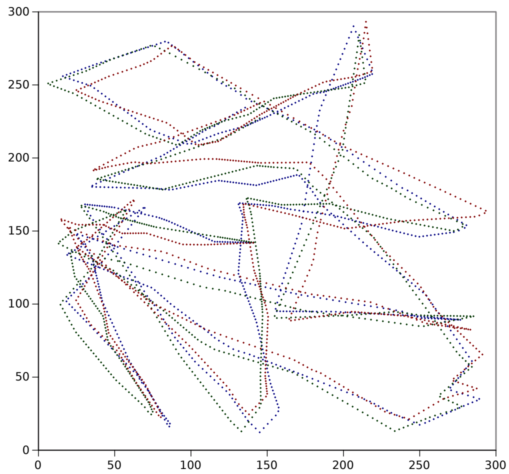


Figure 4.13: Pattern of three mobile nodes using the Nomadic mobility model.

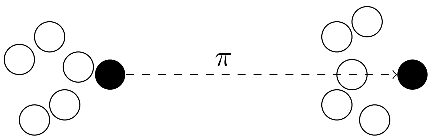


Figure 4.14: Example of a single step movement using PRS, in which five mobile nodes follow the moving black pursued target.

Pursue Mobility model. The *Pursue* (PRS) mobility model is similar to NC. In this case, the mobile nodes attempt to follow a particular target that moves in a certain direction. The mobile nodes use a simple random relative motion while pursuing the target.

We can imagine, for example, police officers that try to follow and catch escaped criminals. Fig. 4.14 shows an example of PRS movement, where five nodes (in white) represent the pursuing nodes that follow the pursued node (in black). PRS is very similar to the NC, hence it shares the same characteristics.

Topology-control based mobility models

A real-time control of UAVs' motion is needed when certain network or mission constraints have to be continuously satisfied. One constraint is to maintain a fully connected network of UAVs at all times, so that a given UAV can talk with any other. In this case, a continuous motion control of UAVs according to the network connectivity level is mandatory. *Topology-control* mobility models are the new generation of mobility models for FANETs, since they enable network topology control through sensible data transmission in the network itself. In this way, the randomized nature of mobility is removed, being replaced by a control solution that is more robust and aware of the UAVs movement according to the mission objectives or network constraints. In [135], the development of a distributed mobility model for UAVs considering connectivity and area coverage is proposed. We present some existing typical swarm models proposed in the literature that could be applied for deploying and controlling groups of UAVs.

Distributed Pheromone Repel. In [39], the *Distributed Pheromone Repel* (DPR) mobility model is proposed, with the aim of describing a robust and random movement of UAVs that perform reconnaissance to detect hostile ground targets. Each mobile node maintains its own pheromone map, which is a grid of the area where each cell contains a timestamp representing the last time the cell was scanned. Once a mobile node scans an area, it marks that area on the map. The information of scanned areas (the local pheromone maps) is regularly broadcast among the mobile nodes. When a mobile node has to move, it can go straight, turn left or turn right, according to its pheromone map.

DPR is able to reproduce a realistic movement thanks to the smooth turns and the objective pheromone map. The problem is that the network connectivity is not considered. An evolution of this mobility model can be the enhancement of the pheromone map including a connection density with the other UAVs, to counterpoise the target pheromone map for the mission.

Self-deployable Points Coverage. A self-organizing flying ad-hoc network is proposed in [136] for disaster scenarios. The *Self-deployable Point Coverage* (SDPC) mobility model deploys a set of UAVs on a disaster area in order to create a communication infrastructure that the victims of the disaster event can use. The aim of each UAV is to cover the maximum number of people on the ground maintaining a connection with the other UAVs.

SDPC is a good compromise between target coverage and connectivity, since the UAVs try to reach the target victims while also taking into consideration the connectivity between themselves. However, the simulation of the proposed model is performed without a realistic movement (smooth acceleration and turns) of UAVs. Although an explicit collision detection procedure is not integrated in this model, own functioning prevents UAVs from approaching each other, since they tend to move so as to cover the maximum possible area. On the other hand, a building/obstacle collision control could be necessary.

Pollution-driven UAV Control. For pollution sensing, the Pollution-driven UAV Control (PdUC) mobility model is proposed [137], allowing UAVs to autonomusly trace pollutant sources and monitor air quality in the area. The problem is that the proposed mobility model is extremely slow, also considering the battery lifetime of UAVs. An improvement is provided from PdUC-Discretized, which optimizes the process of monitoring locations.

4.2 Motivation of realistic mobility models

Generally, before simulating a network scenario, a communication model, the mobility pattern of nodes and other parameters are set. In particular, the specific mobility model adopted by nodes can significantly affect the performance of a simulated aspect of the network, like routing performance [125, 119]. Many research works analyze protocol performance using very trivial mobility models, like the Random Waypoint or the Random Walk models. Nevertheless, in the context of UAV networks, they could not fully reproduce the actual behavior of UAVs. Consequently, network simulations might show incorrect results with respect to those that would be obtained under real conditions. For example, the performance of a particular routing protocol for a specific UAV application scenario can be poor when using the Random Waypoint mobility model, but it could be very good when using a mobility model specifically designed for that application scenario. When flying objects are considered, a realistic mobility model specifically designed for UAVs is needed to effectively highlight possible issues. A significant requirement to get a realistic UAV motion is to have smooth trajectories, due to the aerodynamic conditions. In [138], the authors propose a grid model for UAV path planning that also considers smooth curves.

The work in [129] introduces several reasons to use realistic mobility models rather than simple random ones:

- The mobility model is an important factor affecting the performance of a FANET, so it is important to establish a realistic movement pattern.

- The distribution of node location impacts upon many network characteristics such as network connectivity, average path length and network capacity.

- Random patterns may be very different from actual movement patterns found in the real world, and the simulation results obtained based on these models exhibit obvious differences from more realistic scenarios.

- Typically, the network connectivity varies because the distributions of node location and speed vary over the simulation time horizon. A non-uniform distribution reduces the applicability of existing analytical results, which are typically based on the uniformity assumption.

- Finally, these models are almost always based on simple straight-line patterns which differ from many real world scenarios. UAVs usually mimic the movement of flocks of birds, or shoals of fish. In this case, they should move along continuous, curved trajectories rather than in a simple straight line.

4.3 Representativeness of Existing Mobility Models

When flying devices are considered, we have to define specific characteristics about their movement. Due to the nature of the aerodynamic and mechanical flight, the motion behavior of UAVs is difficult to be recreated on simulators. The motion of an UAV is determined by several factors, such as the path, the speed, the atmospheric condition, etc. Trivial mobility models, like Random Walk, do not take these issues into consideration. A first aspect is the curve of an UAV, which is reproduced by many simple mobility models as a generic sudden change of direction, compared with a realistic curve, which is smooth. Then, a realistic flying device has an acceleration and deceleration while changing its speed. Finally, it could have micro variations in its motion due to turbulence or speed differences between the different propellers. These characteristics are not simulated in several mobility models, which simply change the speed with an infinite acceleration. Finally, safety requirements, like safety distance to avoid collisions should be taken into consideration. In Table 4.1, we illustrate these characteristics and indicate which mobility models can satisfy them. Table 4.2 shows a list of the available mobility model generators and network simulators, including all the mobility models supported by each of them.

Table 4.1: Mobility models and their related realistic and network characteristics.

Mobility Model	Class	Smooth Curve	Smooth Acceleration	Micro Variation	Connectivity Awareness	Collision Avoidance
RW [134]	Randomized	No	No	No	No	No
RWP [121]	Randomized	No	No	No	No	No
RD [122]	Randomized	No	No	No	No	No
MG [123]	Randomized	No	No	No	No	No
BSA [124]	Time-dependent	Yes	Yes	Yes	No	No
GM [126]	Time-dependent	No	Yes	Yes	No	No
ST [128]	Time-dependent	Yes	No	No	No	No
SRCM [129]	Path-planned	Partially	No	No	No	Partially
PPRZM [130]	Path-planned	Partially	No	No	No	No
CLMN [134]	Group	No	No	Yes	Yes	Partially
NC [134]	Group	No	No	Yes	Yes	No
PRS [134]	Group	No	No	Yes	Yes	No
DPR [39]	Topology-control	Yes	No	No	No	No
SDPC [136]	Topology-control	No	No	No	Yes	Partially

Table 4.2: Networks simulators and mobility generators with implemented mobility models (N.A: information Not Available).

Software Application	**Type**	**Mobility Models Implemented**
BonnMotion	Mobility Generator	RW, RWP, GM, MG, RPGM
MobiSim	Mobility Generator	RW, RWP, RD, GM, MG, NC, PRS
NS 2	Network Simulator	Provided by BonnMotion
NS 3	Network Simulator	Provided by BonnMotion, RW, RWP, RD, GM, MG, RPGM
Omnet++	Network Simulator	RWP (provided by INET)
Opnet	Network Simulator	RW, RWP, RD, Group mobility
NetSim	Network Simulator	RW, RWP
GloMoSim/QualNet	Network Simulator	Provided by BonnMotion, RWP, Group
SSFNet	Network Simulator	RW, RWP, GM, MG, RPGM
J-Sim	Network Simulator	RWP
YANS	Network Simulator	N.A.
GTNetS	Network Simulator	RWP

4.4 Adoption of Mobility Models to Application Scenarios

In this section, we illustrate which mobility models can be adequate for the above defined application scenarios and their special cases. For each FANET application scenario, we analyzes their specific characteristics (requisites) and go to define which mobility models may be applied in the considered scenario (solutions), along with their limitations in terms of realism, connectivity and collision avoidance.

In Table 4.3, a summary of application scenarios and feasible mobility models associations is shown. Random Walk, Random Direction, Random Waypoint are generic and too simple mobility models that are based on stochastic movement. They do not reflect any particular kind of movement to reach any task. The nodes simply move randomly, with a linear trajectory, stopping suddenly and then restarting the movement. There is no acceleration model nor a smooth turn model. Every one of these models can be used when there is no need to consider a particular application, but to analyze network performance on a generic MANET, with a certain tolerance. In Figure 4.15, we show an example that allows comparing speed variations throughout time for the standard RWP model, and the equivalent speed changes under traces obtained using the SITL simulator [139], which are more realistic. We can see a huge difference between the two speeds, and, as mentioned, we notice the sharp speed changes in the case of RWP. Hence, in Table 4.3, these three models are not taken into consideration.

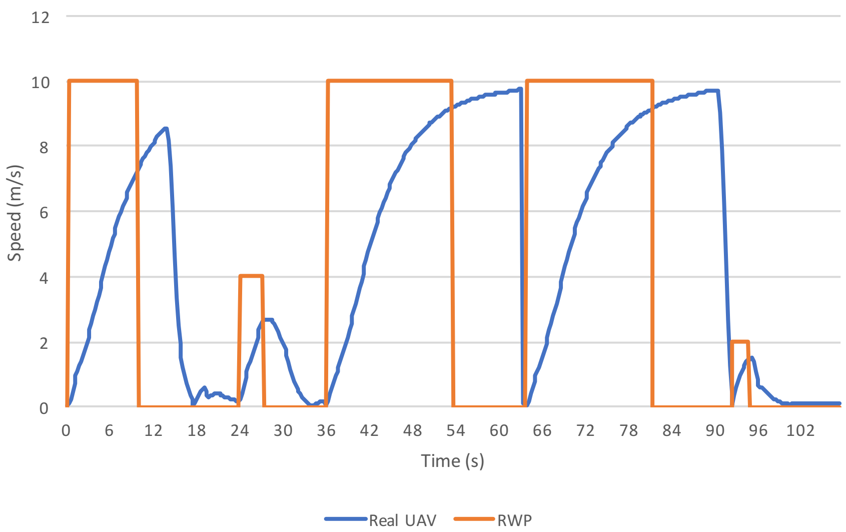


Figure 4.15: Speed variation differences between a real UAV and a simulated UAV using the RWP mobility model.

4.4.1 Search and rescue

Application requisites. Due to the large number of contexts where search operations can be deployed, the mobility behavior of UAVs strongly depends on the type of infrastructure to employ. Typically, the search area is predetermined at the start of the search operation, defining, for each UAV, an available path. If the operation is monitored by a base station, UAVs have to be always connected to it, forming a relaying chain; hence, all the paths need to be designed so that the network results connected all the time. If the search operation is performed in a mountain region, several obstacles have to be considered in the planning of the rescue mission. In emergency scenarios the rescue base needs real-time information about the area (video and audio); this imposes some constraints on the networks created between UAVs, like the connectivity levels that are able to guarantee a minimum throughput and delay.

A typical pattern for search and rescue operations is a simple scan scheme derived from PPRZM, when a rectangular search area is defined. Using multiple UAVs to speed up the task completion, it is possible to plan a multiple search scheme based on scanning. A requirement for this type of solution is UAV connectivity: the scan scheme of each UAV has to be planned according to the others, in order to maintain a connected UAV network as much as possible. Another pattern is a circle scheme, like SRCM, which restricts UAVs to turn around a fixed target as a potential search target location.

BSA, GM or PPRZM can reproduce a randomized search task in a well defined area. BSA has the unrealistic teleportation feature when an UAV gets out from the area. To tackle this problem, when the UAV goes out the area, each UAV needs to delay the time to re-enter. The randomized nature of mobility models brings the problem of connectivity and collisions. In that case, a study of network connectivity compared to the network node density is needed, especially in emergency search scenarios, which requires a guaranteed delivery and a maximum delay limit.

A more robust mobility model for search and rescue scenarios is the DPR mobility model, in which the movements depend on the areas visited by UAVs. In this case, UAVs need a strong connectivity in order to guarantee a good packet delivery for the coordination between UAVs.

In a disaster scenario, occurring for example after an earthquake event, eventual victims are sparsly distributed on the ground. SDPC is the most useful mobility model specifically designed for these events, because it allows the UAVs to reach the victims taking into consideration their connectivity.

4.4.2 Forest Fire detection

Application requisites. Generally, UAVs act as sensor nodes and keep stationary or move slowly around the areas of interest, which could be the areas with greater probability of fire emergence. When a fire is detected by a UAV, the other UAVs should change formation, through a cooperation system, and move to the fire fringe to monitor it. Even here the connectivity between UAVs is a requisite, since a fire event has to trigger an alarm to all the UAVs, which then approach the target area.

Table 4.3: Mobility models feasibility for FANET application scenarios

Application Class	**Mobility Model**	**Scenario Description**
Search and Rescue Operations	BSA, GM, ST	Random search on a specified target area.
	PPRZM	Each UAV selects the scan pattern in random position.
	SRCM	Scanning in a circular area.
	DPR	Scanning an area through repeated checks.
	SDPC	Reaching victims on a disaster area.
Traffic and Urban Monitoring	Static	UAVs as fixed cameras at crossroads.
	MG	Surveillance of city streets.
	SRCM	Patrolling of a crash event before the rescue team arrives.
Reconnaissance and Patrolling	Static	Static first line of defense and patrol.
	SRCM	Surveillance of a target.
	BSA, GM	Missions without path prediction by adversaries.
	PRS	Pursuing of a critical moving target.
	DPR	Real-time missions with awareness of critical areas.
Agricultural Management	CLMN	Field condition checking.
	PPRZM	UAV actions (e.g., irrigation) on cultivated fields.
Environmental Sensing	Static	UAVs as stationary sensor nodes.
	PPRZM	UAVs follow some predefined paths that cover several sensors.
Relaying Network	Static	Static UAV communication infrastructure.
	MG	V2V connectivity among urban vehicles.

Solutions. A simple scheme for UAVs in detecting forest fires is SRCM, where UAVs turn around a forest area with different radius. However, DPR can be used by UAVs that are equipped with specific sensors. They can fly above areas that present particular physical quantities (temperature, humidity, infrared signals etc.), upgrading their pheromone maps according to these data, and moving towards more affected areas.

Another scenario to consider is the case in which a fire is ongoing, and the firefighters are proceeding to extinguish it; a swarm of UAVs can move on the fire edge to monitor and follow its progress. In this case, a group mobility model, in particular CLMN, can be used and adapted to the fire edge.

4.4.3 Traffic and urban monitoring

Application requisites. In this case, the use of UAV-to-infrastructure system in infeasible, since generally the urban environment contains many tall buildings and other obstacles, eliminating the line-of-sight. Hence, a relay-tree UAV system has to be designed so that each UAV can rely upon at least another UAV to send the information to the control base station. As in [36], an optimization problem is required to minimize the number of UAVs needed to build the surveillance system to a specific number of targets. Real-time video is a requirement for this application scenario.

Solutions. For urban scenarios we could have different applications. If we need a surveillance infrastructure on crossroads, UAVs can be positioned stationary as fixed cameras on each crossroads to monitor some events; in this case a simple Static model can be used.

If we need moving UAVs to turn around the urban area, an MG mobility model could be adequate for this purpose. In this case, most of the attention focuses on the connectivity among UAVs, since MG does not support topology control, and for real-time surveillance it can be a critical point to resolve; a possible solution comes from the introduction of other UAVs with a topology control mechanism that can act as relays for urban monitoring through UAVs.

Another scenario that can represent a more specific event is a car accident. A group of UAVs may reach the area before help arrives, in order to check the victims state and detect any dangerous situations. SRCM may be suitable for this type of scenario.

4.4.4 Reconnaissance and patrolling

Application requisites. UAVs in reconnaissance and patrolling scenarios should not have a predictable path, otherwise their positions would be known by potential enemies. Thus, some randomness should be applied, even when guaranteeing a minimum interval time in patrolling a certain area or group of targets. If a cooperation between UAVs to communicate their own covered sectors is possible, the movement of each UAV could change according to the information received by the other UAVs. In [140], an experimental comparison of different mobility models for reconnaissance operations is performed, offering several considerations about the realism of the models.

Solutions. First line of defense patrol is a typical usage of UAVs that fly in a stationary mode to oversee a specific area.

UAVs can also replace people or cars patrolling periodically around a specified target (buildings, people, ground vehicles, etc.); when a mission needs of such type of target surveillance, SRCM may be used. When these targets also move, a group mobility model can satisfy a simulation of this behavior. The specific group mobility model adopted (CLMN, NC or PRS) depends on the target number and behavior. For instance, if a group of ground vehicles proceed along a street, UAV movement can be simulated with CLMN, or PRS, if the target must be pursued by UAVs.

For reconnaissance missions over a specified area, simple random motions can be used. In fight-flight principles, each UAV travels to the involved area, and then gets out by returning to the base station. BSA or GM are good mobility models for these applications since they produce random paths not predictable by enemies.

For missions in which a certain cooperation among UAVs is needed, the DPR mobility model includes a map of attraction points that can represent sensible areas (people concentration, lively areas, areas at risks, etc.).

4.4.5 Agricultural management

Application requisites. UAVs used in agricultural contexts are typically few and have a pre-planned path on the area to be examined. There are no particular real-time or time requirements for agricultural applications: UAVs can move on a predefined path and return to the base when the mission (surveillance, irrigation or disinfection) is ended.

Solutions. Typically, agricultural scenarios require a one-time movement. For actuator UAVs (performing irrigation or disinfection tasks) a scan pattern from PPRZM can be adequate, due to the rectangular nature of the operation areas like cultivated fields. Alternatively, CLMN can be appropriate to check the field area conditions.

4.4.6 Environmental sensing

Application requisites. Generally, UAVs only move to reach their planned positions on the sky or near the ground. Then, they remain stationary to sense a specific aspect of the environment. Alternatively, they can move according to a predefined scheme. Transmission of data through the FANET is not always required: UAVs can detect environment data from their sensors and store them; when they return to the base station, data can be downloaded in just a few seconds. Instead, when real-time sensing is needed, UAVs have to be connected with each other. The sensed data typically contains information about the environment (temperature, humidity, pollution, etc.) and not audio/video. Therefore, there is no need for high throughput.

Solutions. UAVs as sensor nodes are placed stationary on the area to be sensed, periodically sensing environmental information to the base station. Applications for sensor networks are typically performed by DTNs, which do not need real-time communication or continuous network connectivity [114, 141]. Alternatively, each UAV can move by following a predefined path scheme, where each waypoint is located near a static sensor; the moving UAVs gather the data from each sensor (via a wireless connection), and then return to the base station. This case suggests the adoption of a path-planned mobility model, like

PPRZM, which can be eventually customized to adapt it to the desired path scheme (e.g., a path that covers all the sensors on the area).

4.4.7 Relaying Network

Application requisites. UAVs used as relay nodes deploy some backbones reaching the end users/nodes, in order to provide connectivity among them. Typically, the mobility of such backbones is very low, since there is no particular reason why UAVs should move in space, as well as to ensuring connectivity. In fact, the more static are the drones, the better is the network performance. There are cases where the end nodes of the backbones must follow moving targets to guarantee communication with them (e.g., rescue personnel or victims in disaster areas), but the mobility is still limited to the end nodes, and is characterized by low speeds. If the targets to be covered are moving fast, like in the case of maintaining connectivity among vehicles (e.g., smart city scenarios), UAVs can behave as gateways that move along the streets.

Solutions. The request for a communication infrastructure where there is no connectivity can be performed by statically deploying a set of UAVs, which act as a relay chain among UAVs, or even in a heterogeneous scenario involving UAVs, vehicles, pedestrians, etc. [142, 143].

In urban scenarios, UAVs can be used as routers for V2V connectivity, allowing vehicles on the streets to communicate with each other; a simple MG model can represent such scenario.

Chapter 5

Proposed Solutions for Routing Problem

In this chapter, we illustrate our particular contribution in terms of design and implementation of new protocols for FANETs. In Sec. 5.1, we describe the design of our hybrid protocol *G-CR*. Then, in Sec. 5.2, we illustrate our routing protocol for DTNs based on geographic location information and planned location waypoints.

5.1 Greedy-Closer Request (G-CR)

As we have seen, AODV [82] is one of the most popular routing protocol proposed in this context. Its reactive approach to route discovery make it a suitable candidate in presence of mobility. In scenarios where there is a path toward a destination, in absence of wireless errors, AODV is guaranteed to find a path towards the destination, but at the cost of non negligible control traffic produced in the route discovery phase. This problem is further exacerbated in high-density scenarios where due to node movement control traffic might grow exponentially with the network size.

On the other hand, a well known routing protocol class is represented by position-based approaches which rely on a nodes' location alleviating the burden for a route discovery process. In this context, Greedy is the simplest example of a position-based protocol. At each step, Greedy finds the best next node to reach the destination, according to the distances from the destination itself. However, the approach suffers from the local *minima* effect, stalling the packet routing process at intermediary nodes.

We propose a hybrid position-topology based routing approach whereby the benefits of both worlds are combined. We propose *Greedy-Closer Request* (G-CR), which tries to strike a balance between a position-based forwarding and whenever the forwarding process stalls, it recovers by using a topology, reactive-based approach such as AODV. Our idea is to design a routing protocol able to change its behavior according to the local condition of the network. In particular the protocol switches between two modes: *greedy-mode* and *creq-mode* (closer request-mode).

The location information is received by *hello messages*, periodically sent by every node in the network (the packet header of a hello message is shown in Fig. 5.1a). The routing algorithm starts with greedy-mode, where the current node c forwards the data packet to the closest node to the destination among its neighbors. The greedy approach performs until it arrives at a local minimum l. At this point, the protocol switches to creq-mode, with the aim of looking for a node in the network that is closer to the destination than l. Algoritm 1 shows this switching procedure when a data packet is received. The creq-mode starts like AODV with node l broadcasting a *Closer Request Packet* (creq packet), with an initial Time-to-Live (TTL) of TTL_START hops (or using a max distance from l). The creq packet includes the address of l, the distance $dist(l, d)$ and d's coordinates (refer to Fig. 5.1b). Each node k receiving the creq packet (see Algorithm 2) calculates the distance $dist(k, d)$ and compares it with $dist(l, d)$ stored in the packet itself. If k is farther from the destination than node l, it just forwards the creq packet. Otherwise, if it is closer to d than node l, it sends a *Closer Reply Packet* (crep packet) towards l, including its address (refer to Fig. 5.1c. We call this node an *anchor node* (AN). When l receives the crep packet (see Algorithm 3), it sends the data packet to it. At this point, the AN starts the packet forwarding going back to greedy-mode.

When a node receives a creq packet, it stores in its routing table the next hop to reach the requester. In this way, it is possible to correctly forward an eventual incoming crep packet. Similarly, when it receives a crep packet, it stores the next hop related to the reply node, in order to correctly forward the data packet to it.

For each request transmission, a timer is started by the node l. If l does not receive a crep packet before the timer expires, it repeats the request, using a $TTL = TTL + TTL\_INC$. If the crep packet does not arrives again, node l repeats the request incrementing the TTL until arriving at a TTL_THR (threshold value). If $TTL \geq TTL\_THR$ the request for the anchor node is failed and the node l waits for several second before start a new request (or simply drops the data packet).

Similarly, the other nodes receiving a creq packet, store the related entry and start a timer. When the timer expires, the node simply deletes the record. If a node receives a creq packet that is already received, that packet is just discarded.

In the following, we present the pseudo-code of the algorithm. Especially, the behavior

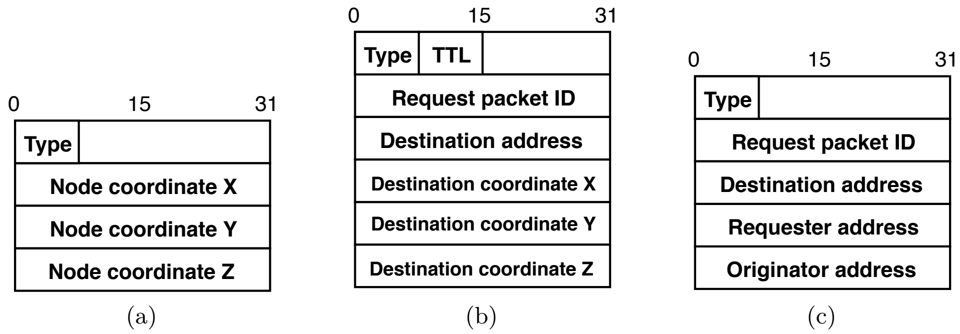


Figure 5.1: Representation of a hello packet header (a), creq packet header (b) and crep packet header (c).

of each node according to the type of packet it receives. Some used terms are defined here below.

- closest(d): find the closest node to the destination of the data packet among its neighbors. Returns null if there are not nodes closer than the current node.
- m.type: return the type of the message (data, creq, crep)
- send(n, m): transmit the message m to the neighbor n.
- buffer(data): store the data packet in the queue.
- store(creq.id): store the rreq id,
- isPresent(creq.id): check if the creq id is already present on the node's memory
- next(n) : next node in routing table to reach node n
- prev(n) : node from which the packet is arrived
- rt : routing table of current node
- myId : id of current node

Algorithm 1 Receiving a data packet

```
procedure RECV(data)
    next ← closest(d)
    if next = NULL then
        if next(d) exists in rt then
            send(next(d), data)
            EXIT
        end if
        buffer(data)
        creq.s ← myId                          ▷ store my id on rreq packet
        creq.d ← d                             ▷ store dest id on creq packet
        store(creq.id);
        send(broadcast, creq);
    else
        send(next, data);
    end if
end procedure
```

Algorithm 2 Receiving a creq packet

```
procedure RECV(creq)
    if !isPresent(creq.id) then
        next(creq.s) ← prev(creq)
        if closest(creq.d) != null then
            crep.r ← creq.s
            crep.a ← myId
            crep.d ← creq.d
            send(next(creq.s), crep)
        else
            store(creq.id)
            send(broadcast, creq);
        end if
    end if
end procedure
```

Algorithm 3 Receiving a crep packet

```
procedure RECV(creq)
    next(crep.s) ← prev(crep)
    next(crep.d) ← prev(crep)
    if crep.r = myId then
        for each data in buffer with destination d do
            send(next(d), data)
        end for
    else
        send(next(crep.r), crep);
    end if
end procedure
```

In Fig. 5.2 a flow chart of the G-CR forwarding protocol is shown. All the actions defined in the pseudocodes are graphically explained in this flow chart, according to the incoming packet type.

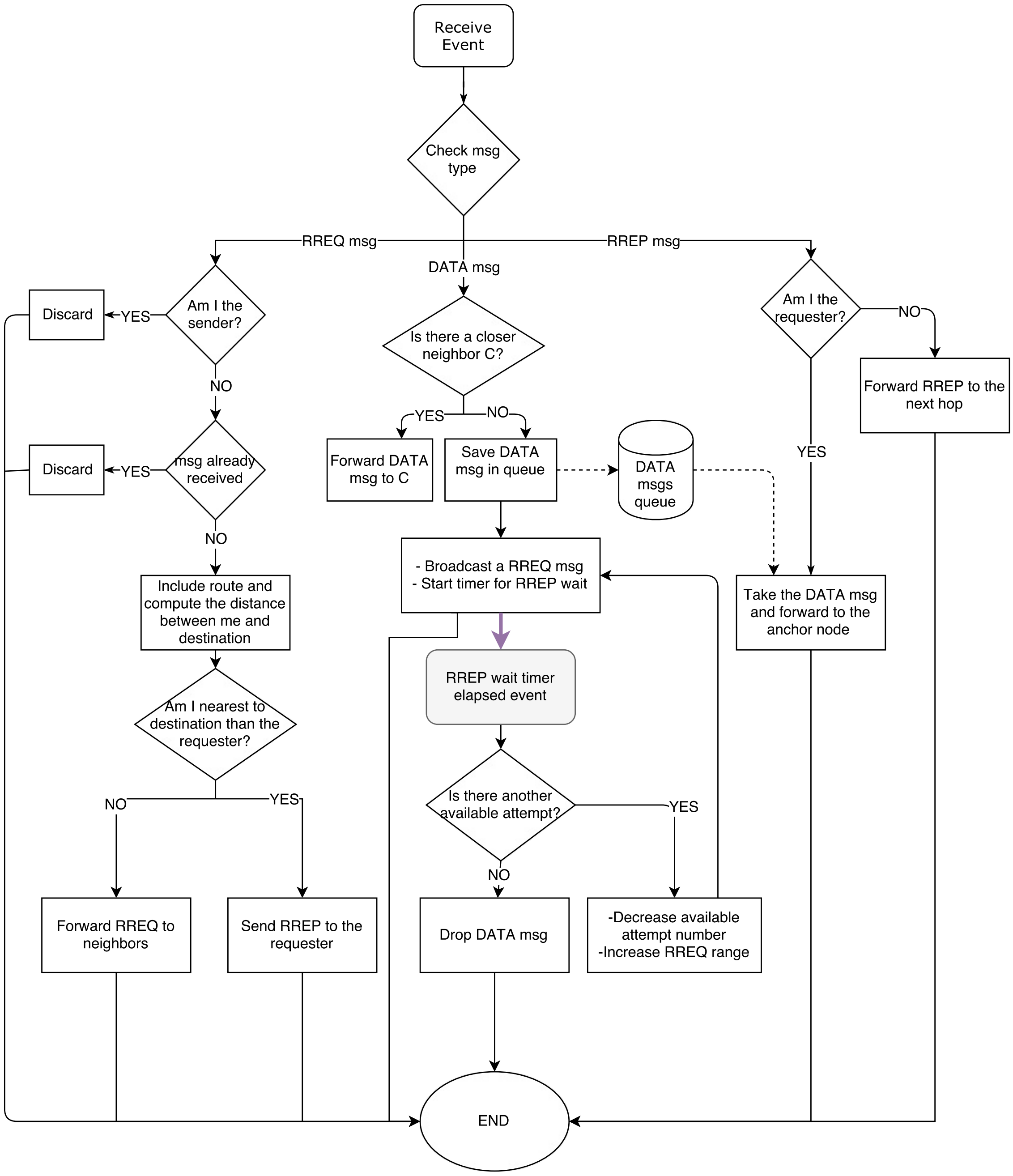


Figure 5.2: Flow Chart representing the routing algorithm process according to the incoming packet's type.

5.2 Location-aware Waypoint-based protocol for DTNs in Search and Rescue Operations

Classic DTN routing protocols use the store-and-forward policy, where each node holds the entire message or chunks of it (bundle) to forward it to any other node at a later stage. Other more advanced routing protocols rely on diverse information to correctly forward the messages among the nodes in the network. Some of these are called "context-aware" routing protocols, since they use information about the network to take forwarding decisions. This information may include the nodes' location and the next positions that the node will reach, when considering a planned path. The search and rescue scenario is a tipical context where rescue team may use a number of UAVs to independently check the ground area of interest in order to find possible targets, following certain predetermined route path. Hence, this and other applications require an already planned route schedule, where the UAVs' path is defined in advance. This aspect may be exploited to the advantage of the routing process.

Our proposed solution uses two information sources regarding the nodes involved: the current location and the mission planned path under the form of waypoints. The UAVs move along predetermined paths, and follow an *a priori* known schedule. This information is used by the protocol to predict the future locations of each relaying node and the time it will reach the locations (time of arrival or time of intercept), in order to transmit any message destined to a certain recipient crossing the path of the relay node. It is hence clear that a node waits for specific selected contacts and transmits the packet only to those. This aspect makes our proposal a partial replication-based strategy, like the Spray and Wait protocol. For this reason, we name our protocol *GeoSaW* (Geographic Spray and Wait).

In our solution, in addition to the node's position, the future path of the node should be communicated. Typically, this information is expressed under the form of a list of coordinates, which represent the waypoints. Hence, at every connection between two UAVs, they exchange their future path in order to process a possible bundle forwarding.

Differently from classic position-based routing protocols, GeoSaW (and more generally, almost all the DTN routing protocols) is just partially affected by the beaconing, because of their intrinsic characteristic. Classic position-based routing protocols need the correct neighborhood information almost every time, since they totally rely on the current position of the nodes to make the forwarding decision; in this case, inaccurate and outdated neighbor information could significantly affect the correct next-hop decision process. On the other hand, our DTN routing protocol mostly relies on the future path of nodes, which does not change over time; once the first beacon is arrived, subsequent beacons are not relevant for the routing process. Hence, we should pay attention to the beacon timing regarding the first beacon.

In this work, we do not analyze this aspect, but we intend to explore the beaconing in the future, when moving from The One simulator to the Network Simulator 3, which reproduces more realistically the communication and protocol aspects of the network.

5.2.1 Forwarding Process

The decision regarding a possible message forwarding is taken at a node c when:

- a new data packet is created from c or arrives at c from a neighbor node: c checks all its neighbor nodes to find any carrier that will reach the message's destination.
- a new connection with another node u takes place: c checks all the messages in its buffer to find any message for which u could reach the destination.

Once one of these cases occurs, GeoSaW works as follows. If a new message is created, an initial Time To Arrival (TTA) value is set to infinite. TTA is the minimum known time the message will take to reach its destination being carried by a node. If the TTA is less strict than infinite, the related carrier node is called *Final Node* (FN); in our work, we assume the ideal case in which an FN will deliver a message to the recipient with very high probability, avoiding delay errors, transmission failures, etc. When c analyzes an available node u, it processes its future path $[w_k, ..., w_n]$ (w_k is the waypoint that u is currently reaching) for the involved data packet, and checks if any segment $[w_i, w_{i+1}]$, $i = k, ..., n-1$, crosses the transmission range of the bundle's recipient; if so, it calculates a new TTA. If the new TTA is less than the current TTA value of the data packet, node u is a new (and better than c FN for that data packet, and c transmits that packet to it, after storing the new TTA into the packet field. Otherwise, c does nothing. Once forwarded, the data packet is normally deleted from c's buffer. In this way, there is only one packet in the entire network, like as in the First Encounter protocol. Another version of the protocol includes the possibility to avoid the deletion of the data packet from the buffer, in order to increase the chance to find a better FN, but at the cost of increasing the overhead. In the performance evaluation the two cases are compared. If c is a FN for the data packet and it encounters another FN node u, it calculates the TTA for that node TTA_u, and compares this value with the TTA stored in the data packet, TTA_c. If $TTA_u < TTA_c$, c stores TTA_u into the packet, transmits the packet to u, and deletes it from its buffer. Otherwise, it does nothing.

Final Node Condition Evaluation

In order to consider a neighbor node u as FN, the current node c evaluates the u's future path $[w_k, ..., w_n]$. For each segment $[w_i,\, w_{i+1}]$, $i = k, ..., n-1$, it checks two conditions:

- 1. the waypoint w_{i+1} is located within the transmission range of the recipient d.
- 2. the waypoint w_{i+1} is located within the blue area shown in Figure 5.3.

If one of these two conditions is satisfied, it is enough to make u an FN, as this means that this node will certainly connect to the recipient.

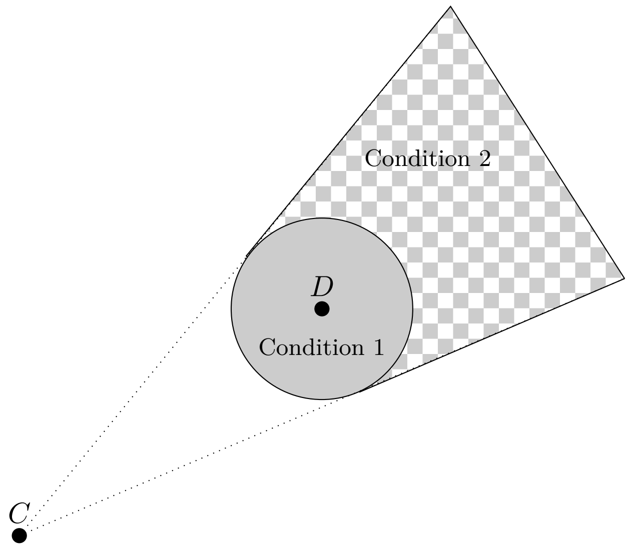


Figure 5.3: The two condition areas in which the next waypoint shall be to turn u into an FN.

In order to analyze the performance of our proposal, we conducted a comparison considering the other traditional DTN routing protocols (Epidemic, Spray andWait, First Contact, Max Prop) introduced in 3.3. The performance comparison results are shown in Sec. 6.6.

Chapter 6

Results

In this chapter, we show the obtained performance trend of the considered routing protocols. In particular, in Sec. 6.1, we introduce the adopted simulation environment for our benchmark. Next, in Sec. 6.2, we discuss the performance comparison results of considered position-based routing protocols. In Sec. 6.3, we analyze some significant protocols in a different environment, reproducing a realistic IoV scenario, in order to outline their behavior. In Sec. 6.5, we discuss the performance results of our proposal G-CR focusing on their overcoming respect to the other solutions, and finally, in Sec. 6.6, we analyze the comparison results of our proposed DTN routing protocol with the classic ones.

6.1 The Network Simulation Environment

The simulation is undertaken in the *Network Simulator 2* [144], widely known as *NS-2*, an object-oriented, discrete event-driven simulator tool useful in studying the dynamic communication networks. The simulator is equipped with a native implementation of topology-based routing protocols (AODV, DSDV, DSR, etc.), and we have implemented the discussed forwarding algorithms adding a new routing module (geo) into the *NS-2*'s code. Our code can be downloaded from [145], in order to let the reader reproduce the same simulations. It is noteworthy to mention, that the targeted simulation environment does not natively support 3D environments. To this end, we had to apply a publicly available patch [146]. For every section we provide some details regarding the adopted simulation parameters. For each simulation, the distance units are assumed to be expressed in meters (m).

Communication wise, for all the following simulations, we adopt the 802.11 g standard as, with respect to other technologies, it offers an excellent tradeoff between coverage and bandwidth; in our previous test [147, 148, 149], we were able to communicate without packet losses up to 1.5 km. Energy wise, the electromechanical part of the drone is the most consuming one, hence employing another communication technology does not have a great impact on the final outcome. Nevertheless, one can always make use of the low power modes offered by the IEEE standard.

6.1.1 Routing Performance Metrics

We introduce the metrics of interest to evaluate the performance of the routing protocols described in this work. Most of the works evaluate the state-of-the-art proposals considering these metrics.

Packet Delivery Ratio. Is the ratio of the number of data packets received by the destination(s) to the number of data packets sent by the source(s). The primary objective of a routing algorithm is to guarantee the delivery of all the packets.

Path Dilation. It is also called *stretch factor* and corresponds to the average ratio of the number of hops traversed by the packet to reach the destination, to the number of hops of the minimum path from source to destination. This performance metric will give us an idea of how well the protocol is performing in terms of packet delivery at different nodes density.

Packet Overhead. It is the number of routing control packets sent (in bytes) divided by the total number of delivered data packets (in bytes). Here, we analyze the average number of routing packets in bytes needed to deliver a single data packet. This is needed because the size of routing packets may vary.

End-to-End Delay. Also called *latency* or *delay*, is the average time taken by a packet to reach the destination. This metric is calculated taking the time at which each packet was transmitted by source and the time at which the same packet arrived to destination. This includes all possible delays caused by buffering during route discovery latency, queuing at the interface queue, retransmission delays at the MAC, propagation and transfer times. This metric is crucial in understanding the delay introduced by path discovery.

6.2 State-of-the-art Performance Comparison

All the algorithms presented in Sec. 3.2 are taken from different proposals and have been assessed on different simulators, under different assumptions, conditions and settings, hence their performance is not comparable considering each single work. To this end, we devise a common set of network configurations used to asses the different metrics under consideration.

6.2.1 Simulation Setting for the Performance Comparison

Inspired by the related works, our simulation scenario consists of N nodes placed in random positions in a 3D cubic space. The cube, in our case, has 1000 meters of side length. For each scenario instance, the nodes are placed randomly inside the 3D cube. Then, the topology is checked and the placed nodes are moved in order to reach a complete connected network, which means that each node can reach any other node in the network through multi-hop communication. The node transmission range is set to 200 meters. To simplify the simulation, we assume that the nodes are not moving, obtaining a static ad-hoc network. The motivation of this choice is twofold: (a) the need to consider initial essential parameters for the comparison (i.e. the minimum path) that would change if the nodes move, and (b) the assumption that the time employed to route a packet from source to destination is irrelevant compared to the physical node movement, and therefore the nodes of the networks are essentially still. A summary of the simulation parameters is shown in Table 6.6.

Table 6.1: Simulation parameters of the comparison environment.

Parameter	Value
NS-2 version	2.35
MAC type	IEEE 802.11g, FreeSpace model
Simulation area	1000 m x 1000 m x 1000 m
Transmission range	200 m
Traffic type	CBR
Data packet size	512 bytes
Queue type	Drop Tail
Number of nodes	50, 100, 150, 200
Node mobility	Static

In order to gather several analysis results, we have performed three different comparisons:

- The first comparison looks into the behavior of the forwarding algorithms under varying network density, ranging in {50, 100, 150, 200} nodes.
- The second comparison analyzes the effect of varying the threshold values LT-R

(Local Threshold Random) and LT-F (Local Threshold Face) of the randomization-based and the face-based forwarding algorithms, respectively.

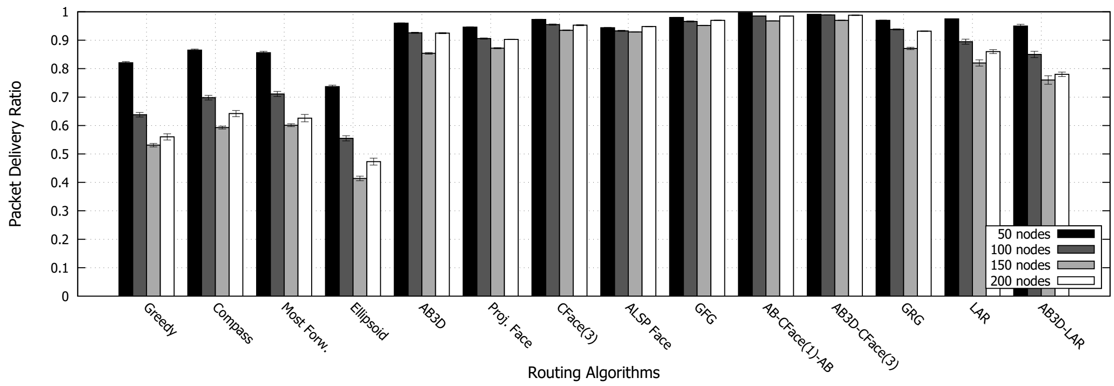
- The third comparison applies to all of the forwarding algorithms when varying the length of the shortest source-destination path, ranging in {1-3, 4-6, 7-9, 10+} hops.

For every comparison and parameter combination, we have performed 500 runs. Each run was combined with a different network topology randomly generated. In order to analyze the inherent performance of each routing algorithm, and inspired by previous work in the field (e.g., [60]), during each run a packet is sent from a random source to a random destination. In all the charts, the error bars are determined considering a 95% confidence interval.

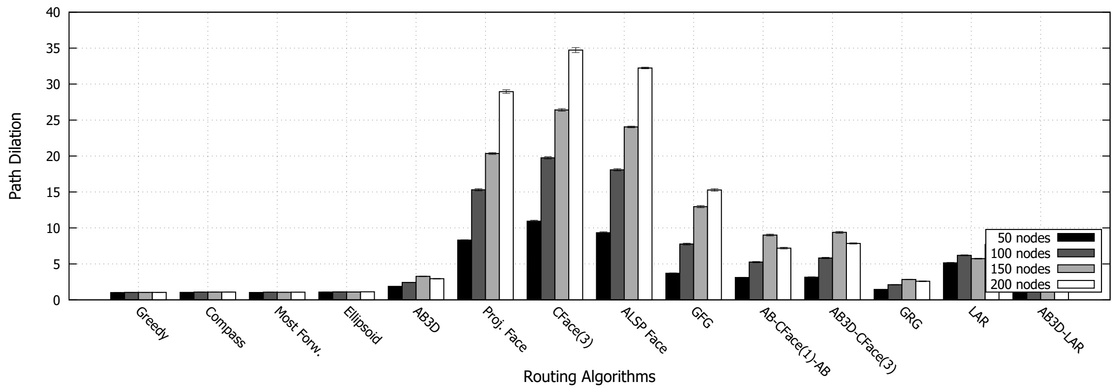
6.2.2 Comparison with dynamic number of nodes

We discuss here the outcome of the experimentation and analyze the protocols' performance under different settings. AB3D-CFace(1)-AB3D is abbreviated to *AB-CFace(1)-AB* in the charts.

The first comparison considers the routing protocols in a set of network instances consisting of 50, 100, 150, 200 nodes. The results are shown in Fig. 6.1. At first glance, from the results shown in Fig. 6.1a, there is a common behavior in the algorithms' delivery ratio, which decreases as the size of graph increases until 150 nodes, and regrows in the transition from 150 to 200 nodes. The deterministic progress-based forwarding algorithms have the lowest delivery rate, less than 60% in the case of 150 nodes. This means that in this scenario (150 nodes) there is more chance of getting stuck in a local minimum. However, they show the lowest path dilation (close to 1), as seen in Fig. 6.1b. AB3D and GRG obtain better results in delivery rate, with a peak in the case of 50 nodes (90%), increasing their path dilation respect to the deterministic solution. GRG reduces the number of hops thanks to the hybridization with Greedy. Projective Face, CFace(3), and ALSP Face) generally achieve better results than the previous ones, but the worst results in term of path dilation. This difference depends on the fact that these algorithms have more possibility to find the destination node, since LT-F resetting. CFace(3) can be defined as Projective Face, only choosing two plans that are *xy* plane and *xz* plane, and with the addition of the *xz* plane. Therefore, since the algorithm has three possible attemps to finding the destination instead of two as in Projective Face, the delivery rate is a little higher. GFG achieves better results compared to ALSP Face, regarding delivery rate and is able to reduce the number of traversed hops. This is probably caused by the greedy method phase that, at first, carries the packet towards the destination and then,



(a)



(b)

Figure 6.1: Delivery rate (a) and path dilation (b) of all algorithms, for different network sizes, and with LT-R=N, LT-F=2N and GT=6N.

if a local minimum is found, the face method phase starts, but with less search space. LAR is affected by a relevant traffic and does not show a very high delivery rate. Not even AB3D-LAR presents a good delivery rate, yet it has almost half of the path dilation when compared to LAR. From what we can see, the transition from 150 nodes to 200 nodes in the network increases the delivery rate, due to the increment of the node density. Focusing on path dilation, LAR and AB3D-LAR present very good values, at the cost of high traffic, due to the nature of the flooding technique.

6.2.3 Comparison with dynamic threshold values

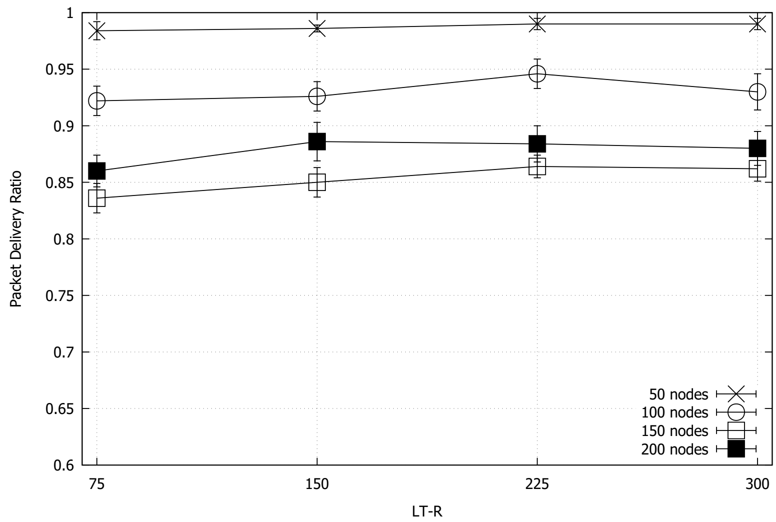
This assessment has the goal to evaluate the effect of the threshold values for the random-based and face-based algorithms, respectively. Our aim is to study the relationship between the LT-R and LT-F values to that of the delivery rate and path dilation. In this experimentation the LT-R and LT-F are dynamic, while the number of nodes N is not. Therefore, to get a fair treatment of the various experimental instances and to allow each algorithm to reach all their thresholds values (i.e., CFace(3) fails after three times LT-F), the GT value is computed as 2 LT-R for the randomized case and as 3 LT-F for the face case. For randomization-based analysis, AB3D is assessed with four different network sizes, whereas the face-based one are assessed with a network size of 150 nodes.

In Fig. 6.2 we can see the effect of varying the LT-R threshold value on the average delivery rate and the average of path dilation of AB3D in different graph sizes. Remember that the parameter combination chosen for the algorithm in this test is $M = 3$, $R = Greedy$, $S = Angle$. As shown in Fig. 6.2a, the delivery rate is generally stable for each LT-R and for each graph size. This means that, in this scenario, the increase of LT-R is useless and this aspect does not change with a different number of nodes. Since a little increase in the delivery rate means more delivered packets added to the average path dilation calculation, there is a little growth of the path dilation value, due to the increase of the number of traversable hops allowed by LT-R, as seen in Fig. 6.2b.

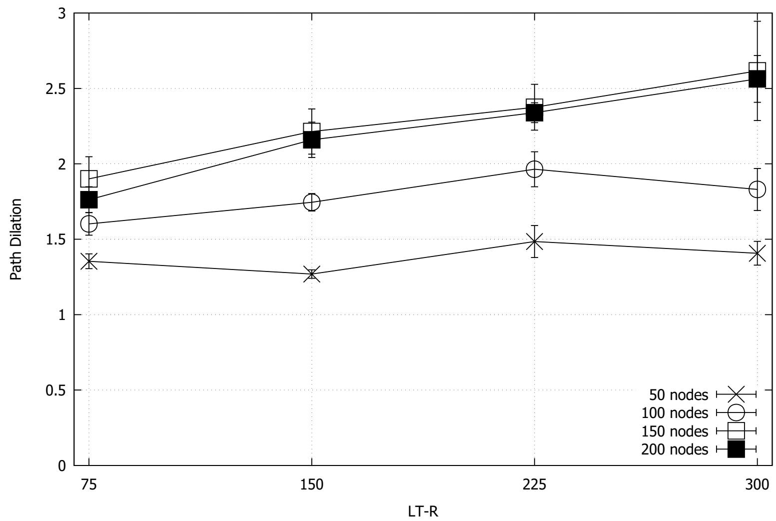
Figure 6.3 shows the effect of varying LT-F on the average delivery and average path dilation of the four algorithms that use face strategy, Projective Face, CFace(3), ALSP Face and GFG. Remember that GFG uses ALSP Face as a recovery phase. The increase of the delivery rate, in Fig. 6.3a, is very significant for all the algorithms until a threshold value of 300 (2N). Figure 6.3b shows the corresponding path dilation which continues to increase even though the delivery rates stops increasing noticeably. This indicates that a lot of looping occurs during the routing process, thus significantly slowing down the progress of the packet towards its destination. ALSP Face, CFace(3) and GFG perform well in terms of delivery rate, than progress-based and randomization-based strategies, but CFace(3) has a path dilation that grows more compared to the other. GFG performs well in delivery rate, and has the least path dilation with a steady value below 10.

6.2.4 Comparison with dynamic min path length

This section shows the results related to the application of all the routing algorithms considered in classes of graphs in which the length of the shortest path between source and destination is respectively 1-3, 4-6, 7-9, and >10 hops. These results are shown in Fig. 6.4. Note that for the first class (1-3 hops), all the algorithms have a high delivery rate



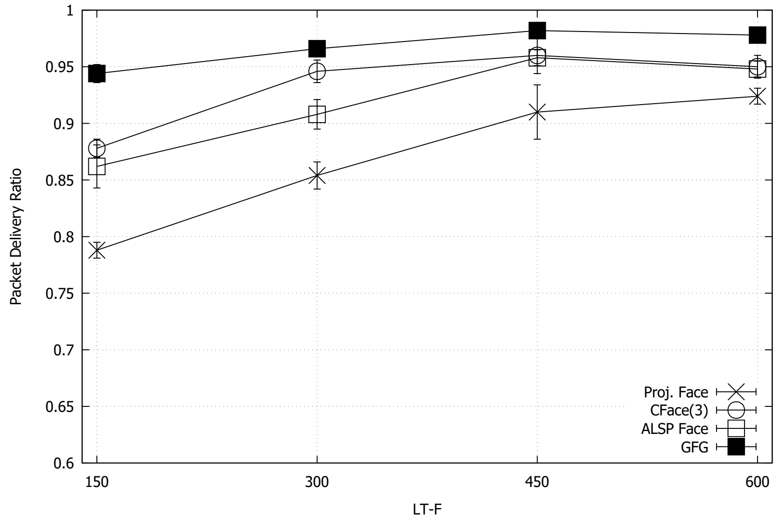
(a)



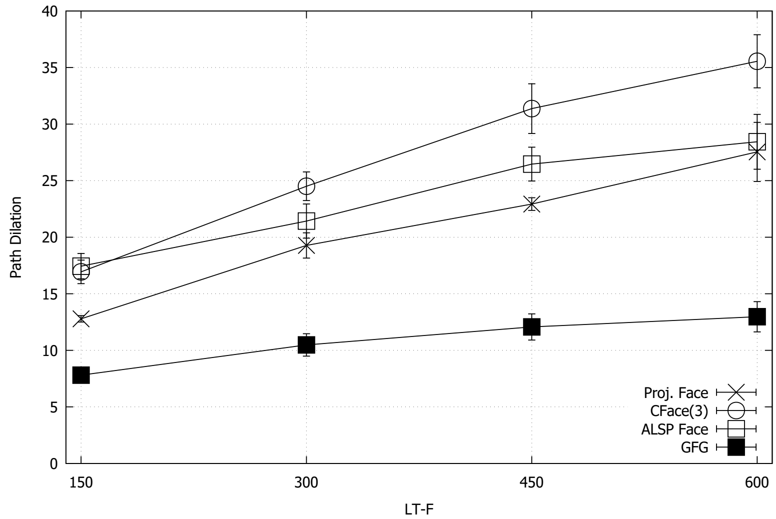
(b)

Figure 6.2: Packet delivery ratio (a) and path dilation (b) of AB3D in a graph of 50, 100, 150 and 200 nodes, with LT-R = 75, 150, 225, 300 and GT = 2 LT-R.

(see Fig. 6.4a) and a small path dilation (see Fig. 6.4b). This happens because the closer the source and destination are, the less chance to find local minima there is. Augmenting the shortest path, deterministic progress-based strategies are the first to worsen their performance; the destination is more distant and there is a greater probability to get stuck in a local minimum. Progress-based algorithms reach about 10% of delivery rate in the case of 10+ hops of the minimum path. AB3D and GRG perform well up to 4-6



(a)

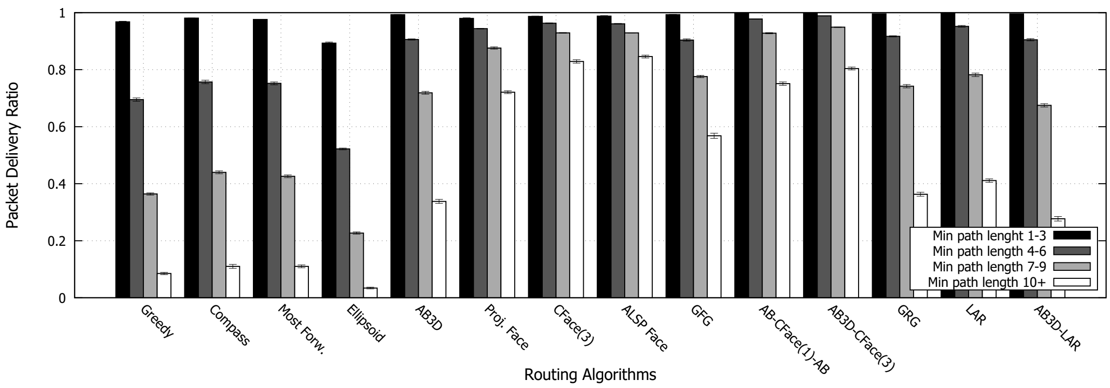


(b)

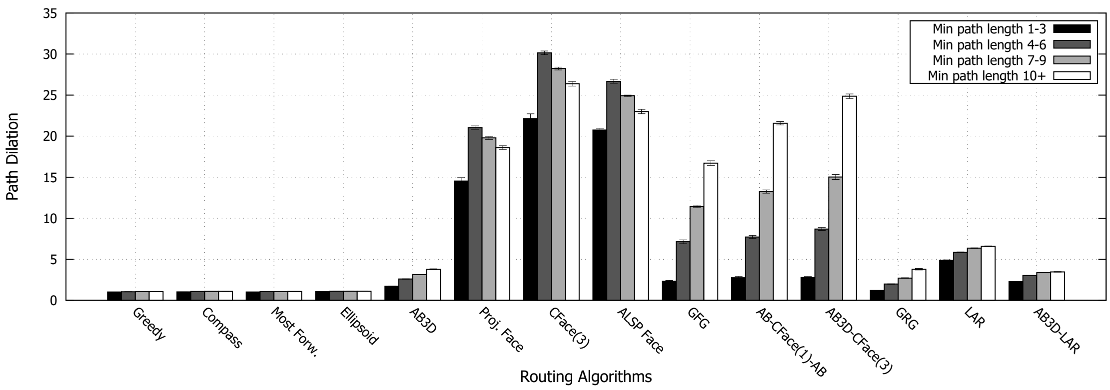
Figure 6.3: Packet delivery ratio (a) and path dilation (b) of Projective Face, CFace(3), ALSP Face and GFG, in a grapf of 150 nodes, with LT-F = 150, 300, 450, 600, GT = 3 LT-F and ABS = 100.

minimum path length, but from this point the delivery rate decreases. This is due to the increase of the number of links from source to destination which reduces the probability of choosing the righ path. However, the path length of the delivered packets remains short (Fig. 6.4a, AB3D and GRG histogram).

The performance of face-based and hybridized algorithms are also good for long paths,



(a)



(b)

Figure 6.4: Packet delivery ratio (a) and path dilation (b) of all algorithms, in a graph of 150 nodes, for different minimum source-destination path lengths, and with LT-R = N, LT-F = 2N and GT = 6N.

due to the fact that they succeed in reaching the destination within LT-F or GT, despite the increasing of the distance from source to destination. However, the traversed path is too long, due to an increment in the number of crossing links during the travel.

6.2.5 Results Discussion

A comparison of the studied position-based algorithms is provided in Table 6.2. It summarizes the main characteristics of each proposal, reporting the results in terms of performance and their salient features. The features highlighted are as follows:

- **Packet delivery ratio**. The ratio between the number of packets delivered to the destination(s) and the total number of packets sent by the source(s).
- **Path dilation**. The average ratio between the number of hops performed by the packets during the algorithm's execution and the number of hops of the minimum path.
- **Scalability**. The ability of a routing protocol to support network expansion in terms of number of nodes and network size, while preserving the performance trend. A routing protocol is scalable when performs well also in large size networks. In this paper, scalability is evaluated based on the path dilation.

From Table 6.2 we can draw further considerations. Progress-based forwarding algorithms are highly scalable, but have a low delivery rate. These algorithms are suitable for dense and uniform networks, which do not have local minima. Furthermore, these can be used in combination with other algorithms to reduce the path to reach the destination. Randomized forwarding strategies have one of the best performances in terms of scalability and delivery rates. They can outperform the progress-based strategy in sparse networks.

Face-based forwarding algorithms have significant path dilation; thus, they are not appropriate for dense networks, because there are many crossed links. Furthermore, in a continuous data flow scenarios we have a significant number of subsequent packets that travel around the network, and the long path followed by each packet can result in a large number of collisions and tailbacks. However, they perform well in sparse networks, where there are few nodes, hence few crossed links. Partial flooding algorithms can be used in small/medium networks, where multi path strategy does not greatly reduces the performances. Hybrid-based algorithms can perform well in a large range of network types, since they combine some advantages from base algorithms: delivery rate is high, path dilation is not high and scalability is high; this depends on the progress/randomized strategy combined with face-based methods.

Table 6.2: Performance summary.

Routing Protocol	Delivery Ratio	Path Dilation	Scalability
Greedy	Low	Very Low	High
Compass	Low	Very Low	High
Most Forward	Low	Very Low	High
Ellipsoid	Low	Very Low	High
AB3D	Medium	Low	High
GRG	Medium	Medium	High
Projective Face	Medium	High	Medium
CFace(3)	High	High	Medium
ALSP face	High	High	Medium
GFG	High	Medium	High
AB3D-CFace(1)-AB3D	High	Medium	High
AB3D-CFace(3)	High	Medium	High
LAR	Medium	High	Medium
AB3D-LAR	Medium	Medium	Medium

6.3 Topology vs Position-based Comparison in IoV

The main goal of this section is to show how topology-based protocols behave differently from position-based protocols in a 3D network characterized by mobile nodes. To this end, we have considered the main representatives routing protocols deemed as the state-of-the-art for classic MANETs and tested them over 3D topologies to evaluate current assets and technical challenges. While the simulator is equipped with a native implementation of the topology-based routing protocols (AODV, DSDV and DSR), we use some of the implemented position-based ones in the previous section. In particular, we take the two hybrid protocols (GRG and GFG) and DFS to make the comparison.

Given the absence of a reference mobility model for FANETs, we simulate this network by employing a synthetic mobility model while adopting realistic, well-established mobility models for other components of the network. While there have been efforts in devising mission or application-specific mobility models for UAVs, our study is not tailored to a specific use-case and goes beyond the specific application scenario.

6.3.1 Simulation Setting for IoV Performance Evaluation

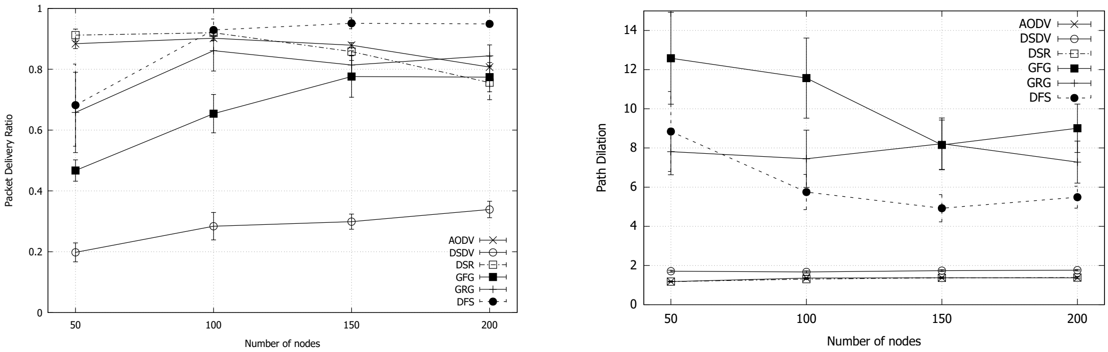
Our simulator environment consists of a set of nodes randomly generated in a cube of side length 1000 units with a transmission range of 250 units. We considered four different cardinalities for the set of nodes in the network: 50, 100, 150 and 200 nodes, in order to evaluate the protocol performance variation. Mobility models proposed in the past have been focused, for instance, on VANETs but not on drone networks [16]; we have hence considered the recommended mobility model but we had to adapt to our considered 3D environment. In essence, nodes alternate movement and stationary periods. For each cardinality of the set of nodes, four mobility constraints are chosen, in terms of pause time: 5, 20, 40 and 100 s, during which the node remains stationary, before resuming to move toward a new destination in the 3D space. The simulation duration is set to 100 s, so that the case with 100 s of pause time corresponds to a network with static nodes. The traffic scenario consists on 10 flows (10 different sources and 10 different destinations) of CBR traffic. These and the other mobility parameters are summarized in Table 6.6. For position-based routing protocols, the proactive beaconing process period (i.e, the time between two consecutive beacon transmissions) is set equal to 0.5 s.

Table 6.3: Simulation parameters of the IoV scenario.

Parameter	Value
MAC type	IEEE 802.11g
Simulation area	1000 m x 1000 m x 1000 m
Transmission range	250 m
Node max speed	10 m/s
Traffic type	CBR
Number of data flows	10
Data packet size	64 bytes
Packet rate	2 pckt/s
Queue type	Drop Tail
Number of nodes	50, 100, 150, 200
Pause times (sec)	5, 20, 40, 100 (static)

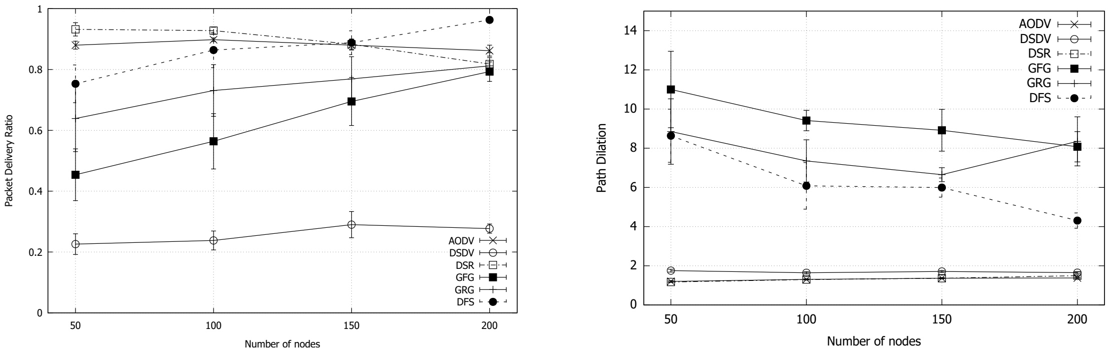
6.3.2 Simulation results of IoV Performance Comparison

In this section we show the performance results of routing protocols in a set of networks of 50, 100, 150, 200 nodes, with pause times 5, 20, 40 and 100 s. Basically, with a 5 s of pause time the nodes moves frequently, whereas with 20 and 40 s of pause time the network has less mobility and with 100 s of pause time all the nodes are still during the whole 100 s of



(a) Packet delivery ratio (b) Path dilation

Figure 6.5: Performance comparison among the protocols in a FANET varying the number of nodes. The nodes are mobile with pause time 5s.

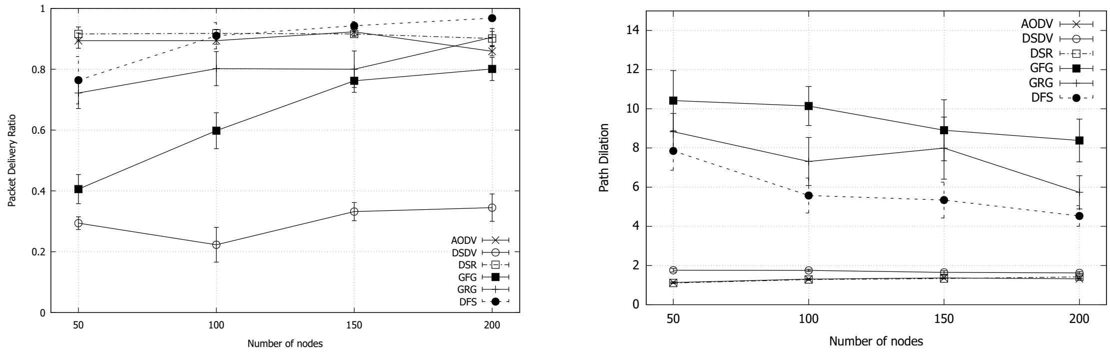


(a) Packet delivery ratio (b) Path dilation

Figure 6.6: Performance comparison among the protocols in a FANET varying the number of nodes. The nodes are mobile with pause time 20s.

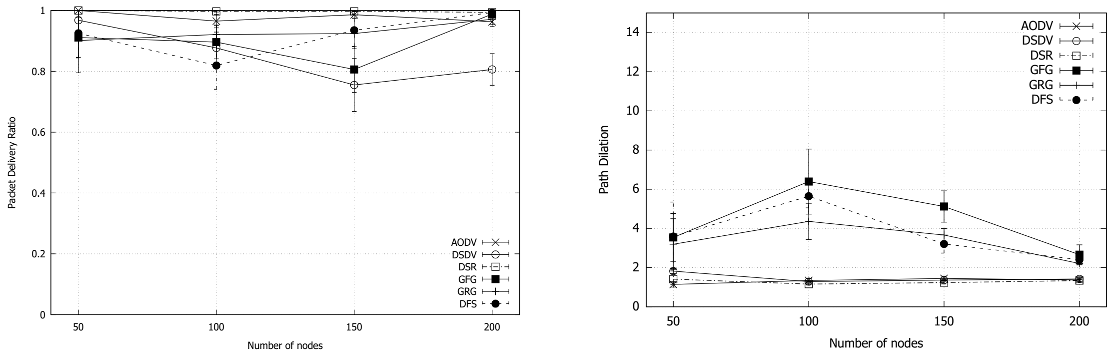
simulation.

Pause time of 5 s. Topology-based and position-based protocols perform in a heterogeneous way. For instance, in Figure 6.5 we can notice the different performances for each of the considered metrics. The delivery rate in low density scenarios(50 nodes) is quite low for position-based protocols, especially when employing the GFG scheme, while AODV and DSR perform at an acceptable level. When increasing the number of nodes, the packet delivery rate of the position-based protocols increases as well. The best performance when considering a network density of 200 nodes, is reached by the DFS scheme (about 95%). In terms of path dilation, position-based protocols achieve the worst performance. This



(a) Packet delivery ratio (b) Path dilation

Figure 6.7: Performance comparison among the protocols in a FANET varying the number of nodes. The nodes are mobile with pause time 40s.



(a) Packet delivery ratio (b) Path dilation

Figure 6.8: Performance comparison among the protocols in a FANET varying the number of nodes. The nodes are static.

is intuitively expected as these schemes rely solely on local knowledge. Instead, all the topology-based protocols perform well, with packets traversing a path of length close to the optimal length. We can also notice that when increasing the number of nodes, the path dilation decreases. This effect is explained due to the fact that having a denser network increases the chances of finding a straight path toward the destination. On the contrary, if a node has a low number of neighbor nodes, it may unfortunately happen that a few out-of-date information from neighbors could lead to a loop among the nodes, and hence to higher values of path dilation, until the neighbor table settles.

Pause time of 20 s. The results evidenced in Figure 6.6 show that there are not many differences when compared to the prior configuration. We see a little growth in the delivery rate for AODV and DSR and the position-based protocols. Also, position-based approaches present a reduction of values, whereas packets in the topology-based schemes on average traverse almost the same number of hops as in the case of 5 s pause time. In general, we can see that AODV and DSR achieve very good performance indexes both in terms of delivery rate and path dilation. In particular, DFS is able to achieve the best data delivery rate when the number of nodes is higher or equal to 100. Unfortunately, this comes at the cost of a path dilation that, although lower than other position-based schemes, results even three or four times wider than with AODV and DSR.

Pause time of 40 s. With a pause time of 40 s, we are considering a network composed by nodes with seldom mobility. DSDV performs well in terms of delivery rate when compared to prior configurations. Even the position-based approaches show better delivery rates than the case of 20 s pause time; furthermore, increasing the node density we achieve higher delivery rates due to the increment of alternative routes that a node can choose. The path dilation values are reduced for position-based protocols, which take a better decision for the next node, since the nodes are stationary for longer periods. In the case of topology-based protocols the path dilation remains between 1 and 2. In general, even in this case, AODV and DSR seem to be the best protocols to constantly ensure high delivery rate and low path dilation.

Pause time of 100 s (static network). In a static network, the delivery rate is as expected very high for all the routing protocols. If a protocol is not able to ensure the delivery of all packets it is due to the unreliable nature of the wireless link. Furthermore, the GFG algorithm still experiences some problems in delivering the packets, since the planarization of a 3D graph is not optimal and the algorithm gets stuck in a loop. The lowest performance of position-based protocols are shown in the case of 100 nodes, while with 200 nodes they are all able to reach more than 95% of delivery rate.

6.4 Performance of routing protocols in a realistic urban environment

In this section we analyze the performance of routing protocols considered in the previous section in an IoV environment. A multi-tier IoV network is comprised of moving cars, drones and traffic intersections equipped with sensing and communication capabilities. In

Figure 6.9: Example of IoV scenario.

this envisaged scenario, communication takes place amongst entities in a multi-hop fashion. Cars could exploit the IoVs sensing capabilities, gathering and merging information from different complementary sources in order to extend the drivers' perception beyond direct line of sight. The goal is to asses the feasibility of current routing proposals for a general IoV.

6.4.1 Simulation Setting of the Realistic IoV Scenario

Figure 6.9 depicts a potential deployment consisting of a heterogeneous, general purpose IoV network. The area represents a portion of a city with static sensors positioned in lamps and/or at traffic intersections, moving cars and drones flying above them (all these nodes are also endowed with communication capabilities). In the specific, the scenario consists of a total of 64 nodes arranged in different ways moving according to a specific mobility model. In particular, there are 40 mobile vehicles, arranged on a 4 x 4 grid of length 500 m, whose streets have a width of 10 m. Each vehicle is initially positioned at a random crossroad. Moreover, each vehicle randomly selects an axis (either x or y) and a crossroads on that axis, proceeding towards that point. The speed is randomly chosen from 14 m/s to 20 m/s (50 km/h - 72 Km/h). Next, there are 20 nodes representing flying drones, randomly deployed above the vehicles' grid at a random altitude ranging between 50 and 200 m. These flying drones follow a Random Waypoint mobility model, with a fixed speed of 10 m/s and a pause time of 5 s, during which the drone is assumed to perform some task (e.g., taking a picture, sensing enviromental conditions, collecting/distributing some data, etc.). As stated earlier, we employ a synthetic mobility model given the absence of a reference mobility model for FANETs. Along with the mobile nodes, at each crossroad there is one static node, representing e.g., access points (on buildings, stations or simple poles). A summary of the simulation parameters is reported in Table 6.4.

Table 6.4: Simulation parameters for the envisaged IoV scenario.

Parameter	Value
Simulation area	500 m x 500 m x 200 m
Transmission range	125 m
Vehicle movement characteristics	
Positioning	Random
Mobility model	Manhattan grid (3 x 3) blocks
Number of nodes	40
Speed of nodes	[17, 20] m/s
Pause time	0 s
UAV movement characteristics	
Positioning	Random, altitude from 50 to 200 m
Mobility model	Random Waypoint
Number of nodes	20
Speed of nodes	10 m/s
Pause time	5 s
Static nodes	
Positioning	One at each intersection
Number of nodes	4

6.4.2 Simulation Results of Realistic IoV Scenario Comparison

As we can see in Figure 6.10, DSDV has the lowest performance in terms of delivery ratio, with an average of 10% of the packets reaching the destination. This is due to node mobility causing path disruptions with the protocol not being able to counteract the effects. On the other side, the rest of the protocols achieve acceptable performance indexes, but none is capable to guarantee absolutely reliable packet delivery. When analyzing the path dilation in Figure 6.11, AODV and DSR achieve a good performance, along with DSDV. Instead, in position-based approaches data packets traverse long paths before finally reaching the destination. This could easily lead to the undesirable effect of packets queuing up in the nodes' buffers, and flows ending up interfering with each other. Considering all, there is no clear winner amongst the studied protocols and no one seems to provide outstanding performances. We believe this represents a crucial technical challenge that needs the

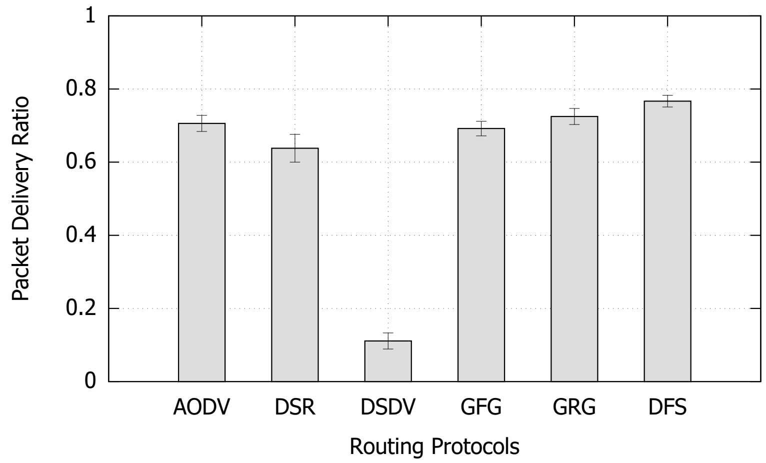


Figure 6.10: Performance comparison (packet delivery ratio) among the considered protocols in the scenario depicted in Figure 6.9.

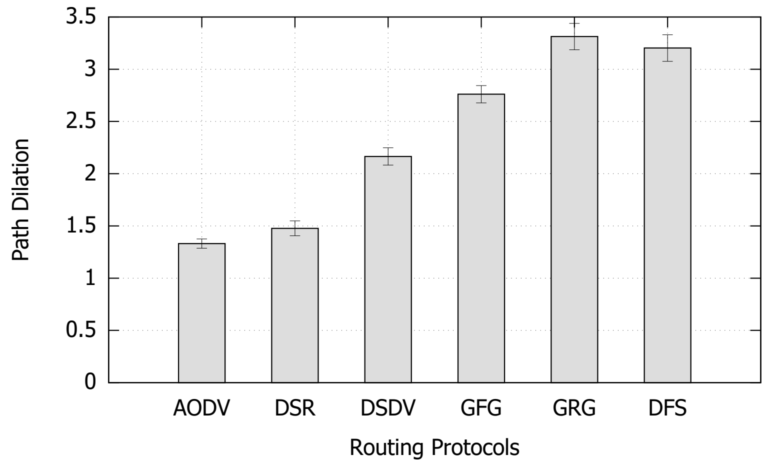


Figure 6.11: Performance comparison (path dilation) among the considered protocols in the scenario depicted in Figure 6.9.

researchers' attention in order to enable IoV.

6.5 G-CR Performance Evaluation

In this section, we discuss the experimental settings and strategy used to evaluate our proposal G-CR. Next, we present a comparison analysis in terms of delivery ratio, end-to-end delay and control overhead.

6.5.1 Simulation Setting of G-CR Performance Evaluation

To analyze the performance of our proposal, we devised a 3D network scenario of size 1000 x 1000 x 500 m. All the nodes move according to the 3D Random Waypoint mobility model, with a fixed speed of 10 m/s and a pause time of 60 s. We evaluate the protocols under different density scenarios, varying from 100, 120, 140, 160, 180 and 200 nodes, placed randomly in the scenario. Each simulation configuration is run 50 times and consists of a source and a destination node chosen randomly from the population. Source and destination nodes are involved in communication consisting of a CBR flow. We choosed a starting TTL = 1, to permit any 1-hop node to reply if it is an AN; then, we set a TTL_INC = 2 and a TTL_THR = 7, in order to explore a large part of the network in the worst case. Additional simulation details are shown in Table 6.6.

The scenario is implemented in NS-2 and for the comparison we considered three other routing solutions: AODV, Greedy and Face.

To compare the amount of control packets produced by our proposal, we introduce the analysis of another metric, called *Normalized Control Overhead*; it represents the number of routing control packets sent (in bytes) divided by the total number of delivered data packets (in bytes). Here, we analyze the average number of routing packets in bytes needed to deliver a single data packet. This is needed because the size of routing packets may vary. This metric is evaluated differently for G-CR, AODV, Greedy and Face routing, and is defined as in the following (the control overhead of each protocol is called co<ProtocolName>):

$$coG{-}CR = (creq + crep + hello)/data$$

$$coAODV = (rreq + rrep + rerr)/data$$

$$coGreedy = hello/data$$

$$coFace = hello/data$$

6.5.2 Preliminary Results of G-CR Performance Evaluation

In Fig. 6.12 the average delivery ratio and the relative error of the considered routing protocols under varying network densities are shown. We can see that G-CR and AODV have a very high delivery rate when compared to the pure position-based protocols. As expected, Greedy and Face have significant low values. The reason is that, since their total dependence on local neighbor information (neighbor table) in order to make the forwarding decision, most of the time these tables (the neighbors' locations) are out of date, especially

Table 6.5: Simulation parameters in the mobile scenario.

Parameter	Value
Simulation area	1000m x 1000m x 500m
Simulation time	300 s
Number of data flows	1
Data packet size	512 bytes
Data packet rate	2 KB/s (4 pkts/s)
Number of nodes	100, 120, 140, 160, 180, 200
Transmission range	200 m
Node speed	10 m/s
Pause time	60 s
Configuration of protocols	
Starting TTL	1
TTL_INC	2
TTL_THR	7
per hop time	10 ms
waiting rrep	TTL * 2 * per hop time
Position-based parameters	
Hello packet timer	15 s
Face routing LT-F	100

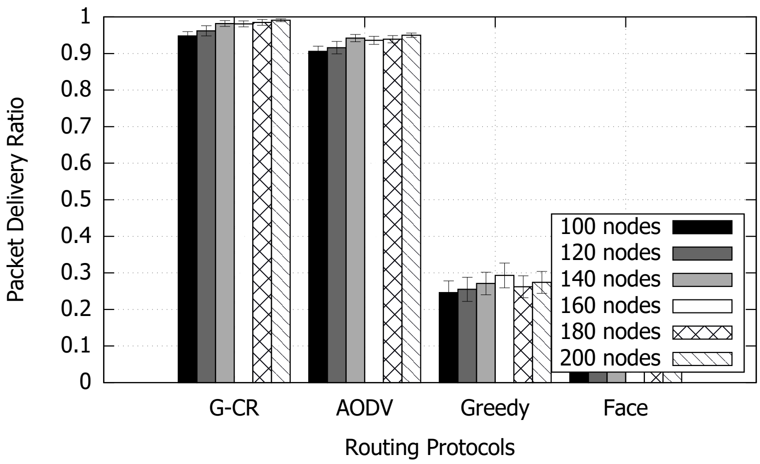


Figure 6.12: Packet delivery ratio comparison with varying network densities.

when the hello interval is considerable. This leads to packet drop/loss, wrong paths, loops, and fictious local minima. However, we chose 15 s as hello interval in order to reduce the routing overhead of position-based protocols. Furthermore, Greedy has difficulties in

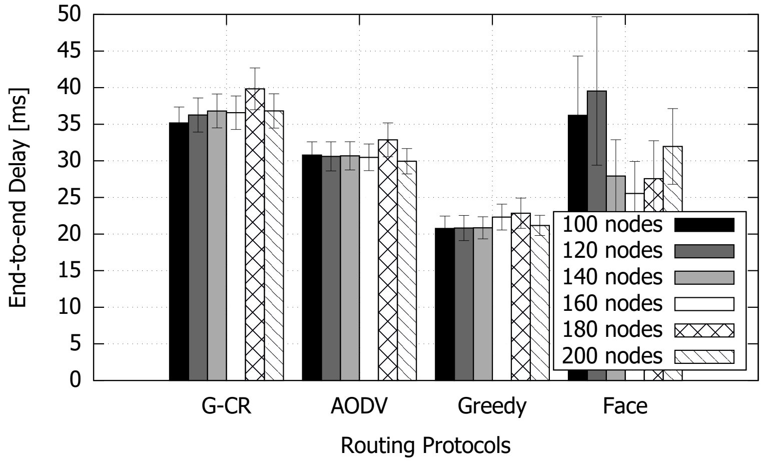


Figure 6.13: End-to-end comparison with varying network densities.

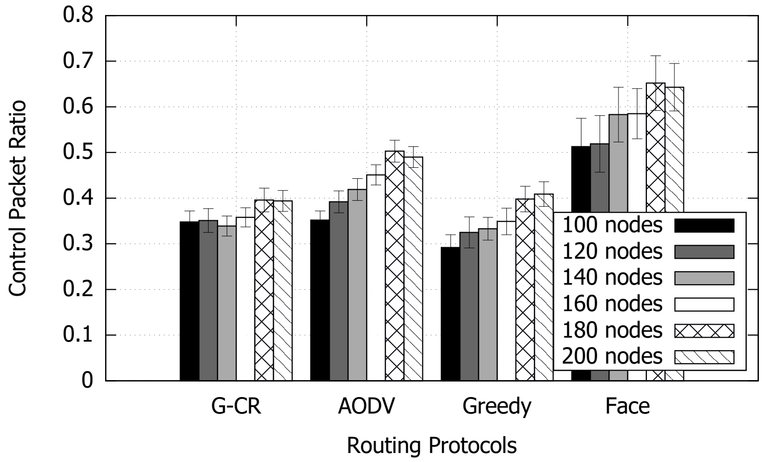


Figure 6.14: Control traffic ratio comparison under different network densities.

reaching a high delivery rate, due to the peculiar local minima problem, and Face gets stuck into loops for the lack of planarization. G-CR's delivery ratio is slightly higher than AODV's one and it steadily increases with an increasing network density.

Figure 6.13 shows the average end-to-end delay of the data packets. G-CR exhibits slightly higher delay when compared to AODV, of about 5 ms. The reason is that, while AODV receives the rrep the first time it sends a rreq (if a route is available), G-CR should try several creq before finding an anchor node. The wait time for a crep and the subsequent creq could increase the delivery time for some packets. This multiple request system is needed by the algorithm and without it G-CR would behave as Greedy in terms of packet delivery. But it is only the case of a local minimum solution that increases (by a lot) the average, with the advantage of delivering those packets.

A noticeable difference between G-CR and AODV is the control overhead as shown in Fig. 6.14. The amount of control packets sent for a delivered data packet is reduced using G-CR. This fact suggests an advantage in using limited request flooding rather that the approach taken by AODV. Pure position-based approaches incur less overhead; however, this comes at a cost of a much lower delivery rate when compared to our proposal.

6.6 Location-aware DTN Proposal Performance Evaluation

In this section, we present the performance results of our DTN proposal obtained via simulation. In this case we did use another simulator, called *The One*.

6.6.1 The One Simulator

The One (The *Opportunistic Network Environment*) simulator [150] is designed for the DTN routing protocols. Written in Java, it is capable of simulate routing of packets (defined there messages) between nodes with various DTN routing algorithms and sender and receiver types. The One can generate node movements using differnet mobility models, and provides a graphic visualization of both mobility and packet forwarding in real time in its interface.

6.6.2 Simulation Setting of the DTN Scenario

For the evaluation, we use a typical search and rescue scenario, where some (searching) UAVs perform a scanning search in a specific area. This scenario is similar to the one introduced in [112] for DTN protocols evaluation. When any UAV has some relevant information about the search, it has to transmit this information to the base station, which is located outside the search area. The only way to deliver the information is through the delivery UAVs, that could have different tasks, not necessarily related to the search operation. They work in a larger operation area, which contains the search area. In Figure 6.15 a graphic representation of the search and rescue scenario is shown. This network is typically of low density. In Table 6.6 the parameters are given as the values used in the simulations. A RWP mobility model for the delivery UAVs is chosen as a preliminary model. It is clear that all the waypoints of the RWP model are already planned at the start of the simulation, in order to simulate a planned path for each UAV. For each configuration, 100 runs are performed in order to have an average value of the three metrics: packet delivery ratio, overhead ratio, and latency. These three metrics are studied by varying the number of delivery UAVs, to evaluate the evolution of the performance when changing the network density.

We conducted two comparisons. The first one considers our protocol compared against the other traditional DTN routing protocols (Epidemic, Spray and Wait, First Contact, Max Prop). In particular, for Spray and Wait, the chosen number of allowed copies is set to 5. The second comparison considers different parameters combinations of the proposed protocol.

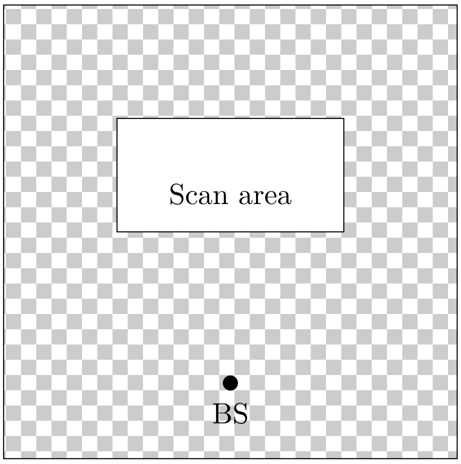


Figure 6.15: The simulated searche and rescue scenario. In the white rectangle area the searching UAVs perform a scanning task, moving only within that area. The other nodes (delivery UAVs) perform some other actions in the larger checkerboard area. The base station (BS) is located outside the scanning area, but inside the larger area. In this way only the delivery UAVs can deliver the packets originated by the scanning UAVs.

6.6.3 Comparison with classic DTN protocols

In this simulation, we compare our proposed protocols with the classic DTN protocols implemented in The One. In Figure 6.16, we can see that the average packet delivery ratio of GeoSaW is almost 1, like Epidemic and MaxProp, for all node densities. On the contrary, the values for the First Contact protocol grow until arriving to a value slightly greater than 90%, while Spray And Wait starts with a low value, and grows as expected, althoug remaining under 90% in the case of 20 nodes.

Figure 6.17 shows the average overhead as the number of nodes increases. This value generally increases as the number of nodes increases, with an exception in the case of Spray and Wait and GeoSaW. The reason is that, while Spray and Wait contains a max threshold value (the max number of copies), for GeoSaW the number of packets allowed is

Table 6.6: Simulation parameters of the DTN environment.

Parameter	Value
MAC type	IEEE 802.11g, FreeSpace model
Simulation area	4000 m x 4000 m
Simulation time	60 min
Transmission range	500 m
Transmission speed	1 Mbps
Buffer size	50 MB
Packet size	250 KB
Packet TTL	30 min
Packet creation interval	30 s
Number of scanning UAVs	5
Number of delivery UAVs	5, 10, 15, 20
Scanning UAVs mobility model	Scan
Delivery UAVs mobility model	RWP

just 1. The other protocols do not have a packet number limitation, allowing an indefinite overhead growing. GeoSaW presents the best results in terms of overhead, since it uses just one packet copy.

Figure 6.18 shows the average latency as the number of nodes increases. GeoSaW presents medium results, in particular better than Spray And Wait. The average time difference between Epidemic values and GeoSaW values is about 120 s. This difference is due to the limitation of GeoSaW path prediction: when a node meets another node, it knows the entire future path of that node, but it is not able to predict future connections with other nodes that could deliver the packet faster; such ideal case could present the same performance as Epidemic in terms of latency.

6.6.4 Parameter variation comparison

In this experiment, we analyze in detail our protocol when varying several parameters. The first parameter is the number of packet copies allowed, ranging from 1 to 3. The second parameter is the permission to keep or to delete the packet copy in the sending node when it is forwarded. In total, we have 6 combinations. The parameter combination used in the general comparison (previous section) is with one copy with deletion of the packet in the sending node when it is forwarded.

As illustrated in Figure 6.19, the delivery ratio for all the combinations is very high, with a slight decline in the case of a single packet copy. In fact, even when considering the values of the confidence intervals, the delivery ratio does not provide us with enough

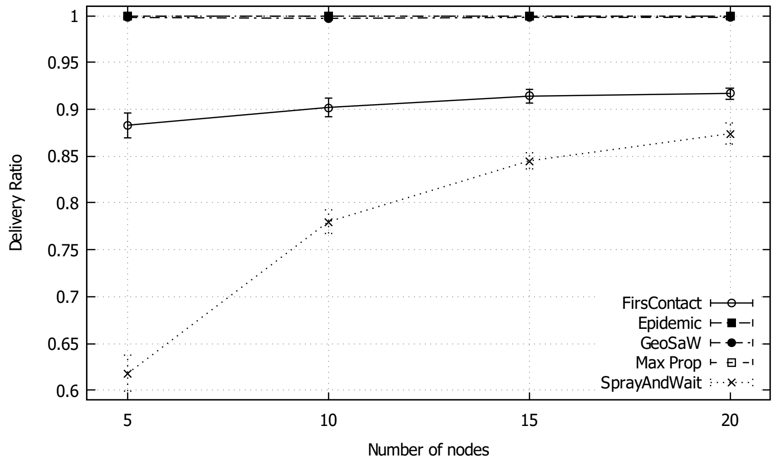


Figure 6.16: Average delivery ratio of GeoSaW compared with other DTN routing protocols when varying the number of nodes.

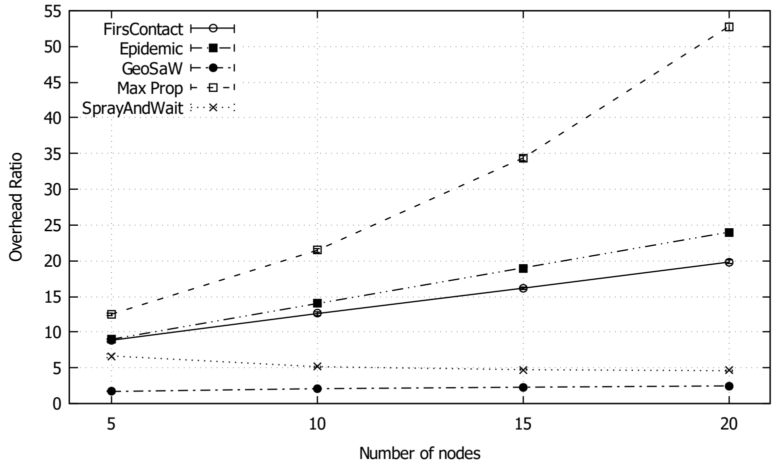


Figure 6.17: Average overhead ratio of GeoSaW compared with other DTN routing protocols when varying the number of nodes.

information about the parameter combination differences.

Figure 6.20 shows the overhead, which increases with the number of nodes. The one copy with packet deletion version performs better than the other versions, slightly exceeding the value 2 in the cases of 15 and 20 nodes. We notice a faster overhead increase in the case of “no deletion” with respect to “deletion”; this is due to the absence of packet deletion after the forwarding, resulting in a rapid increase of packet copies when the number of nodes (and hence the number of forwarders) increases.

Finally, the packet latency is shown in Figure 6.21. As expected, the copy with deletion

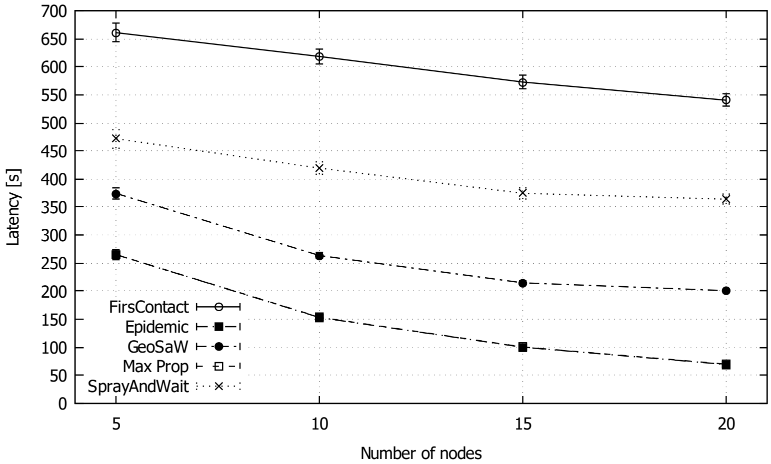


Figure 6.18: Average latency of GeoSaW compared with other DTN routing protocols when varying the number of nodes.

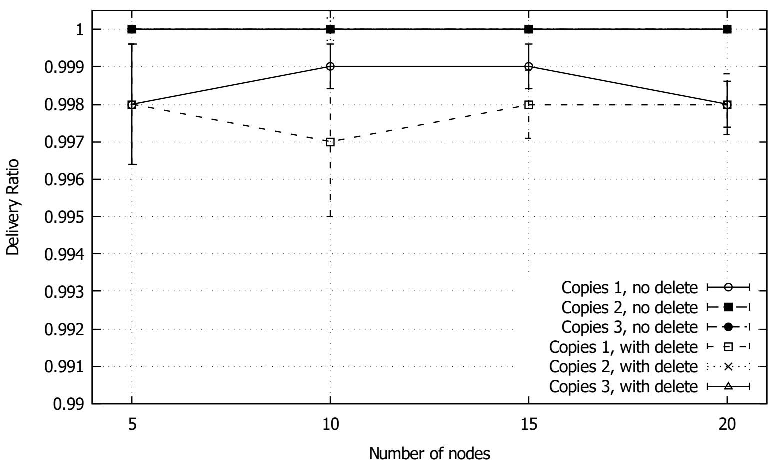


Figure 6.19: Average delivery ratio of GeoSaW with different parameters combinations when varying the number of nodes.

combination is the worst case, while all the other combinations achieve good values, less than 150 s in the case of 20 nodes. The number of copies could be chosen based on the requirements of the specific application that can tolerate a maximum packet delivery delay.

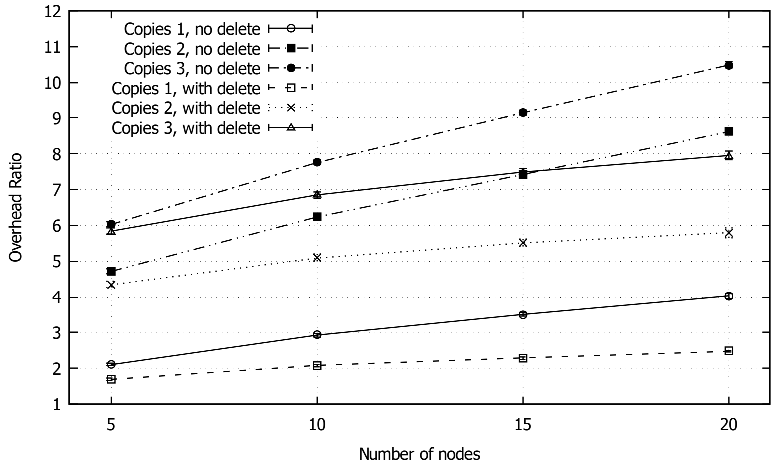


Figure 6.20: Average overhead ratio of GeoSaW with different parameters combinations when varying the number of nodes.

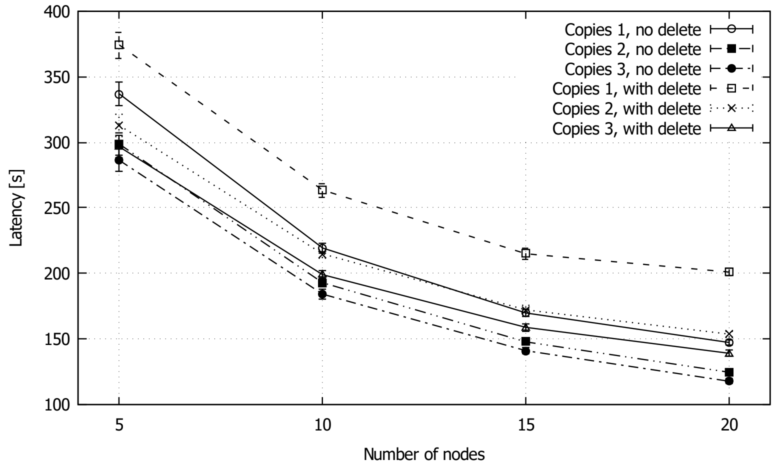


Figure 6.21: Average latency of GeoSaW with different parameters combinations when varying the number of nodes.

Chapter 7

People Perception Issues

In this chapter, we give a critical discussion about the possible impact of people feeling and preference about the presence of UAVs, and how these aspect may affects the characteristics of a deployed FANET in terms of communication performance. People perception and preference can be achieved by a series of interviews composed by targeted questions, able to find out relevant information regarding available UAV positioning, speed, movement, etc., in order to reach the least impact to the people itself. Our discussion is based on the work presented in [151].

7.1 UAVs from the Population Point of View

When a possible FANET is deployed over a city or another context, the UAVs need to interact between them but also with the environment. Considering situations in which UAVs fly over a populated area, there could be problems in terms of privacy, security and people perception, which led the creation of some drone regulations.

People impact, drone regulations and other aspects, may affect the way in which a FANET can be deployed over a specified area, because each UAV has to be positioned an move under determined conditions. Hence, to better inform privacy and security enhancing legislation to regulate where UAVs can go and what data they can collect, store, and disseminate, we first need to understand how users currently perceive UAVs, their purposes, and capabilities.

7.1.1 The Privacy and Security Problems

UAV can vary size and capacity from toy UAVs to military UAVs. Initially favored for military purposes, UAVs are now also used in private sector, law enforcement and hobbyist.

UAVs have various sensors, like camera for real-time video. For this reason, their usage may affect privacy because they have the ability to collect, retain, use and disclose personal information. Many studies focused on how users sense their security and privacy around UAVs. Many people feel UAVs could violate privacy if there is an unwanted intrusion into private spaces. But if people are aware of what the UAV was being to use (emergency, search and rescue, urban monitoring) they do not fear. Some studies [151] found that participants had a neutral attitude towards UAVs and did not consider UAVs to be overly unsafe, risky, beneficial, or threatening. In the US, a same study results were not definitive because users were largely unfamiliar with UAVs and still forming impressions. However, most users said benefits of UAVs outweigh risks.

7.2 Negative Perception of UAVs

In this section, we describe which types of negative impact the people may be affected by the UAVs presence.

7.2.1 UAVs as Privacy Invasive

Participants in several studies told that UAVs could be used for spying, or recording without consent. Unlike prior works, participants also mentioned not knowing where a drone was looking. People describes the house's windows the first element of UAVs privacy invasion. Another issue is the potential to be recorded either in public or in their own homes through a window and not knowing that the recording was happening. If UAVs are deployed by institutions for surveillance, the people may be affected by disclosing personal information.

7.2.2 UAVs Cause Fear of Damage

People may be worried about physical harm or death because of UAVs malfunctioning and falling out of the sky. Or they may be just surmised damage or injury from collisions could be an issue before they saw the drone or model for the first time. The UAVs' capability to carry items may lead to other fear about accidental falling down of such items. It is clear that there is need of some air traffic safety.

7.3 Interview Key Aspects

In the following, we provide a set of preliminary targeted metrics able to understand how the people would accept the presence of UAVs. From their responses, it is possible

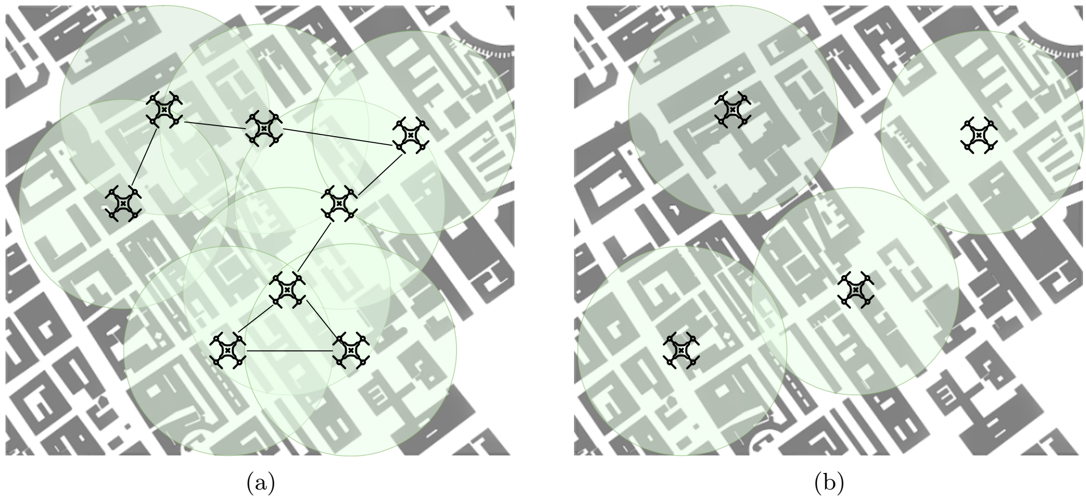


Figure 7.1: Illustration of a scenario in which (a) an arbitrary amount of UAVs fly on a residential area, and (b) the same scenario with a restriction of number of UAVs.

to make guidelines to the positioning and behavior of UAVs over a urban scenario. A possible set of interviews should take these metrics into account to a subsequent awareness of FANET displacement. We analyze these key aspect in order to understand how the FANET communication infrastructure may be affected.

UAVs density/number. People may put limits on the number of UAVs in a vicinity, for example to minimize the risk of UAVs collision. Or also, someone could be scared looking at many UAVs at the same time. A restriction may be a maximum allowed number of UAVs per unit area or volume.

The reduction of density of UAVs may reduce the communication chances, because of the reduction of available wireless links between UAVs, also with the possibility to make a FANET a DTN. In Fig. 7.1, a representation of this problem is illustrated. A very low UAV density may lead to a DTN, with no instantaneous end-t-end paths.

UAVs speed and sound. The speed and sound of UAVs may affect people in several ways. High speed movement may cause fear and noise perception from a civilian's point of view. Sound produced by UAV's flying may significantly disturb and annoy. There should be more stringent licensing requirements because UAVs could injure people when flying fast, scare or make people nervous if they fly too close to them, or if they fly in large numbers or in small spaces.

Based on the people response, UAVs may be forced to fly slowly. In the case of a DTN,

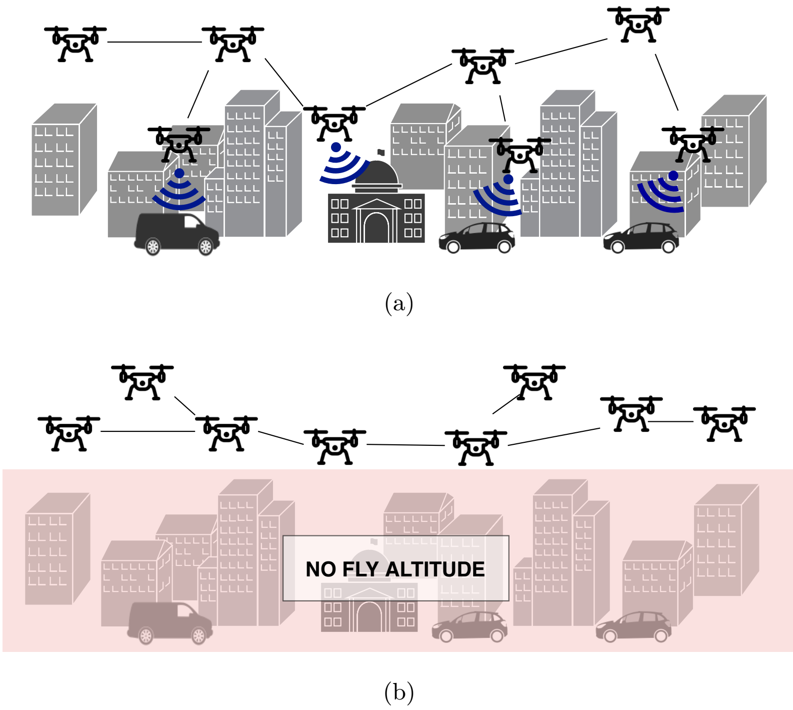
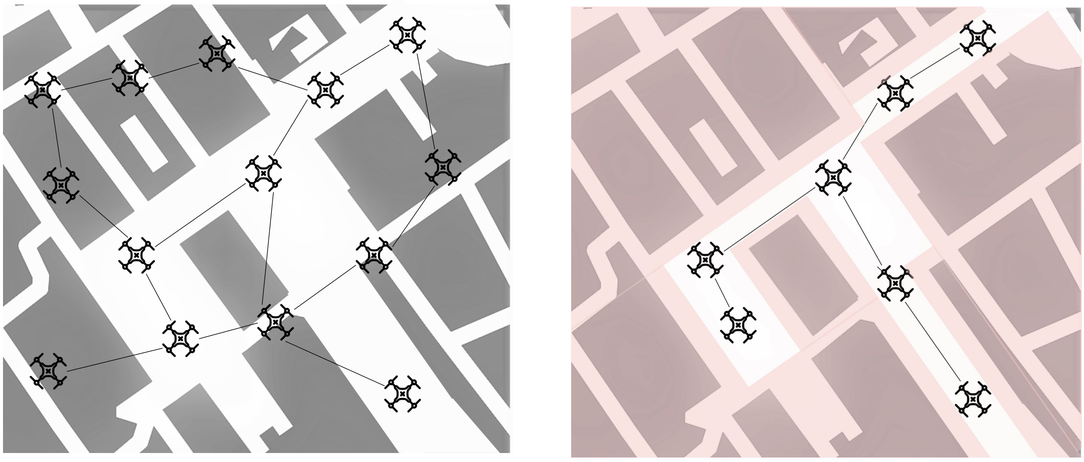


Figure 7.2: Illustration of a scenario in which (a) UAVs fly freely at different altitudes, according to the involved application, and (b) the same scenario with minimum altitude restriction.

this obligation could slow down the movement of any UAV carrying operation information. Consequently, the communication and cooperation are negatively affected, slowing down the entire task given by the operation control.

UAVs altitude. Altitude is perceived differently from people. Someone might feel that UAVs could not zoom in from high altitudes, and if there is no need of low distance from the ground, UAVs could easily assume high altitudes. On the other hands, other people could recommend UAVs fly at a different altitude to better separate them out from people and to minimize crashes into other UAVs.

The high altitude could be a problem when UAVs have to communicate with ground devices. If the transmission range of their communication apparatus is not sufficient to cover the high distance, the flying device has to physically move towards the destination receiver. In this case, a FANET may assume opportunistic characteristics (DTN). For example, in Fig. 7.2, a case with a minimum altitude restriction, results UAVs cannot always directly connect with ground devices. If different UAVs altitudes is preferred, we obtain a 3D network, leading to a few UAVs density and hence some routing problems.



(a)   (b)

Figure 7.3: Illustration of FANET scenario where (a) UAVs can be positioned and move freely over the buildings, and (b) where there is no-fly areas (red color).

UAVs proximity in Buildings or Sensible Areas. People may feel invaded if UAVs fly in public or private spaces, close to their houses or office buildings. Some consideration should be taken when concerning of UAVs looking into windows, or whether an UAV could turn around fast enough without running into a building and cause infrastructure damage to electrical components or balconies. The population may agree that UAVs should not be allowed to fly near residences, especially in densely populated areas, and that they should be far enough from buildings that they could not record sensitive data. Another problem is the fall down risk: if UAVs fly over populated areas, a possible crash or malfunctioning may lead damage to the beneath people. A solution may avoid pedestrian ways and sidewalks, and use highways so that even if UAVs fell, people would be protected in their cars.

A minimum distance from building and sensible areas introduces some constraints in the FANET displacement. The allowed area is reduced to low dense building ones. Therefore, buildings and sensible areas are, in this case, represented by obstacles to bypass, enhancing communication issues. An example is drawn in Fig. 7.3, where UAVs cannot freely move, and the FANET has to assume a limited topology shape.

Chapter 8

Conclusions

In this work, we focused on routing issues and solutions in a FANET architecture supporting potential application scenarios. Particular focus is given to the context of IoT and IoV, comparing routing protocols and analyzing their behavior in innovative scenarios, like smart cities. A compherensive state-of-the-art of position-based routing has depeened on 3D networks. We started by reviewing the underlying techniques describing the forwarding criteria, discussing the problems and limitations that emerge. From this point, we proposed our solutions that allow an improvement of routing performance, making the nodes aware of their geographic location information. Indeed, using context-aware strategies, it is possible to provide useful information in order to support packet forwarding, allowing a more precise next node selection. With the advent of GPS, the location information detection is easy, low-cost and getting more efficient, and it is a good point of view for the design of innovative networking technology. Our work outlines us positive results in terms of routing performance, by applying locationinformation on routing strategies. We claim that more work should be done to deeply investigate about this resource, specially considering the recent advent of smart environments, which rely on location information in a lot of scenarios.

8.1 Summary of Results

The high dynamic nature of FANET demands for networking techniques able to tackle the resulting issues, especially in routing. In this context, we started by exploring position-based routing protocols and performing a critical analysis, as well as running them in diverse simulation scenarios. By changing different network characteristics (node density, node speed, etc.), we could figure out which routing protocol may be adequate for a particular situation.

From a compherensive comparative evaluation we found that deterministic progress-based strategies perform well in very dense networks, but not in a sparse networks, due to the problem of local minima. Algorithms that use this strategies, for instance Greedy, may be used in combination with other algorithms, in order to reduce effectively the number of nodes traveled. The random component in a forwarding decision offers a better chance to reach the destination. In this work, we have recovered the randomization-based algorithms AB3D, with different input parameters. With a better combination of possible parameters, AB3D reaches a delivery rate of 80% in one of the worst case (150 nodes), with a path length of at most three times the minimum path length in the considered scenarios. Hybrid solutions seem to achieve better performance; for example, combining Greedy with AB3D (getting GRG), a slight improvement of packet delivery and path dilation can be observed. Face-based strategies are able to reach very high values of performance in packet delivery, with the cost of a very high path length. The attempt to hybridize ALSP Face with Greedy (getting GFG) has in part raised from the heavy traffic. However, the application of algorithms that use 2D concepts, as the planarization of a graph, does not seem to be a very efficient idea applied in 3D environments. The number of crossing links is significant when the graph is projected, and the followed path is not always *towards* the target node, unlike the 2D case, generating unnecessary traffic.

We have explored some basic behaviors of topology-based and position-based routing protocols on a variety of network graphs that represent a possible IoV scenario. The considered topologies are different by number of nodes and by pause times. Our results shed lights on which are the open technical challenges in routing messages over an IoV. More in detail, we have noticed that topology-based protocols such as AODV and DSR achieve acceptable performance in terms of both delivery rate and path dilation, whereas position-based protocols achieve higher data rates in high density scenarios but at the cost of a large path dilation. In general, these tests assessed the efficacy of state-of-the-art topology-based protocols in supporting general data transmissions over an IoV, with no one capable of providing any delivery guarantee. We believe that this would be crucial to support applications in IoV scenarios (e.g., to support safety and distributed control for automated vehicles, or just for entertainment applications) and we hence encourage researchers in devising new routing solutions specifically designed for this purpose.

Analyzing the literature about routing protocols, we found out the lack of adequate mobility models for specific simulation scenarios. Mobility model selection strongly depends on the type of simulation and application scenario that is involved in a FANET analysis. The performance of a FANET (e.g., packet delivery and packet delay) can vary significantly with different mobility models, and the choice of a suitable one is crucial for critical applications. In this disseration, we have explored several mobility models, extracting their advantages and disadvantages in terms of motion realism, randomization, network connectivity and collision avoidance. Pure randomized mobility models are trivial and too unrealistic for applications that engage flying devices. Mobility models that include smooth turns and speed changes can reproduce more realistically these randomized movements. We explored some topology-control based mobility models, which include a mission-based movement for UAVs. Furthermore, we have associated the most fitting mobility model with each application scenario, so as to provide guidelines to researchers creating simulation experiments about drone ad-hoc networks.

Next, we presented a hybrid, position and topology-based routing approach for 3D topology networks, which tries to strike a balance between a greedy and reactive approach to packet forwarding. G-CR starts with a greedy approach, falling back to a reactive, position-based approach whenever the process is stuck in a local minimum. The evaluation analysis shows the benefits of the approach when compared to pure position-based solutions, while at the same time achieving better performance trend with respect to AODV in high density scenarios.

Finally, we have designed, implemented and compared a DTN routing protocol that relies on the planned UAV's path to make the message forwarding decision. We analyzed its performance results, showing advantages in terms of delivery ratio and overhead. However, it still suffers from significant delays. Indeed, in DTNs, the major issue is the huge delays involved, which are not desirable in search and rescue and other emergency scenarios. In our work, we noticed long delays in the message delivery, even if the forwarding algorithm tries to reduce the latency by forwarding the message to the fastest UAV.

8.2 Future Work

Extending the current analysis, we plan to compare the various protocols considering network density and network size which vary independently so as to show how modifying only one of them could affect the performance [152]. An additional next step would be that of studying the impact of mobility in network performance under varying network conditions. Moving towards a more realistic UAV-to-UAV communication model, we need to consider an antenna radiation pattern reflecting a toroid shape rather than a sphere currently employed in the simulation study. Another interesting research direction would be the study of a targeted position modification scheme (e.g., using operations research analysis) whereby nodes autonomously adjust their position in order to augment the chances of packet delivery.

Further research on mobility models can of course be done. Randomized mobility models can be modified with the inclusion of smooth characteristics in order to add a realistic flight behavior. Another research branch can be devoted to a deeper examination of flying devices' motion in the real world to reproduce more accurate mobility models. A good approach is the attempt to combine the existing mobility models, in order to merge the best movement characteristic of each of them. For example, the MG mobility model could include additional features coming from the ST mobility model, to reproduce more smooth curves when an UAV approaches a crossroad. Also, we noticed that, to the best of our knowledge, there are no mobility models that consider collisions against buildings or other external obstacles. For example, a randomized mobility model that considers prohibited flying areas can implement such scenarios. A particular consideration could be taken in this direction. Finally, we woud also like to investigate how different mobility models and routing strategies may affect the QoS of the considered application [153, 154].

In DTN context, we plan to tackle the problem of long latency, which is still significant in our proposal, and find a way to minimize it using additional UAVs, for example deploying an ad-hoc optimized path that reaches the recipient when necessary [113, 155]. Furthermore, we also plan to implement our protocol in the Network Simulator 3, where the wireless communication aspects and protocols are more realistically modelled.

Appendix A

List of Publications

Below are enumerated the scientific publications of the candidate related to the research activities discussed throughout this dissertation.

International Conferences and Workshop

(C1) Armir Bujari, Luca Busato, Claudio E. Palazzi, **Daniele Ronzani**, "Benchmarking of Routing Algorithms in 3D MANETs", in Proc. of the *4th ACM Workshop on Micro Aerial Vehicle Networks, Systems, and Applications* (DroNet'18). ACM, New York, NY, USA, 2018.

(C2) Armir Bujari, Claudio E. Palazzi, **Daniele Ronzani**, "Comparing Routing Protocols over a 3D IoT", in Proc. of the *Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing* (Mobihoc '18), ACM, New York, NY, USA, 2018.

(C3) Armir Bujari, Claudio E. Palazzi, **Daniele Ronzani**, "A Hybrid Reactive and Position-based Approach to Packet Routing in 3D Topology Networks", in Proc. of *2018 Wireless Days* (WD'18), Dubai, United Arab Emirates, April 2018.

(C4) Armir Bujari, Claudio E. Palazzi, **Daniele Ronzani**, "FANET Application Scenarios and Mobility Models", in Proc. of *the 3rd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications* (DroNet '17), ACM, New York, USA, June 2017.

(C5) Stefano Munari, Claudio E. Palazzi, Giacomo Quadrio, **Daniele Ronzani**, "Network Traffic Analysis of a Small Quadcopter", in Proc. of *the 3rd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications* (DroNet '17), ACM, New York, USA, June 2017.

(C6) Giacomo Quadrio, Armir Bujari, Claudio E. Palazzi, **Daniele Ronzani**, Dario Maggiorini, Laura Anna Ripamonti, "Network analysis of the steam in-home streaming game system" (poster), in Proc. of *the 22nd Annual International Conference on Mobile Computing and Networking* (MobiCom), ACM, New York, NY, USA, 2016.

(C7) Giacomo Quadrio, Armir Bujari, Claudio E. Palazzi, **Daniele Ronzani**, Dario Maggiorini, Laura Anna Ripamonti, "Network analysis of the Sony Remote Play system", in Proc. of *IEEE Symposium on Computers and Communication* (ISCC), Messina, Italy, 2016.

(C8) Armir Bujari, Claudio E. Palazzi, **Daniele Ronzani**, "Multimedia transmissions over vehicular networks", in Proc. of *IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications* (PIMRC), Valencia, Spain, 2016.

International Journals

(J1) Armir Bujari, Carlos T. Calafate, Juan-Carlos Cano, Pietro Manzoni, Claudio E. Palazzi, **Daniele Ronzani**, "Location-aware Waypoint-based Routing Protocol for Airborne DTNs", *Sensors*, submitted, 2018.

(J2) Armir Bujari, Ombretta Gaggi, Claudio E. Palazzi, **Daniele Ronzani**, "Would Current Ad Hoc Routing Protocols be Adequate for the Internet of Vehicles? A Comparative Study", *IEEE Internet of Things Journal*, March 2018.

(J3) Armir Bujari, Claudio E. Palazzi, **Daniele Ronzani**, "A Comparison of Stateless Position-based Packet Routing Algorithms for FANETs", *IEEE Transactions on Mobile Computing*, March 2018.

(J4) Armir Bujari, Carlos T. Calafate, Juan-Carlos Cano, Pietro Manzoni, Claudio E. Palazzi, **Daniele Ronzani** "Flying ad-hoc network application scenarios and mobility models", *International Journal of Distributed Sensor Networks*, vol. 13, no. 10, Oct 2017.

(J5) Armir Bujari, Marco Furini, Federica Mandreoli, Riccardo Martoglia, Manuela Montangero, **Daniele Ronzani**, "Standards, Security and Business Models: Key Challenges for the IoT Scenario", *Springer Mobile Networks and Applications*, vol. 23, no. 1, Feb 2017.

Bibliography

[1] S. Basagni et al. “Mobile Ad hoc Networking”. In: *Jonh Wiley & Sons* (2004).

[2] F. M. Adbuljalil and S. K. Bodhe. “A Survey of Integrating IP Mobility Protocols and Mobile Ad-hoc Networks”. In: *IEEE Communications Surveys & Tutorials* 9.1 (2007), pp. 14–30.

[3] Dario Floreano and Robert J. Wood. “Science, technology and the future of small autonomous drones”. In: *Nature* 521.7553 (2015), pp. 460–466.

[4] S. M. R. Islam et al. “The Internet of Things for Health Care: A Comprehensive Survey”. In: *IEEE Access* 3 (2015), pp. 678–708.

[5] X. Hu, L. Yang, and W. Xiong. “A Novel Wireless Sensor Network Frame for Urban Transportation”. In: *IEEE Internet of Things Journal* 2.6 (2015), pp. 586–595.

[6] J. An et al. “A Crowdsourcing Assignment Model Based on Mobile Crowd Sensing in the Internet of Things”. In: *IEEE Internet of Things Journal* 2.5 (2015), pp. 358–369.

[7] A. Zanella et al. “Internet of Things for Smart Cities”. In: *IEEE Internet of Things Journal* 1.1 (2014), pp. 22–32.

[8] G. Betis et al. “The ieee smart cities initiative — accelerating the smartification process for the 21st century cities [point of view]”. In: *Proceedings of the IEEE* 106.4 (2018), pp. 507–512.

[9] M. Asadpour et al. “Micro aerial vehicle networks: an experimental analysis of challenges and opportunities”. In: *IEEE Communications Magazine* 52.7 (2014), pp. 141–149.

[10] I. Bekmezci, O.K. Sahingoz, and Ş. Temel. “Flying Ad-Hoc Networks (FANETs): A survey”. In: *Ad Hoc Networks* 11.3 (2013), pp. 1254–1270.

[11] J. Xie et al. “A Survey and Analysis of Mobility Models for Airborne Networks”. In: *IEEE Communications Surveys & Tutorials* 16.3 (2013), pp. 1221–1238.

[12] M. Quaritsch et al. "Networked UAVs as aerial sensor network for disaster management applications". In: *Elektrotechnik und Informationstechnik* 127.3 (2010), pp. 56–63.

[13] Intelligence Digital Magazine. "INTELLIGENCE: Better Living Through Drones". In: *https://www.theintelligenceofthings.com/article/better-living-through-drones/* (2015). (Visited on 03/28/2017).

[14] *Six Ways Drones Are Revolutionizing Agriculture*. URL: https://www.technologyreview.com/s/601935/six-ways-drones-are-revolutionizing-agriculture/.

[15] Corrado Rametta and Giovanni Schembra. "Designing a Softwarized Network Deployed on a Fleet of Drones for Rural Zone Monitoring". In: *Future Internet* 9.1 (2017).

[16] J. Harri, F. Filali, and C. Bonnet. "Mobility Models for Vehicular Ad-hoc Networks: a Survey and Taxonomy". In: *IEEE Communications Surveys & Tutorials* 11.4 (2009), pp. 19–41.

[17] G. Karagiannis et al. "Vehicular Networking: A Survey and Tutorial on Requirements, Architectures, Challenges, Standards and Solutions". In: *IEEE Communications Surveys & Tutorials* 13.4 (2011), pp. 584–616.

[18] I.F. Akyildiz et al. "Wireless Sensor Networks: a Survey". In: *Computer Networks* 38.4 (2001), pp. 393–422.

[19] I. F. Akyildiz and I. H. Kasimoglu. "Wireless Sensor and Actor Networks: research challenges". In: *Ad Hoc Networks* 2.4 (2004), pp. 351–367.

[20] *Unmanned Aerial Vehicle*. URL: http://www.thefreedictionary.com/Unmanned+Aerial+Vehicle.

[21] Jürgen Scherer et al. "An autonomous multi-UAV system for search and rescue". In: *DroNet 2015* (2015), pp. 33–38.

[22] A. Purohit et al. "Deployment of Swarms of Micro-Aerial Vehicles: from Theory to Practice". In: *In Proc. of IEEE ICRA* (2014).

[23] Ozgur Sahingoz. "Networking Models in Flying Ad-Hoc Networks (FANETs): Concepts and Challenges". In: *Journal of Intelligent & Robotic Systems* 74 (Apr. 2014).

[24] Evsen Yanmaz, Robert Kuschnig, and Christian Bettstetter. "Achieving air-ground communications in 802.11 networks with three-dimensional aerial mobility". In: *Proceedings - IEEE INFOCOM* (2013), pp. 120–124.

[25] Yao-Hua Ho, Yu-Ren Chen, and Ling-Jyh Chen. “Krypto: Assisting Search and Rescue Operations Using Wi-Fi Signal with UAV”. In: *Proceedings of the First Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use (DroNet)* (2015), pp. 3–8.

[26] Sonia Waharte and Niki Trigoni. “Supporting search and rescue operations with UAVs”. In: *Proceedings - EST 2010 - 2010 International Conference on Emerging Security Technologies, ROBOSEC 2010 - Robots and Security, LAB-RS 2010 - Learning and Adaptive Behavior in Robotic Systems* (2010), pp. 142–147.

[27] Edison Pignaton De Freitas et al. “UAV relay network to support WSN connectivity”. In: *2010 International Congress on Ultra Modern Telecommunications and Control Systems and Workshops, ICUMT 2010*. IEEE, 2010, pp. 309–314.

[28] Pengcheng Zhan, Kai Yu, and A. Lee Swindlehurst. “Wireless relay communications with unmanned aerial vehicles: Performance and optimization”. In: *IEEE Transactions on Aerospace and Electronic Systems* 47.3 (2011), pp. 2068–2085.

[29] *Swarming Micro Air Vehicle Network (SMAVNET II)*. URL: http://smavnet.epfl.ch/.

[30] Milan Erdelj, Michał Król, and Enrico Natalizio. “Wireless Sensor Networks and Multi-UAV systems for natural disaster management”. In: *Computer Networks* 124 (2017), pp. 72 –86.

[31] M. Erdelj et al. “Help from the Sky: Leveraging UAVs for Disaster Management”. In: *IEEE Pervasive Computing* 16.1 (2017), pp. 24–32.

[32] Nicola Roberto Zema et al. “MeDrone: On the use of a medical drone to heal a sensor network infected by a malicious epidemic”. In: *Ad Hoc Networks* 50 (2016), pp. 115 –127.

[33] Luis Merino et al. “A cooperative perception system for multiple UAVs: Application to automatic detection of forest fires”. In: *Journal of Field Robotics* 23.3-4 (2006), pp. 165–184.

[34] Khaled A Ghamry and Youmin Zhang. “Fault-Tolerant Cooperative Control of Multiple UAVs for Forest Fire Detection and Tracking Mission”. In: *2016 3rd Conference on Control and Fault-Tolerant Systems (SysTol)*. 2016, pp. 133–138.

[35] Tariq Samad, John S. Bay, and Datta Godbole. “Network-centric systems for military operations in urban terrain: The role of UAVs”. In: *Proceedings of the IEEE* 95.1 (2007), pp. 92–107.

[36] Per Magnus Olsson et al. “Generating UAV communication networks for monitoring and surveillance”. In: *11th International Conference on Control, Automation, Robotics and Vision, ICARCV 2010*. IEEE, 2010, pp. 1070–1077.

[37] R Reshma, Tirumale Ramesh, and P. Sathishkumar. “Security situational aware intelligent road traffic monitoring using UAVs”. In: *2016 International Conference on VLSI Systems, Architectures, Technology and Applications (VLSI-SATA)*. IEEE, 2016, pp. 1–6.

[38] Wajiya Zafar and Bilal Muhammad Khan. “Flying Ad-Hoc Networks: Technological and Social Implications”. In: *IEEE Technology and Society Magazine* 35.2 (2016), pp. 67–74.

[39] Erik Kuiper and Simin Nadjm-Tehrani. “Mobility models for UAV group reconnaissance applications”. In: *Second International Conference on Wireless and Mobile Communications, ICWMC 2006* (2006).

[40] M. Tortonesi et al. “Multiple-UAV Coordination and Communications in Tactical Edge Networks”. In: *IEEE Communications Magazine* 50.10 (2012), pp. 48–55.

[41] T. Samad, J. S. Bay, and D. Godbole. “Network-Centric Systems for Military Operations in Urban Terrain: The Role of UAVs”. In: *In Proc. of IEEE* 95.1 (2007), pp. 92–107.

[42] Jorge Torres-Sánchez et al. “Configuration and Specifications of an Unmanned Aerial Vehicle (UAV) for Early Site Specific Weed Management”. In: *PLoS ONE* 8.3 (2013). Ed. by Derek Abbott, e58210.

[43] Haiyang Chao et al. “Band-reconfigurable Multi-UAV-based Cooperative Remote Sensing for Real-time Water Management and Distributed Irrigation Control”. In: *IFAC Proceedings Volumes* 41.2 (2008), pp. 11744–11749.

[44] Xiaohui Li et al. “A Hybrid PSO Algorithm Based Flight Path Optimization for Multiple Agricultural UAVs”. In: *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2016, pp. 691–697.

[45] Peng Wei, Quanquan Gu, and Dengfeng Sun. “Wireless sensor network data collection by connected cooperative UAVs”. In: *2013 American Control Conference*. IEEE, 2013, pp. 5911–5916.

[46] Oscar Alvear et al. “Mobile Pollution Data Sensing Using UAVs”. In: *The 13th International Conference on Advances in Mobile Computing and Multimedia* (2015), pp. 393–397.

[47] Catia Prandi, Paola Salomoni, and Silvia Mirri. “mPASS: Integrating People Sensing and Crowdsourcing to Map Urban Accessibility”. In: *Proceedings of the IEEE International Conference on Consumer Communications and Networking Conference* (2014), pp. 10–13.

[48] Silvia Mirri et al. “On combining crowdsourcing, sensing and open data for an accessible smart city”. In: *Proceedings - 2014 8th International Conference on Next Generation Mobile Applications, Services and Technologies, NGMAST 2014*. IEEE, 2014, pp. 294–299.

[49] Stefan Poslad. *Ubiquitous computing: smart devices, environments and interactions*. John Wiley & Sons, 2011.

[50] A. Bujari and C. E. Palazzi. “AirCache: A Crowd-Based Solution for Geoanchored Floating Data”. In: *Mobile Information Systems* 2016 (2016).

[51] G. Marfia et al. “Safe Driving in LA: Report from the Greatest Intervehicular Accident Detection Test Ever”. In: *IEEE Transactions on Vehicular Technology* 62.2 (2013), pp. 522–535.

[52] C. E. Palazzi, M. Roccetti, and S. Ferretti. “An Intervehicular Communication Architecture for Safety and Entertainment”. In: *IEEE Transactions on Intelligent Transportation Systems* 11.1 (2010), pp. 90–99.

[53] A. Nandan et al. “Co-operative Downloading in Vehicular Ad-hoc Wireless Networks”. In: *Proc. of IEEE/IFIP WONS*. 2005, pp. 32–41.

[54] Gustavo Marfia, Marco Roccetti, and Alessandro Amoroso. “A New Traffic Congestion Prediction Model for Advanced Traveler Information and Management Systems”. In: *Wireless Communications and Mobile Computing* 13.3 (2013), pp. 266–276.

[55] Armir Bujari. “A Network Coverage Algorithm for Message Broadcast in Vehicular Networks”. In: *Mobile Networks and Applications* 21.4 (2016), pp. 668–676.

[56] L. Bononi, L. Donatiello, and M. Furini. “Real-Time Traffic in Ad-Hoc Sensor Networks”. In: *Proc. of IEEE ICC*. 2009, pp. 1–5.

[57] D. Floreano and Wood J. R. “Science, Technology and the Future of Small Autonomous Drones”. In: *Nature* 521 (2015), pp. 460–466.

[58] Julien Fleureau et al. “Generic Drone Control Platform for Autonomous Capture of Cinema Scenes”. In: *Proc. of ACM MobiSys/DroNet*. Singapore, Singapore, 2016, pp. 35–40.

[59] A. Bujari et al. “Geo-anchored Floating Data for Mobile Users”. In: *Proc. of IEEE ICMEW*. 2013, pp. 1–5.

[60] A. Boukerche et al. “Routing Protocols in Ad hoc Networks: A Survey”. In: *Computer Networks* 55.13 (2011), pp. 3032–3080.

[61] R. Karthikeyan, K. Sasikala, and R. Reka. “A Survey on Position-based Routing in Mobile Ad hoc Networks”. In: *International Journal of P2P Network Trends and Technology* 31 (2013).

[62] M. Mauve, J. Widmer, and H. Hartenstein. “A Survey of Position-based Routing in Mobile Ad hoc Networks”. In: *IEEE Network Magazine* 15.6 (2001), pp. 30–39.

[63] I. Stojmenovic. “Position-Based Routing in Ad Hoc Networks”. In: *IEEE Communications Magazine* 40.7 (July 2002), pp. 128–134.

[64] A. H. Iche and M. R. Dhage. “Location-based Routing Protocols: A Survey”. In: *International Journal of Computer Applications* 109 (2015).

[65] M. Heissenbuttel and T. Braun. “Optimizing Neighbor Table Accuracy of Position-based Routing Algorithms”. In: *In Proc. of IEEE INFOCOM* (2005).

[66] S. Durocher, D. Kirkpatrick, and L. Narayanan. “On Routing with Guaranteed Delivery in Three-Dimensional Ad hoc Wireless Networks”. In: *In Proc. of ICDCN* (2008), pp. 546–557.

[67] J. Li et al. “A Scalable Location Service for Geographic Ad hoc Routing”. In: *In Proc. of ACM MOBICOM* (2000), pp. 120–130.

[68] Jianqi Liu et al. “A survey on position-based routing for vehicular ad hoc networks”. In: *Telecommunication Systems* 62.1 (2016), pp. 15–30.

[69] Asis Nasipuri, Robert Castañeda, and Samir R. Das. “Performance of multipath routing for on-demand protocols in mobile Ad Hoc networks”. In: *Mobile Networks and Applications* 6.4 (2001), pp. 339–349.

[70] S. Basagni, I. Chlamtac, and V. Syrotiuk. “A Distance Routing Effect Algorithm for Mobility (DREAM)”. In: *In Proc. of MobiCom* (1998), pp. 76–84.

[71] Y. B. Ko and N. H. Vaidya. “Location-Aided Routing (LAR) in Mobile Ad hoc Networks”. In: *ACM/Baltzer Wireless Networks* 6.4 (2000), pp. 307–321.

[72] A. Bujari et al. “Would Current Ad-Hoc Routing Protocols be Adequate for the Internet of Vehicles? A Comparative Study”. In: *IEEE Internet of Things Journal* (2018), pp. 1–1.

[73] Gurkan Tuna, Bilel Nefzi, and Gianpaolo Conte. "Unmanned aerial vehicle-aided communications system for disaster recovery". In: *Journal of Network and Computer Applications* 41 (2014), pp. 27 –36.

[74] Kevin Fall. "A Delay-tolerant Network Architecture for Challenged Internets". In: *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*. SIGCOMM '03. Karlsruhe, Germany: ACM, 2003, pp. 27–34.

[75] C. E. Palazzi et al. "An overview of opportunistic ad hoc communication in urban scenarios". In: *2014 13th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET)*. 2014, pp. 146–149.

[76] Pietro Manzoni et al. "Epidgeons: Combining Drones and DTNs Technologies to Provide Connectivity in Remote Areas". In: *Proc. of ACM MobiSys/DroNet*. Florence, Italy, 2015, pp. 57–58.

[77] S. M. Tornell et al. "DTN Protocols for Vehicular Networks: An Application Oriented Overview". In: *IEEE Communications Surveys & Tutorials* 17, no. 2 (2015), pp. 868–887.

[78] T. Jonson et al. "Application of Delay Tolerant Networking (DTN) in Airborne Networks". In: *MILCOM 2008 - 2008 IEEE Military Communications Conference*. 2008, pp. 1–7.

[79] C. Giannini et al. "Delay Tolerant Networking for smart city through drones". In: *2016 International Symposium on Wireless Communication Systems (ISWCS)*. 2016, pp. 603–607.

[80] Radosław O Schoeneich et al. "Autonomous Drone for Delay-tolerant Networks in Indoor Applications". In: *International Journal of Distributed Sensor Networks* 12.8 (2016).

[81] C. Perkins and P. Bhagwat. "Highly Dynamic Destination-Sequenced Distance-Vector (DSDV) Routing for Mobile Computers". In: *In Proc. of ACM SIGCOMM* (1994), pp. 234–244.

[82] C. Perkins, E. Royer, and S. R. Das. "Ad hoc On-demand Distance Vector (AODV) Routing". In: *In Proc. of IEEE HotMobile* (1999), pp. 90–11.

[83] D. Johnson and D. Malts. "Dynamic Source Routing in Ad-hoc Wireless Networks". In: *T. Imielinski, H. Korth (Eds.), Mobile Computing, Kluwer Academic Publisher* (1996), pp. 153–181.

[84] M.G. Lewis F.L. Templin B. Bellur R.G. Ogier. "Topology Broadcast based on Reverse Path Forwarding (TBRPF)". In: *RFC 3684, IETF Network Working Group* (2002).

[85] P. Jacquet T. Clausen. "Optimized Link State Routing Protocol (OLSR)". In: *RFC 3626, IETF Network Working Group* (October 2003).

[86] Gregory G. Finn. "Routing and Addressing Problems in Large Metropolitan-scale Internetworks". In: *Technical Report ISU/RR-87-180, USC Information Sciences Institute (ISI)* (1987).

[87] I. Stojmenovic and X. Lin. "Loop-free Hybrid Single-path/flooding Routing Algorithms with Guaranteed Delivery for Wireless Networks". In: *IEEE Transactions on Parallel and Distribution Systems* 12.10 (2001), pp. 1023–1032.

[88] E. Kranakis, H. Singh, and J. Urrutia. "Compass Routing on Geometric Networks". In: *In Proc. of CCCG* (1999), pp. 51–54.

[89] H. Takagi and L. Kleinrock. "Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals". In: *IEEE Transactions on Communications* 32.3 (1984), pp. 246–257.

[90] K. Yamazaki and K. Sezaki. "A Proposal of Geographical Routing Protocols for Location-aware Services". In: *Electronics and Communications in Japan* 87.4 (2004).

[91] C. Liu and J. Wu. "Efficient Geometric Routing in Three Dimensional Ad hoc Networks". In: *In Proc. of IEEE INFOCOM* (2009), pp. 2751–2755.

[92] R. Nelson and L. Kleinrock. "The Spatial Capacity of a Slotted ALOHA Multihop Packet Radio Network with Capture". In: *IEEE Transactions on Communications* COM-32 (1984), pp. 684–694.

[93] P. Bose and P. Morin. "Online Routing in Triangulations". In: *In Proc. of ISAAC* (1999), pp. 113–122.

[94] T. Fevens, A. E. Abdallah, and B. N. Bennani. "Randomized AB-Face-AB Routing Algorithms in Mobile Ad hoc Networks". In: *In Proc. of ACM ADHOC-NOW* (2005).

[95] A. E. Abdallah, T. Fevens, and J. Opatrny. "Randomized 3D Position-based Routing Algorithms for Ad hoc Networks". In: *In Proc. of MOBIQUITOUS* (2006), pp. 1–8.

[96] R. Wattenhofer R. Flury. "Randomized 3D Geographic Routing". In: *In Proc. of IEEE INFOCOM* (2008).

[97] P. Bose et al. "Routing with Guaranteed Delivery in Ad hoc Wireless Networks". In: *In Proc. of ACM DIAL-M* (1999), pp. 48–55.

[98] K. R. Gabriel and R. R. Sokal. "A New Statistical Approach to Geographic Variation Analysis". In: *Systematic Zoology (Society of Systematic Biologists)* 18.3 (1969), pp. 259–270.

[99] G. Kao, T. Fevens, and J. Opatrny. "Position-based Routing on 3D Geometric Graphs in Mobile Ad hoc Networks". In: *In Proc. of CCCG* (2005), pp. 88–91.

[100] G. Kao, T. Fevens, and J. Opatrny. "3D Localized Position-based Routing with Nearly Certain Delivery in Mobile Ad hoc Networks". In: *In Proc. of IEEE ISWPC* (2007).

[101] B. Karp and H.T. Kung. "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks". In: *In Proc. of ACM MOBICOM* (2000), pp. 243–254.

[102] S. Liu, T. Fevens, and A. E. Abdallah. "Hybrid Position-based Routing Algorithms for 3D Mobile Ad hoc Networks". In: *In Proc. of IEEE MSN* (2008), pp. 177–186.

[103] Stefano Basagni et al. "A Distance Routing Effect Algorithm for Mobility (DREAM)". In: *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking.* MobiCom '98. Dallas, Texas, USA: ACM, 1998, pp. 76–84.

[104] A.E. Abdallah, T. Fevens, and J. Opatrny. "Hybrid Position-based 3D Routing Algorithms with Partial Flooding". In: *In Proc. of CCCG* (2006), pp. 1135–1138.

[105] A. E. Abdallah, T. Fevens, and J. Opatrny. "High Delivery Rate Position-based Routing Algorithms for 3D Ad hoc Networks". In: *Computer Communications* 31.4 (2007), pp. 807–817.

[106] I. Stojmenovic, I. Russell, and M. Vukojevic. "Depth First Search and Location Based Localized Routing and QoS Routing in Wireless Networks". In: *In Proc. of IEEE ICPP* (2000), pp. 173–180.

[107] Amin Vahdat and David Becker. *Epidemic Routing for Partially-Connected Ad Hoc Networks.* Tech. rep. Technical Report CS-2000-06, Department of Computer Science, Duke University, 2000.

[108] J. Burgess et al. "MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks". In: *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications.* 2006, pp. 1–11.

[109] Ari Keränen, Jörg Ott, and Teemu Kärkkäinen. “The ONE Simulator for DTN Protocol Evaluation”. In: *Proceedings of the 2Nd International Conference on Simulation Tools and Techniques*. Simutools ’09. Rome, Italy: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009, 55:1–55:10.

[110] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. “Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks”. In: *Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking*. WDTN ’05. Philadelphia, Pennsylvania, USA: ACM, 2005, pp. 252–259.

[111] K. Peters et al. “A geographical routing protocol for highly-dynamic aeronautical networks”. In: *2011 IEEE Wireless Communications and Networking Conference*. 2011, pp. 492–497.

[112] S. Krug and J. Seitz. “Challenges of applying DTN routing protocols in realistic disaster scenarios”. In: *2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*. 2016, pp. 784–789.

[113] Ruslan Kirichek, Alexander Paramonov, and Karine Vareldzhyan. “Optimization of the UAV-P’s Motion Trajectory in Public Flying Ubiquitous Sensor Networks (FUSN-P)”. In: *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*. Ed. by Sergey Balandin, Sergey Andreev, and Yevgeni Koucheryavy. Cham: applyi Springer International Publishing, 2015, pp. 352–366.

[114] A. Bujari et al. “A solution for mobile DTN in a real urban scenario”. In: *2012 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*. 2012, pp. 344–349.

[115] Armir Bujari et al. “Delay Tolerant Networking over the Metropolitan Public Transportation”. In: *Mobile Information Systems* 2016.10 (2016), pp. 1–14.

[116] M. Le, J. s. Park, and M. Gerla. “UAV Assisted Disruption Tolerant Routing”. In: *MILCOM 2006 - 2006 IEEE Military Communications conference*. 2006, pp. 1–5.

[117] J. Kwon and S. Hailes. “Scheduling UAVs to bridge communications in delay-tolerant networks using real-time scheduling analysis techniques”. In: *2014 IEEE/SICE International Symposium on System Integration*. 2014, pp. 363–369.

[118] Jinman Jung, Yookun Cho, and Jiman Hong. “Impact of Mobility on Routing Energy Consumption in Mobile Sensor Networks”. In: *International Journal of Distributed Sensor Networks* 8.3 (2012), p. 430439.

[119] Jean-Daniel Biomo et al. "Unmanned Aerial ad Hoc Networks: Simulation-Based Evaluation of Entity Mobility Models' Impact on Routing Performance". In: *Aerospace* 2.3 (2015), pp. 392–422.

[120] S. Krug et al. "Movement patterns for mobile networks in disaster scenarios". In: *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014*. 2014, pp. 1–6.

[121] David B Johnson and David A Maltz. "Dynamic Source Routing in Ad Hoc Wireless Networks". In: *Mobile Computing* 353 (1996), pp. 153–181.

[122] E.M. Royer, P.M. Melliar-Smith, and L.E. Moser. "An analysis of the optimum node density for ad hoc mobile networks". In: *ICC 2001. IEEE International Conference on Communications. Conference Record (Cat. No.01CH37240)* 3 (2001), pp. 857–861.

[123] Fan Bai, Narayanan Sadagopan, and Ahmed Helmy. "The IMPORTANT framework for analyzing the impact of mobility on performance of Routing protocols for Adhoc Networks". In: *Ad Hoc Networks* 1.4 (2003), pp. 383–403.

[124] Z J Haas. "A New Routing Protocol for the Reconfigurable Wireless Network". In: *Proceedings of the 1997 IEEE 6th International Conference on Universal Personal Communications, ICUPC'97* 2 (1997), pp. 562–566.

[125] Tracy Camp, Jeff Boleng, and Vanessa Ann Davies. "A Survey of Mobility Models for Ad Hoc Network Research". In: *Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications* 2.5 (2002), pp. 483–502.

[126] Viswanath Tolety. "Load Reduction in Ad Hoc Networks Using Mobile Servers". In: *Master Thesis, Colorado School of Mines* (1999).

[127] Dan Broyles, Abdul Jabbar, and Jpg Sterbenz. "Design and analysis of a 3–D gauss-markov mobility model for highly-dynamic airborne networks". In: *Proceedings of the International Telemetering Conference (ITC)* 46 (2010), pp. 388–404.

[128] Yan Wan et al. "A smooth-turn mobility model for airborne networks". In: *IEEE Transactions on Vehicular Technology* 62.7 (2013), pp. 3359–3370.

[129] Wei Wang et al. "A novel mobility model based on semi-random circular movement in mobile ad hoc networks". In: *Information Sciences* 180.3 (2010), pp. 399–413.

[130] Ouns Bouachir et al. "A mobility model for UAV ad hoc network". In: *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on* (2014), pp. 383–388.

[131] "Paparazzi UAV". In: *http://paparazzi.enac.fr/* (2014).

[132] Somsong Thounhom and Tanapat Anusas-amornkul. "The Study of Routing Protocols for UAVs using Paparazzi Mobility Model with Different Altitudes". In: *Proceedings of the 2016 International Conference on Communication and Information Systems - ICCIS '16*. New York, New York, USA: ACM Press, 2016, pp. 106–111.

[133] Xiaoyan Hong et al. "A group mobility model for ad hoc wireless networks". In: *Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems - MSWiM '99* (1999), pp. 53–60.

[134] Miguel Sánchez and Pietro Manzoni. "ANEJOS: A Java based simulator for ad hoc networks". In: *Future Generation Computer Systems* 17.5 (2001), pp. 573–583.

[135] Mohamed Ayoub Messous, Sidi Mohammed Senouci, and Hichem Sedjelmaci. "Network connectivity and area coverage for UAV fleet mobility model with energy constraint". In: *IEEE Wireless Communications and Networking Conference, WCNC*. Vol. 2016-September. IEEE, 2016, pp. 1–6.

[136] J. Sanchez-Garcia et al. "A Self Organising Aerial Ad Hoc Network Mobility Model for Disaster Scenarios". In: *Proceedings - 2015 International Conference on Developments in eSystems Engineering, DeSE 2015* (2016), pp. 35–40.

[137] Oscar Alvear et al. "A Discretized Approach to Air Pollution Monitoring Using UAV-based Sensing". In: *Mobile Networks and Applications* (2018).

[138] Kun Yao et al. "An adaptive grid model based on mobility constraints for UAV path planning". In: *Proceedings of 2016 2nd International Conference on Control Science and Systems Engineering, ICCSSE 2016*. IEEE, 2016, pp. 207–211.

[139] "SITL Simulator". In: *http://ardupilot.org/dev/docs/sitl-simulator-software-in-the-loop.html* (2016).

[140] Dalimir Orfanus and Edison de Freitas. "Comparison of UAV-based reconnaissance systems performance using realistic mobility models". In: *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2014 6th International Congress on* (2014), pp. 248–253.

[141] Sobin CC et al. "A Survey of Routing and Data Dissemination in Delay Tolerant Networks". In: *J. Netw. Comput. Appl.* 67.C (May 2016), pp. 128–146.

[142] Maria Fazio et al. "Automatic IP Address Configuration in VANETs". In: *Proceedings of the 3rd International Workshop on Vehicular Ad Hoc Networks*. VANET '06. Los Angeles, CA, USA: ACM, 2006, pp. 100–101.

[143] A. Hamidian et al. “Deployment and Evaluation of a Wireless Mesh Network”. In: *2009 Second International Conference on Advances in Mesh Networks*. 2009, pp. 66–72.

[144] *Network Simulator 2*. URL: http://www.isi.edu/nsnam/ns/.

[145] *Position-based protocol implementation on NS-2*. URL: http://www.math.unipd.it/~cpalazzi/FANET-routing.html.

[146] N. Mahmood, J. DeDourek, and P. Pochec. “M2ANET Simulation in 3D in NS2”. In: *In Proc. of SIMUL*. Nice, France, 2014, pp. 24–28.

[147] F. Fabra et al. “On the impact of inter-UAV communications interference in the 2.4 GHz band”. In: *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*. 2017, pp. 945–950.

[148] F. Fabra et al. “A methodology for measuring UAV-to-UAV communications performance”. In: *2017 14th IEEE Annual Consumer Communications Networking Conference (CCNC)*. 2017, pp. 280–286.

[149] Seilendria Hadiwardoyo et al. “Experimental Characterization of UAV-to-Car Communications”. In: *Computer Networks* 136 (Mar. 2018).

[150] *The Opportunistic Network Environment simulator (The One)*. URL: https://akeranen.github.io/the-one/.

[151] Victoria Chang, Pramod Chundury, and Marshini Chetty. “Spiders in the Sky: User Perceptions of Drones, Privacy, and Security”. In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI ’17. Denver, Colorado, USA: ACM, 2017, pp. 6765–6776.

[152] Ivan Stojmenovic. “Simulations in wireless sensor and ad hoc networks: Matching and advancing models, metrics, and solutions”. In: *IEEE Communications Magazine* 46.12 (2008), pp. 102–107.

[153] L. Donatiello and M. Furini. “Ad hoc networks: a protocol for supporting QoS applications”. In: *Proceedings International Parallel and Distributed Processing Symposium*. 2003, 8 pp.–.

[154] Emerson A. Marconato et al. “IEEE 802.11n vs. IEEE 802.15.4: A Study on Communication QoS to Provide Safe FANETs”. In: *Proceedings - 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN-W 2016*. IEEE, 2016, pp. 184–191.

[155] Carla De Francesco, Luigi De Giovanni, and Claudio Enrico Palazzi. “The Interference-aware Drone Ad-hoc Relay Network Configuration problem”. In: *Electronic Notes in Discrete Mathematics* 69 (2018), pp. 317 –324.

[156] Anders Lindgren, Avri Doria, and Olov Schelén. “Probabilistic Routing in Intermittently Connected Networks”. In: *SIGMOBILE Mob. Comput. Commun. Rev.* 7.3 (July 2003), pp. 19–20.