People's Democratic Republic of Algeria
Larbi Ben M'hidi University, Oum El Bouaghi
Faculty of Exact Sciences, Natural and Life Sciences
Department of Mathematics and Computer Science

FINAL STUDY THESIS

In order to obtain the MASTER's degree in Computer Science

Specialization: Distributed Architectures

THEME

SIMULATION OF DATA TRANSMISSION IN A FANET NETWORK

Presented by **ANOUAR BOUCETTA** on 23/06/2024 to the jury composed of:

President	Dr. DEHIMI NOUR ELHOUDA	Oeb university
Supervisor	Dr. ZAIDI SOFIANE	Oeb university
Examiner	Dr. NASRI AHLEM	Oeb university

Academic Year 2023/2024

In the Name of Allah, the Most Beneficent, the Most Merciful

Acknowledgments

Thanks to Allah, to whom all praises go, this work has been accomplished.

In particular, I express my gratitude to my supervisor Dr. ZAIDI SOFIANE, for his valuable advice and assistance throughout the entire duration of this work. A big thank you to all the jury members Dr.Dhimi Nour Elhouda & Dr.Nasri Ahlem for the honor they have bestowed upon me by agreeing to evaluate my work.

I am also thankful to my family, friends, and colleagues for their continuous encouragement and support.

Dedications

To my parents NOUAR and ZOHRA, Your unwavering love and support have been my greatest source of strength.

To my siblings AMINA, HOUSSAM ans MAHDI, Your support and familial love have propelled me forward.

To my future wife -DJIHANE-, Your support is my greatest achievements and the foundation of all my future dreams.

To my friends, Your unwavering support has made the journey vibrant and fulfilling.

Abstract

Abstract:

Communication is one of the most significant design issues for multi-UAV (Unmanned Aerial Vehicle) systems, as it is essential for UAVs' cooperation and collaboration. The infrastructure can facilitate communication between UAVs if each one is directly linked to it, such as through a satellite or a ground station. However, the capabilities of multi-UAV systems are limited by this infrastructure-based communication design. The issues arising from entirely infrastructure-based UAV networks can be resolved by ad hoc networking between UAVs.

FANETs have emerged as a viable option for a variety of unmanned aerial system applications. For instance, FANETs might be utilized for search and rescue operations or urban monitoring. These networks, however, have unique communication issues along with a wide range of specialized features. Consequently, many research projects use network simulation to examine their performance.

Simulating data transmission via the FANET network is the aim of our work. In particular, this work starts with a review of the basics of wireless and ad-hoc networks. Next, we explore FANETs, including their architecture, characteristics, design considerations, communication, applications, and routing protocols. Furthermore, we describe the simulation approaches, focusing on the NS-2 simulator for data transmission modeling in FANETs, and includes Ubuntu installation (dual boot with Windows 10). Lastly, we go into great length about the suggested data transmission paradigm in FANETs, including its design, implementation, and performance assessment.

Key words: FANET, UAV, communication, simulating, networks.

ملخص:

تُعد الاتصالات إحدى أهم قضايا التصميم الخاصة بأنظمة الطائرات دون طيار المتعددة، حيث إنها ضرورية لتعاون وتضافر الطائرات دون طيار. يمكن للبنية التحتية أن تسهّل الاتصال بين الطائرات دون طيار إذا كانت كل واحدة منها مرتبطة بها مباشرة، من خلال قمر صناعي أو محطة أرضية مثلاً. ومع ذلك، فإن قدرات الأنظمة متعددة الطائرات دون طيار محدودة بسبب تصميم الاتصالات القائم على البنية التحتية. يمكن حل المشكلات الناشئة عن شبكات الطائرات دون طيار القائمة على البنية التحتية بالكامل عن طريق الربط الشبكي المخصص بين الطائرات دون طيار.

Abstract

وقد برزت شبكات FANETs كخيار قابل للتطبيق لمجموعة متنوعة من تطبيقات الأنظمة الجوية غير المأهولة. على سبيل المثال، يمكن استخدام شبكات FANET في عمليات البحث والإنقاذ أو المراقبة الحضرية. ومع ذلك، فإن لهذه الشبكات مشاكل اتصال فريدة من نوعها إلى جانب مجموعة واسعة من الميزات المتخصصة. وبالتالي، تستخدم العديد من المشاريع البحثية محاكاة الشبكة لفحص أدائها.

يهدف عملنا إلى محاكاة نقل البيانات عبر شبكة .FANET على وجه الخصوص، يبدأ هذا العمل بمراجعة أساسيات الشبكات اللاسلكية والشبكات المخصصة. بعد ذلك، نستكشف شبكاتFANET ، بما في ذلك بنيتها وخصائصها واعتبارات تصميمها واتصالاتها وتطبيقاتها وبروتوكولات التوجيه. علاوةً على ذلك، نصف أساليب المحاكاة، مع التركيز على محاكي NS-2 لنمذجة نقل البيانات في شبكات الشبكات اللاسلكيةFANET ، ويتضمن تثبيت نظام Ubuntu الإقلاع المزدوج مع نظام التشغيل (Windows 10). وأخيراً، نتناول بإسهاب كبير نموذج نقل البيانات المقترح في شبكات FANET، بما في ذلك تصميمها وتنفيذها وتقييم أدائها.

الكلمات المفتاحية: شبكات FANETs، محاكاة، الطائرات دون طيار، الاتصالات، شبكات.

Résumé :

La communication est l'un des principaux problèmes de conception des systèmes multi-drones (véhicules aériens sans pilote), car elle est essentielle à la coopération et à la collaboration entre les drones. L'infrastructure peut faciliter la communication entre les drones si chacun d'entre eux y est directement relié, par exemple par l'intermédiaire d'un satellite ou d'une station au sol. Toutefois, les capacités des systèmes multi-drones sont limitées par cette conception de la communication basée sur l'infrastructure. Les problèmes posés par les réseaux de drones entièrement basés sur l'infrastructure peuvent être résolus par la mise en réseau ad hoc entre les drones.

Les FANET sont apparus comme une option viable pour toute une série d'applications de systèmes aériens sans pilote. Par exemple, les FANET peuvent être utilisés pour des opérations de recherche et de sauvetage ou pour la surveillance des villes. Toutefois, ces réseaux présentent des problèmes de communication uniques ainsi qu'un large éventail de caractéristiques spécialisées. Par conséquent, de nombreux projets de recherche utilisent la simulation de réseau pour examiner leurs performances.

La simulation de la transmission de données via le réseau FANET est l'objectif de notre travail. En particulier, ce travail commence par un examen des bases des réseaux sans fil et ad-hoc. Ensuite, nous explorons les réseaux FANET, y compris leur architecture, leurs caractéristiques, les considérations de conception, la communication, les applications et les protocoles de routage. En outre, nous décrivons les approches de simulation, en nous

concentrant sur le simulateur NS-2 pour la modélisation de la transmission de données dans les FANET, et nous incluons l'installation d'Ubuntu (double démarrage avec Windows 10). Enfin, nous examinons en détail le paradigme de transmission de données proposé dans les FANET, y compris sa conception, sa mise en œuvre et l'évaluation de ses performances.

Les mots clés : les FANET, la simulation, drones, la communication, réseaux.

Table of contents

Table of contents

Table of contents

Table of contents

Table of contents

List of tables

List of tables:

List of figures

List of figures:

List of abbreviation

List of abbreviations:

- FANET: flying ad-hoc network
- MANET: Mobile Ad hoc Network
- VANET: Vehicular Ad Hoc Network
- UAV: Unmanned Aerial Vehicles
- NS: network simulator
- TCL: Tool Command Language
- WPAN: Wireless Personal Area Networks
- WLAN: Wireless Local Area Networks
- WWAN: Wireless Wide Area Networks
- WMAN: Wireless Metropolitan Area Networks
- MSS: Mobile Support Stations
- BS: Base Stations
- MU: mobile sites or units
- Wi-Fi: Wireless Fidelity
- U2U: UAV-UAV
- U2I: UAV-Infrastructure
- DARPA: Defense Advanced Research Projects Agency
- IEEE: the Institute of Electrical and Electronics Engineers
- IETF: Internet Engineering Task Force's
- PDAs: Personal Digital Assistants
- SPAN: Smart phone ad hoc network
- WMN: Wireless mesh network
- WSNs: Wireless sensor networks
- GBSs: ground-based systems
- LoS coverage: Line of Sight coverage
- QoS: Quality of Service
- MAC: Media Access Control
- TCP: Transmission Control Protocol
- UDP: User Datagram Protocol
- DTN: Delay-tolerant networks

List of abbreviation

- GPS: Global Positioning System
- SAR operations: Search and Rescue operations
- LCAD paradigm: Locate, Communicate, Access, and Deliver paadigm
- VINT: Virtual Inter Network Test bed
- NSF: The National Science Foundation
- IDE: Integrated Development Environment
- MATLAB: MATrix LABoratory
- GRUB: GRand Unified Bootloader
- BIOS: Basic Input/Output System

General introduction

General introduction:

A Flying Ad-Hoc Network (FANET) is a type of wireless communication network specifically designed for Unmanned Aerial Vehicles (UAVs), commonly known as drones. FANETs enable UAVs to communicate with each other and with ground stations without relying on a fixed infrastructure. This network is dynamic and self-organizing, adapting in real-time to the movement and changing positions of the UAVs.

The American military used unmanned aerial vehicles (UAVs) for the first time during the end of World War. The idea was to steer distantly piloted aircraft without a pilot. Thanks to technological advancements, data exchanged across FANETs may now include a variety of formats, including text, photos, videos, and more. Numerous military and civilian applications, such as the search for and rescue of survivors of disasters, are possible with these data. Additionally, they are employed in a number of operations related to environmental and natural phenomenon control, such as border monitoring, managing forest fires, etc.

In order to test and analyze communication protocols and information transmission across a FANET network without material losses or damages, researchers use regression modeling, relying on tools like NS-2 (Network Simulator-2).

In our work, we simulated data transmission across FANET networks. To do this work, we went through several steps, starting with installing UBUNTU, NS-2.35, and PATCH 3D tools. Next, we created the UAVs' mobility and replicated data transmission over the FANET. In order to determine whether there is a loss in the data transmission between the emitter and the receiver, we have finally generated the trace of the data packets sent and received throughout the simulation.

In this note, you'll find four main chapters, each diving into different parts of our research:

Chapter 01: Getting to Know Wireless and Ad-hoc Networks

Here, we start by introducing wireless networks, breaking down what they are, their types, and how they communicate. We then move on to explore ad-hoc networks, digging

into their background, how they're modeled, what makes them unique, and the different types out there.

Chapter 02: Flying Ad-hoc Networks (FANETs)

This section is all about FANETs, we explore FANETs, including their architecture, characteristics, design considerations, communication, applications, and routing protocols.

Chapter 03: Taking a Look at Simulation

Here, we step into the world of network simulation, with a focus on FANET simulators. We describe the simulation approaches, focusing on the NS-2 simulator for data transmission modeling in FANETs, and includes Ubuntu installation (dual boot with Windows 10).

Chapter 04: Design and Implementation of Data Transmission in FANETs

Finally, we get down to business with the design and implementation of data transmission protocols specifically for FANETs. We break it down with diagrams, explain how to make it happen with Tcl scripts, and show you the results of putting it all into action.

I extend my heartfelt gratitude to my supervisor, Dr. Zaidi Sofiane, for his unwavering support, guidance, and expertise throughout this project.

Chapter 01

Generalities on Wireless and Ad-hoc Networks

1. Chapter 01: Generalities on Wireless and Ad-hoc Networks

1.1.Introduction:

Wireless and ad-hoc networks are revolutionizing the way we connect and communicate, playing a crucial role in the development of modern communication systems. These networks, which eliminate the need for wired connections, offer flexibility, cost savings, and a wide range of applications from personal to metropolitan scales. The growth of mobile environments, the evolution of wireless communication technologies, and the emergence of ad-hoc networks have created a dynamic field that addresses both the demands and challenges of modern connectivity. This chapter provides a comprehensive overview of wireless networks, detailing their types, mobile environments, communication technologies, and the history and characteristics of ad-hoc networks. By understanding these fundamentals, we can appreciate the complexity and potential of wireless and ad-hoc networks in enhancing our digital landscape.

1.2.Wireless Networks:

1.2.1. Definition:

Computer networks that are connected wirelessly don't use any form of wire. Businesses can avoid the expensive process of installing cables inside buildings or connecting disparate equipment locations by utilizing a wireless network. Radio waves serve as the foundation of wireless systems, and their implementation occurs at the physical level of network architecture. [1]

1.2.2. Categories of Wireless Networks:

There are four main types of wireless networks:

i. **Wireless Personal Area Networks (WPAN):** are networks with a limited range that link devices in a constrained region. WPANs typically link devices that are within arm's length, while their range can stretch up to thirty feet. A WPAN may link suitable devices close to a central location via Bluetooth technology. For example, you can connect a headphone to a laptop on your desk.

ii. **Wireless Local Area Networks (WLAN):** are wireless networks that, compared to WPANs, employ radio waves rather than Bluetooth technologies. When using the internet, an access point often consists of one or more cables. For example, a wired internet connection that is connected to a router transmits a

wireless signal to other devices. WLANs are used to establish connections to the internet and local resources. Spread-spectrum or OFDM technologies allow the range to be extended across a complete building or campus, or it can be restricted to a single room or residence.

iii. **Wireless Wide Area Networks (WWAN):** may be kept up over vast regions, such cities or nations, using a number of satellite systems, antenna locations, or cell phone signals. WWANs, which have a large coverage area, offer a means of maintaining connectivity when other means of network access are not accessible.

iv. **Wireless Metropolitan Area Networks (WMAN):** link many WLANs in a city or other metropolitan region, such as various buildings. [2]

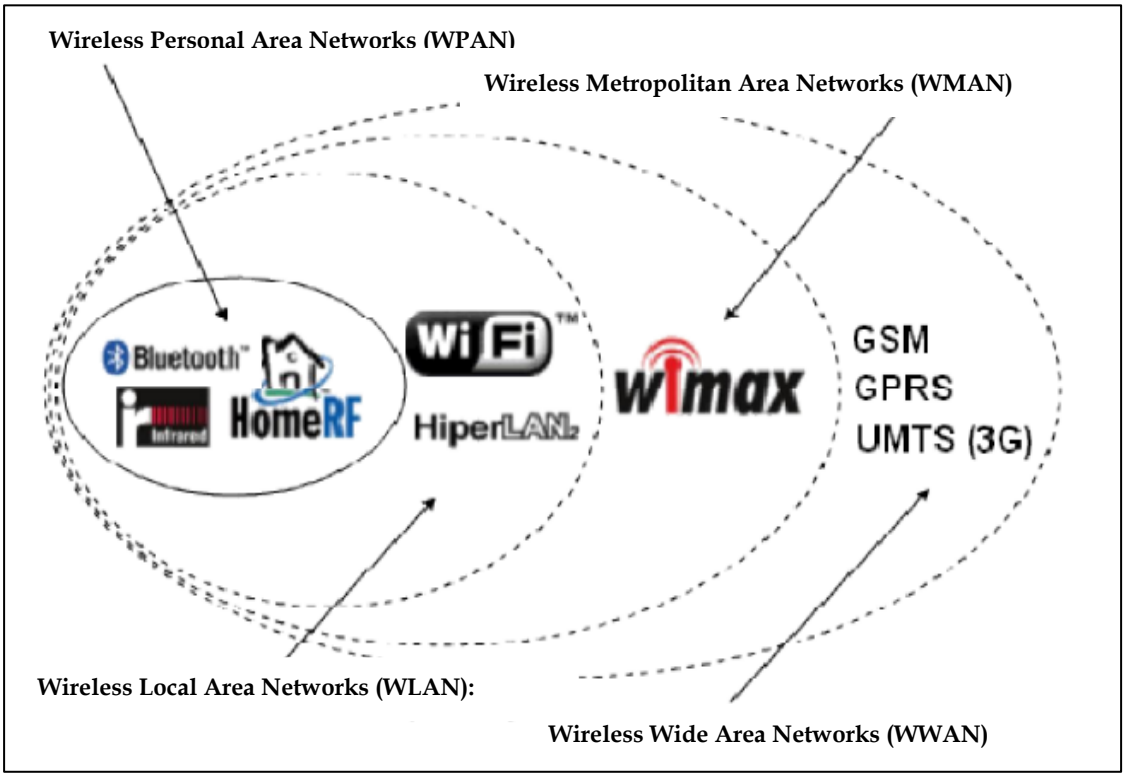

Figure 1-1 Categories of Wireless Networks *[3]*

1.2.3. Mobile environments:

A mobile environment is a system composed of mobile units that allows its users to access information regardless of their geographical positions, The wireless network offers two modes of operation: the infrastructure mode and the ad hoc or infrastructure-less mode. [4]

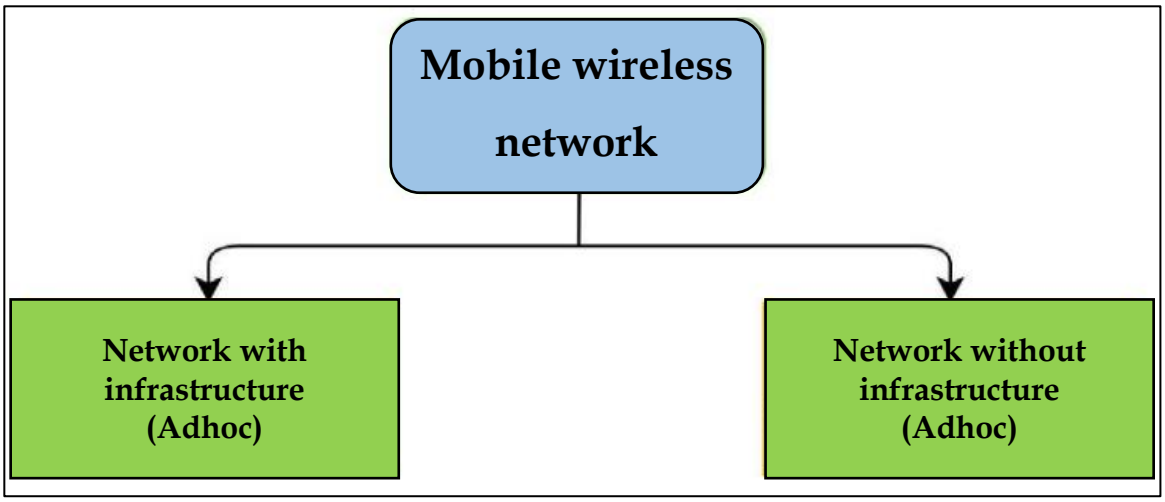

Figure 1-2 Categories of Mobile Networks *[3]*

i. **Mobile Network with Infrastructure:** Certain fixed sites, referred to as Mobile Support Stations (MSS) or Base Stations (BS), are outfitted with wireless communication interfaces for direct communication with mobile sites or units (MU) situated within a constrained geographic area known as a cell. This configuration is known as infrastructure mode, also referred to as Basic Service

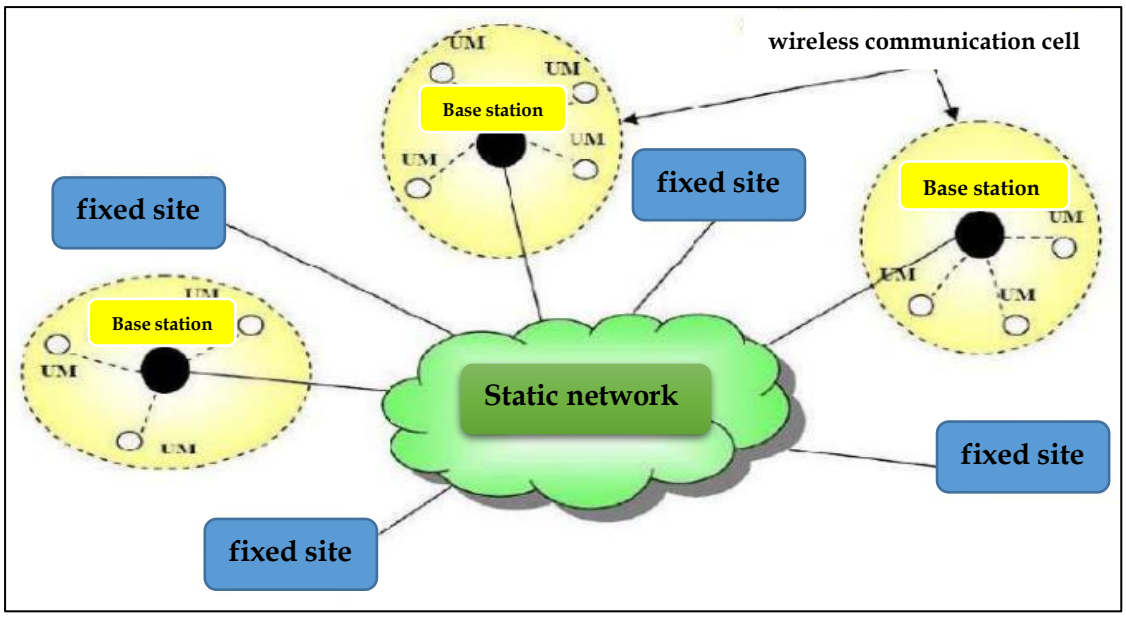

Figure 1-3 Cellular Networks *[3]*

Set (BSS) mode. Every base station represents a cell that mobile units may communicate with. While the wireless links have restricted capacity, which significantly reduces the volume of information transferred, the fixed locations are connected via a wired communication network, which is often dependable and fast. A mobile unit in this model can only have a direct connection to one base station at a time. [4]

ii. **Mobile Network without Infrastructure:** The mobile network without infrastructure, also known as an **Ad hoc network** or Independent Basic Service Set (IBSS), does not include the "fixed site" entity. All sites in the network are mobile and communicate directly using their wireless communication interfaces. The absence of infrastructure or a wired network composed of base stations forces the mobile units to act as routers, participating in the discovery and maintenance of paths for other hosts in the network. [4]

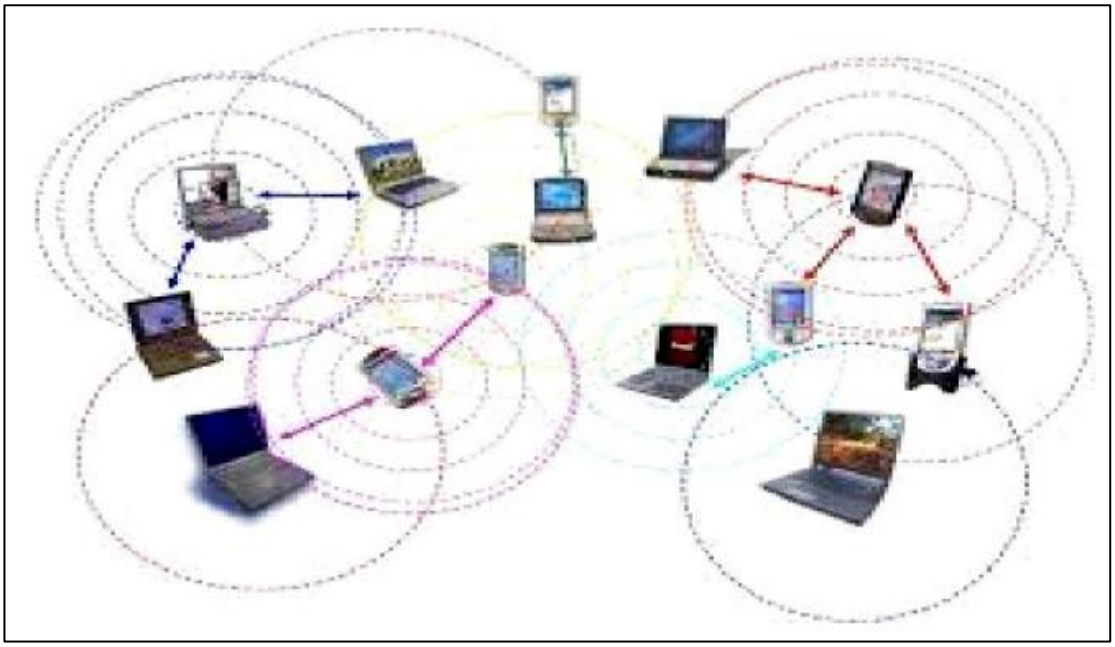

Figure 1-4 AD-HOC network [3]

1.2.4. Wireless Communication Technologies:

Many wireless communication technologies are potential choices for providing reliable and flexible communication links for the rapid deployment of FANET. A list of probable candidates that can be selected for the various communication links, UAV-UAV (U2U) and UAV-Infrastructure (U2I), is tabulated. Wireless technologies can be divided into two main categories: short-range and long-range communication technologies. Short-range

communication technologies, such as Wi-Fi, ZigBee, and Bluetooth, are used for short-distance communications, while long-range communication technologies, such as cellular, WiMAX, and satellite, can be used for more extensive areas. [3, p. 6]

1.3.Ad hoc networks:

1.3.1. History:

The earliest radio networks were created in the 1970s as part of the US military's DARPA (Defense Advanced Research Projects Agency) initiative. These networks shared the broadcast channel by repeating packets in order to increase the total coverage area. They already had a distributed design. DARPA then went on to construct the Survivable Radio Networks in 1983. Overcoming obstacles was the main objective, especially by permitting the growth of networks with a lot of nodes and controlling energy and security. Still, these networks were only ever studied from a military perspective. The development of wireless networks surrounding stationary bases was made possible by the IEEE 802.11 standard, and it wasn't until the late 1990s that civilian research started to focus on the problems associated with these networks. [4]

1.3.2. Definition:

The ad hoc networks that interest us are those that were analyzed and documented in 2007 by the Internet Engineering Task Force's (IETF) MANET (Mobile Ad hoc Network) working group in France. RFC 2501 offers a formal specification of these networks. Ad hoc networks are made up of mobile platforms, or nodes, that may move freely and link several hosts and wireless devices (such as a router). Thus, an independent system of movable nodes is called an ad hoc network. This system can use gateways to communicate with fixed networks or run independently. [4]

1.3.3. Modeling an Ad Hoc Network:

An ad hoc network can be modeled by a Gt=(Vt,Et), where Vt represents the set of nodes (e.g., mobile units or hosts) in the network and Et models the set of connections that exist between these nodes. If e=(u,v)∈Et, it means that nodes u and v are able to communicate directly at time t.

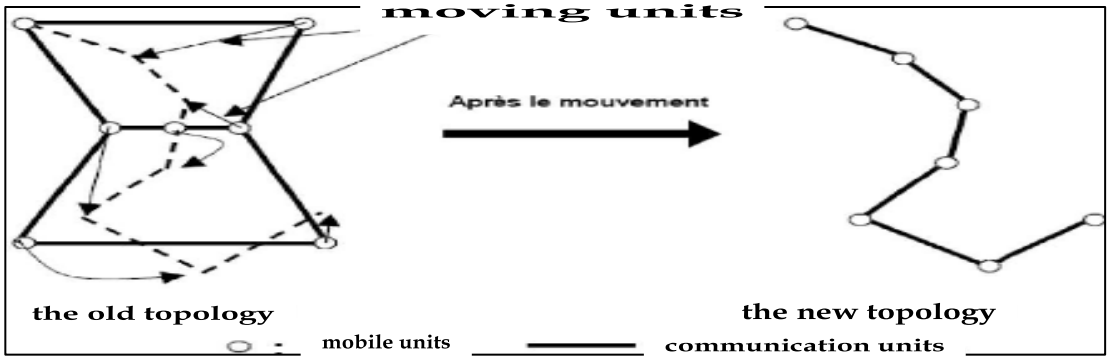

Figure 1-5 Change in Ad Hoc Network Topology [4]

The network topology can change at any moment. In the following example (FIG 1.5), node A sends a data stream to node E, with the data being routed through nodes G and F. After movement, we notice that the routing of data becomes more complex since the stream no longer follows the path through G and F, but must traverse all the nodes B, C, D, G, and F. Thus, the network is dynamic and unpredictable, making disconnections of units very frequent. [4]

1.3.4. Characteristics of Ad Hoc Networks:

Mobile Ad Hoc Networks (MANETs) are characterized by the following:

i. **Absence of Centralized Infrastructure:** Ad hoc networks are distinct from other mobile networks in that they don't have any kind of centralized management or pre-existing infrastructure. Establishing and sustaining network connectivity is the ongoing responsibility of the mobile hosts.

ii. **Dynamic Topology:** The network's mobile units travel at will and without restriction. As a result, the topology of the network might alter quickly and without warning at any time. There are two types of linkages in a topology: unidirectional and bidirectional.

iii. **Energy Constraint:** Mobile devices often have limited battery life, and in certain instances—like PDAs (Personal Digital Assistants)—the battery life is severely

constrained, which results in slower processing times. The routing feature has already used some of the energy. This restricts the range of apps and services that any node can support.

iv. **Limited Bandwidth:** Using a common communication channel is a basic feature of networks built on wireless communication. Each server can use a limited amount of bandwidth as a result of this sharing.

v. **Node Heterogeneity:** A mobile node may be equipped with one or more radio interfaces with varying transmission capabilities and operating on different frequency ranges. This heterogeneity in capabilities can result in asymmetric links within the network. Additionally, nodes may differ in terms of processing capacity (CPU, memory), software, and mobility (slow, fast). In such cases, dynamic adaptation of protocols is necessary to support these diverse situations.

vi. **Security and Vulnerability:** The fundamental problem with ad hoc networks is not so much the physical media as it is the reality that every node is equal and can be required for the network to function. There is a greater chance of unwanted entry into the network, which makes it harder to detect intrusions or denial-of-service assaults. The reporting of intrusion detection information is made more difficult by the absence of centralization.

vii. **Multihop:** An ad hoc network is characterized as "multihop" because multiple mobile nodes can participate in routing and act as intermediate routers. [4]

1.3.5. Types of ad hoc network:

To access the internet, users do not require a router or wireless base station. Users can utilize an ad hoc network without a router or wireless base station. An ad hoc network is a local area network (LAN). It is not in use for very long. Ad hoc networks can be permanently configured, in which case they transform into wireless networks. To characterize a solution that is specific to a single object, an ad-hoc network is created when objects and people who wish to interact come together and establish communication. Just for a little period of time. The wireless card connects the two of them. There will be a breakdown in the connection between the two devices. Ad hoc networks allow several devices to join at once, yet the network might not function. Another term for a network that is assembled on the fly is ad hoc mode. As Fig. 1.6 illustrates, ad hoc networks come in several varieties. [5]

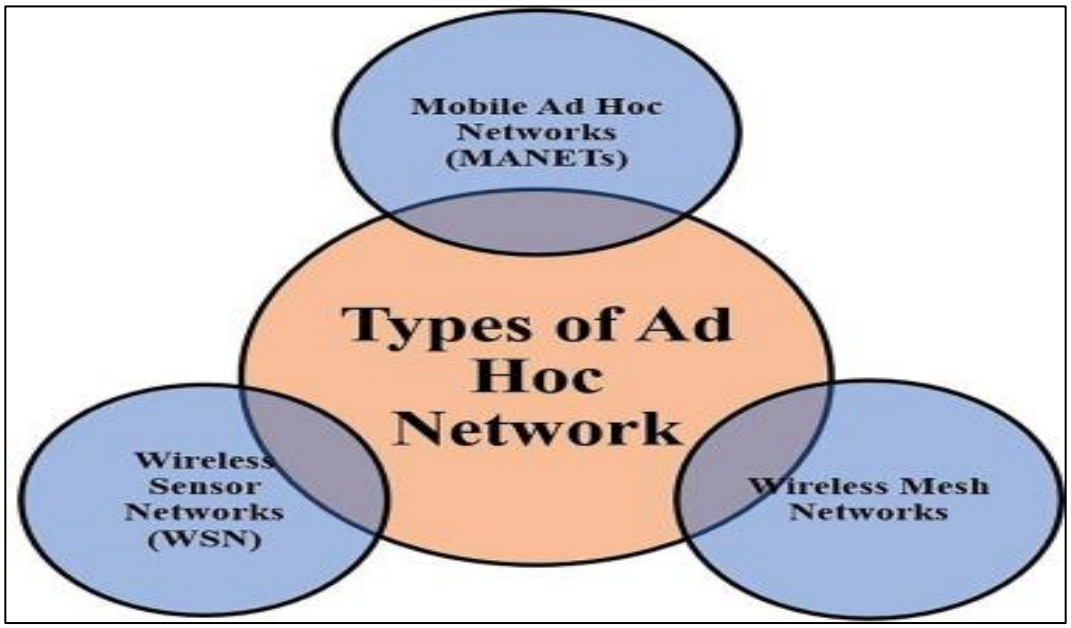

Figure 1-6 Types of Ad Hoc Network [5]

i. **Mobile ad hoc networks (MANETs):** It is constructed on top of both wireless ad hoc networks and link-layer networks. It also permits the creation of a mobile network. The nodes link wireless data as they travel. A network is this collection of nodes. It requires no setup and is self-configuring and self-healing. Because the network's topology is dynamic, MANET nodes are free to relocate. These networks are instantaneously assembleable and link and interact via wireless and mobile devices (such as iPads, computers, and cellphones). MANET characteristics are seen in Fig. 1.7. [5]

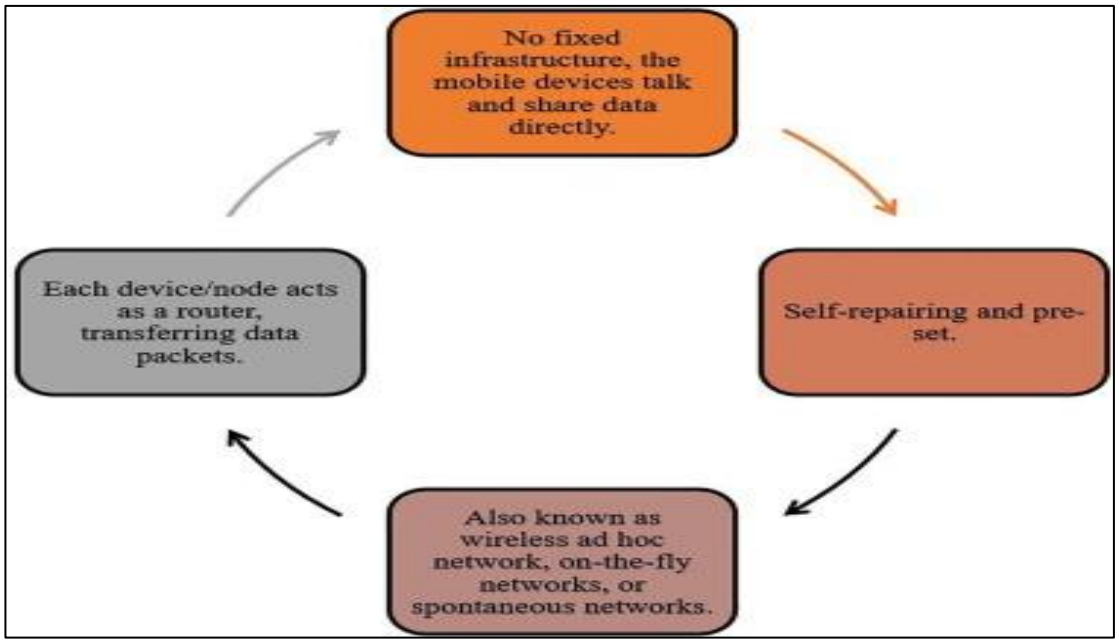

Figure 1-7 MANETs Features [5]

ii. **Smart phone ad hoc networks (SPANs):** Ad hoc groups are groups of people who meet sometimes to utilize phones. If a few phones are near one other and have ad hoc network technology, they can create an ad hoc network. Peer-to-peer networks may be created by smartphone users without the usage of conventional network infrastructure. Wireless access points, cellular carriers, and conventional network infrastructure are not required. Bluetooth and Wi-Fi are already integrated into smartphones that are available on the market. Wi-Fi SPANs work on the same principle as Wi-Fi ad-hoc mode, which enables instantaneous phone communication via a transparent neighbor and route-finding technique known as fine neighbor. Ad hoc routing, or multi-hop routing, and relays set SPANs apart from conventional hub-and-spoke

networks, such as Wi-Fi direct. Peers are free to join and quit the group because there is no group leader, making it impossible to determine who is in command of the group. Additionally, SPANs lack a group leader. It's moving swiftly, which facilitates communication and interpersonal connections. These intelligent gadgets require a special infrastructure in order to link the wireless devices. [5]

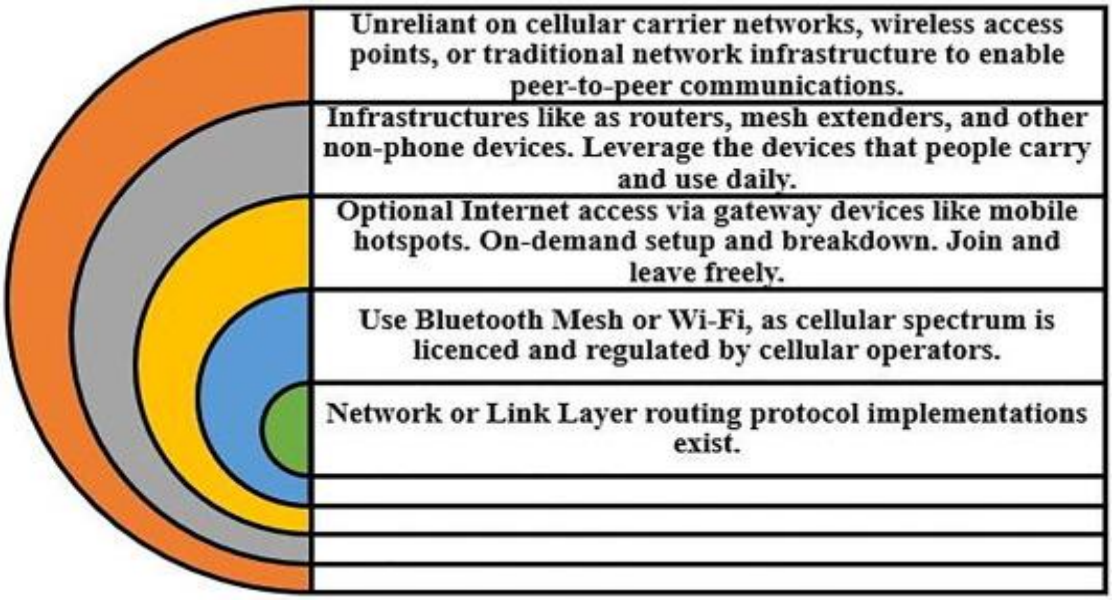

Figure 1-8 Features of SPANs [5]

iii. **Wireless mesh network (WMN):** Numerous wireless mesh routers make up multi-hop wireless networks. A multi-hop wireless network is a wireless mesh network. This backbone, which resembles a mesh, connects all of these routers wirelessly. Users can access the network using a laptop or smartphone with wireless connectivity by connecting to a router, which also serves as a wireless access point. The backbone mesh network is used by people to send and receive data to and from one another. Wi-Fi networks are linked to one or more routers that are connected to other networks, such as the internet. As seen in Fig. 1.9, wireless mesh networks, also known as wireless ad hoc networks (WANET), are used for the following purposes. [5]

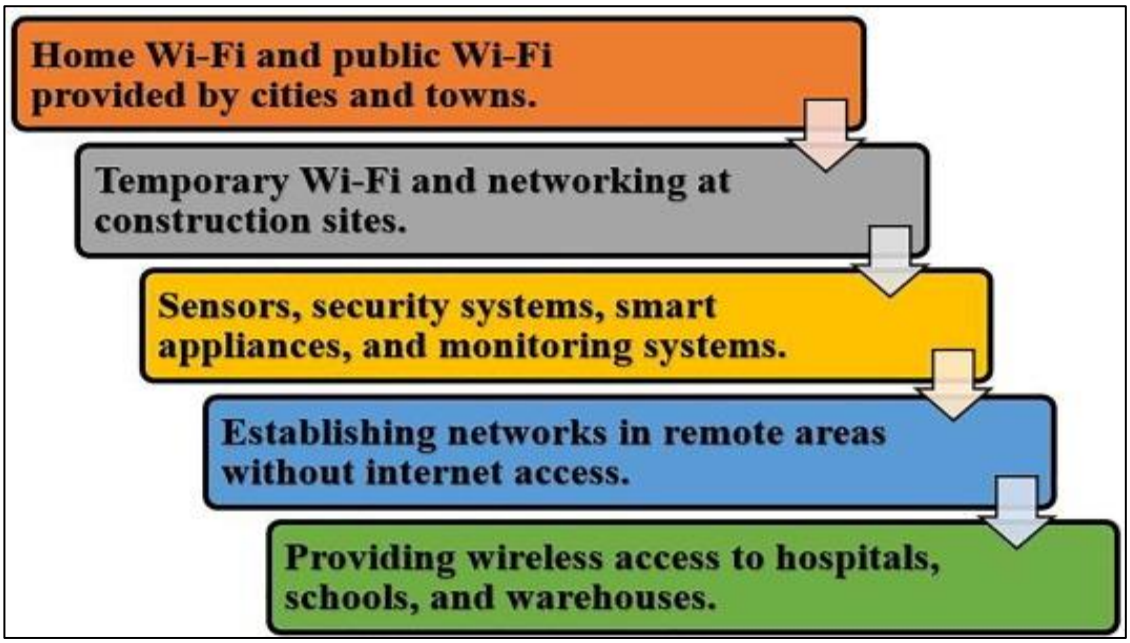

Figure 1-9 WMN are frequently used for the following purpose [5]

iv. **Wireless sensor networks (WSNs):** A network known as a wireless sensor network (WSN) operates without the requirement for any form of infrastructure. Users may observe more wireless sensors, the environment around them, and how the system operates. In certain locations, people utilize WSNs to control and keep an eye on the surroundings. To exchange data with others, the base station of the WSN system is online. Wireless Sensor Nodes (WSNs) are nodes that exchange data wirelessly with one another. These folks make use of this data to learn more about themselves. Nodes are frequently dispersed haphazardly and have little power. There are several restrictions on how WSN security may be configured. They are difficult to set up for security because of their low memory, processing power, battery life, and bandwidth. Someone who wishes to violate someone else's control, privacy, or availability may attack that person. It also offers security tips for the keys. They talk about a novel approach to WSN routing that takes energy consumption into account. The Internet of Things (IoT) relies on several important technologies, and WSN is discussed as a means of enabling the IoT. IoT security requires a large number of devices and an open infrastructure. In Fig. 1.10, components visible to users of wireless sensor networks are shown together. [5]

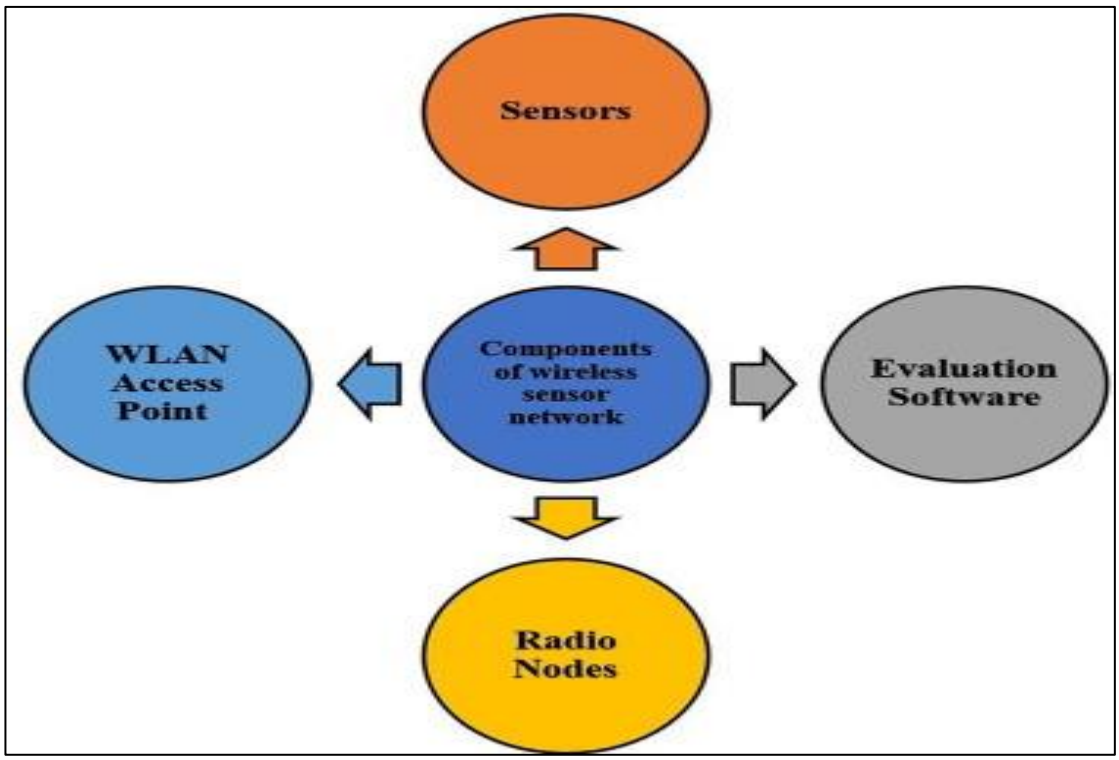

Figure 1-10 Components of wireless sensor network [5]

1.4.Conclusion:

The exploration of wireless and ad-hoc networks reveals a rapidly evolving domain with significant implications for communication technologies. Wireless networks, with their various categories like WPANs, WLANs, WWANs, and WMANs, demonstrate the versatility and adaptability of wireless communication in different settings. The study of mobile environments highlights the operational modes and infrastructure requirements, emphasizing the importance of infrastructure and ad-hoc modes. Ad-hoc networks, with their unique history, dynamic topology, and diverse applications, underscore the potential for decentralized, flexible, and robust communication systems. Understanding the characteristics, types, and challenges of these networks is essential for advancing technology and addressing the future needs of connectivity and communication in an increasingly mobile and interconnected world.

Chapter 02

Chapter 02: Flying ad-hoc networks (FANETs)

2. Chapter 02: Flying ad-hoc networks (FANETs)

2.1. Introduction:

Flying ad-hoc networks (FANETs) represent a remarkable fusion of cutting-edge technology and innovative applications. As an integral subset of vehicular ad-hoc networks (VANETs), FANETs leverage the autonomy and mobility of unmanned aerial vehicles (UAVs) to revolutionize various domains, ranging from military reconnaissance to civilian emergency response. This chapter delves into the intricacies of FANETs, exploring their network definition, architecture, characteristics, design considerations, communication protocols, applications, and simulation methodologies.

FANETs embody the convergence of advanced communication paradigms and unmanned aerial systems, offering unparalleled opportunities for dynamic and agile network deployments. By interconnecting UAVs in an ad-hoc manner, FANETs enable swift mission completion, cost-effective operations, and enhanced situational awareness. However, realizing the full potential of FANETs necessitates a deep understanding of their unique attributes and challenges, spanning from mobility constraints to energy autonomy.

In this chapter, we embark on a comprehensive journey through the realm of FANETs, unraveling their architecture intricacies, design nuances, communication modalities, and real-world applications. Through meticulous examination and analysis, we aim to illuminate the path towards harnessing the transformative power of FANETs in diverse operational scenarios.

2.2.FANETs network definition:

Ad hoc networks of drones, or FANETs (Flying ad hoc network), are a subfamily of VANETs. They emerged in response to the need to interconnect multiple drones or UAVs (Unmanned Aerial Vehicles) within a multi-UAV system. A drone is defined as a pilotless aircraft capable of autonomous navigation, either using onboard systems or through remote control. Typically equipped with sensors, drones determine their position and collect information about an area of interest. They offer new applications in both military and civilian domains, such as interconnecting terrestrial ad hoc networks, conducting search and rescue operations after natural disasters, detecting and tracking military targets, monitoring forest fires, agricultural remote sensing, etc. [6]

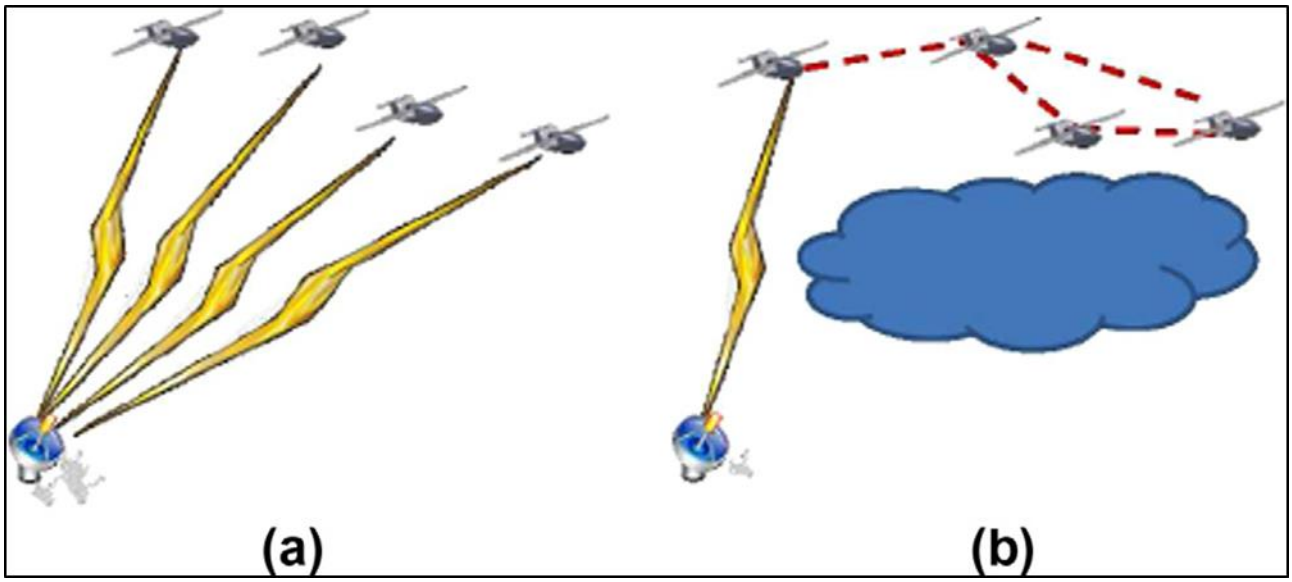

Figure 2-1 A FANET application scenario for reliable multi-UAV communication network [6]

In a multi-UAV system (FIG 2.1), ground or satellite base stations facilitate infrastructure-based UAV communications.

However, as the number of UAVs within the system increases, direct communication between UAVs emerges as an appealing alternative to centralized communication routed through base stations. This can be achieved by establishing an ad hoc network between UAVs. Indeed, leveraging multiple drones in ad hoc mode offers several benefits:

- Reduction in mission completion time: Reconnaissance, surveillance, and rescue missions can be expedited based on the number of drones deployed.
- Reduction in overall maintenance costs: Rather than relying on a single costly large UAV, employing several mini-drones with minimal individual maintenance expenses is a more cost-effective approach. [6]

2.3.FANET architecture:

Similar in standard to MANETs, FANETs' nodes are fusing together to create unique characteristics. Two types of communication that can be formed between the nodes that make up a traditional FANET are depicted in Figure 2.2: [7]

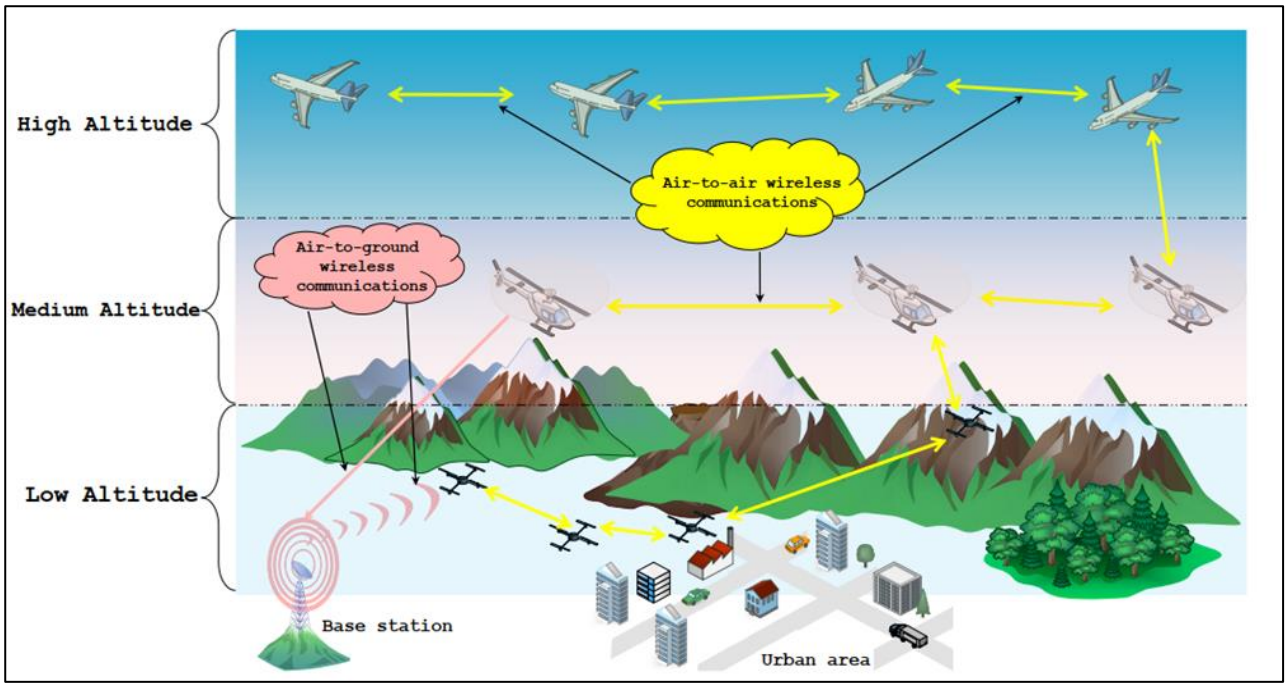

Figure 2-2 Wireless communications between FANET nodes [7]

2.3.1. Air-to-air wireless communications:

To get beyond the limitations on transmission ranges imposed by communication between UAVs and terrestrial base stations, UAVs can speak with one another using a pure ad hoc architecture. Furthermore, multi-hop communications and other applications can be supported by this type of wireless communication when a node wishes to establish a transmission of a data packet to another node that is outside of its range.

2.3.2. Air-to-ground wireless communications:

Not every UAV in FANET is able to connect with the systems that are currently in place, such satellites and ground stations. However, in order to enhance and expand connection as well as offer new services, only specific UAVs are able to communicate with infrastructures.

2.3.3. Hybrid communication:

Hybrid communication is a mix of UAV-to-UAV and UAV-to-GCS communications (see Fig. 5.c). As a result, the UAV can communicate data directly to the GCS in a single hop or via many UAVs in the mission area. [8]

FANET can be viewed as a specific instance of MANET, with a few notable distinctions, such as the high degree of mobility and large distances that separate FANET nodes. To get over these limitations, more communication channels and novel approaches

might be needed. It is thought that wireless communication between FANET nodes is a difficult operation that requires communication rules in the form of routing protocols to support the efficacy of such transmissions. [7]

2.4.FANET characteristics:

FANET stands out as a particular kind of MANET due to its distinct characteristics that define the nodes and the surrounding area. FANET can be characterized by a number of primary attributes, including UAV density, propagation model, topology, scalability, and localization. [7]

This type of network also distinguishes other features, which are as follows and are explained and discussed:

2.4.1. Unmanned Aerial Vehicles (UAVs):

FANET nodes typically fly in the skies and move more faster than MANET nodes. These nodes are miniature mobile drones (UAVs) that can operate automatically without human intervention. UAVs may be controlled using algorithms, which is one of their key features. Their top speeds can surpass 400 kilometers per hour. FANET's node density is poor due to long distances between nodes, requiring longer transmission ranges (up to kilometers). FANET topologies are dynamic, leading to frequent connection failures between nodes.

2.4.2. Network connectivity:

FANET's low density and great mobility make it a sparsely linked network. This causes fluctuations in link quality, potentially leading to loss of connectivity and performance deterioration. A potential option is to establish an ad hoc network between nodes to improve communication coverage. Nodes can interact with each other even if they cannot connect to the ground infrastructure.

2.4.3. Energy autonomy:

The power and motions of FANET nodes are fueled and provided by the nodes' own energy resources. Indeed, there is no energy constraint because each node is fitted with rechargeable batteries that are constantly refilled as the UAVs move. Batteries can also be powered by nodes' resources such as solar energy, gasoline, electricity, and so on. Additionally, moving a UAV requires far more energy than calculating data. FANET nodes

have no energy power limits, unlike MANET nodes, which need developers to consider energy usage by communication protocols to lengthen network lifetime. Mini UAVs suffer from restricted cargo capacity due to energy consumption constraints. Mini UAVs differ from typical UAVs in terms of speed, altitude, and weight, which impact energy autonomy.

2.4.4. Quality of Service (QoS) requirements:

Certain FANET applications require effective real-time services, such as an application specialized to transmitting aerial imagery and video for real-time monitoring. To maintain the stability of real-time applications, several metrics must be considered, including latency, available bandwidth, packet losses, and jitter. For video transmission, QoS primarily aims to decrease latency between UAVs and receivers. Sufficient bandwidth is required for efficient video transmission to the desired destination. These QoS requirements are critical, particularly when significant judgments must be made by the organization receiving the video. As a result, real-time applications must meet a minimum of QoS criteria while accounting for the extremely dynamic nature of FANETs.

2.4.5. Mobility models:

The motions of nodes in FANET are often predetermined. External circumstances, such as weather or missions, might cause node motions to change, affecting the mobility model directly. Mobility models designed for MANET may not be appropriate for FANET, resulting in insufficient path planning. FANET distinguishes between several types of UAVs, requiring unique attributes for each. Overall, FANET distinguishes three types of UAVs: large UAVs, small UAVs, and tiny UAVs. Table 1 provides a quick comparison of three types of UAVs: [7]

Table 1 Comparative study between each kind of UAVs [7]

Properties	mini UAVs	small UAVs	large UAVs
Transmission range	*Small*	*Medium*	*Large*
Altitude	≈300m	≈2000m	>3000m
Energy autonomy	*Restricted*	*Not restricted*	*Not restricted*
Speed	*Medium*	*High*	*Very high*

A FANET protocol is considered satisfactory if it meets all of the above-mentioned performance criteria [9] [10]. FANETs require strong protocols that are tested and proven in real-world circumstances. However, the protocols established for VANETs and MANETs cannot be immediately implemented to meet the specific needs of FANETs [9] [10] [11]. The primary distinctions between MANETs, VANETs, and FANETs are shown below (Table 2). able 2: [14]

Table 2 Difference between FANETs, VANETs, MANETs [7]

Characteristics	MANETs	VANETs	FANETs
Density	*High*	*High*	*Low*
Network connectivity	*High*	*Medium*	*Low*
Energy autonomy	*Low*	*High*	*High (Depends on UAV kind)*
Topology variation	*Occasionally*	*Frequently*	*Very frequently*
Scalability	*Medium*	*High*	*Low*
QoS	*Low*	*High (Depends on the application)*	*High (Depends on the application)*
Mobility models	*Random*	*Restricted throug roads pattern*	*Predefined by mobility models*
Node speed	*Medium*	*High*	*Very high*

2.5.FANET design considerations:

The differentiating properties of FANET necessitate special design considerations. This part covers key FANET design concerns, including flexibility, scalability, latency, UAV platform limits, and bandwidth requirements.

2.5.1. Adaptability:

Various parameters in FANET can change during multi-UAV operations due to their mobile nature and shifting operational requirements. UAV failures and environmental factors like weather can impact system performance. Mission updates may necessitate flight plan adjustments. Designing FANET to adapt to these changes is crucial, requiring flexible physical and network layers, effective routing protocols, and adaptable transport layers.

2.5.2. Scalability:

The collaborative effort of UAVs can significantly enhance system performance compared to single-UAV systems, which is the primary motivation for using multi-UAV

systems. Performance improvements often correlate with the number of UAVs deployed. For instance, a greater number of UAVs can expedite search and rescue operations. Therefore, FANET protocols and algorithms should ensure seamless operation of any number of UAVs with minimal performance degradation.

2.5.3. Latency:

FANET, like other networks, prioritizes latency as a key design consideration. FANET latency requirements vary per application. Real-time FANET applications, like military monitoring, require data packets to be transmitted with a certain latency. Low latency is necessary for avoiding collisions among several UAVs.

2.5.4. UAV platform constraints:

The deployment of FANET communication gear on the UAV platform comes with limits. The weight of the gear is a crucial factor in the performance of UAVs. Lighter hardware implies lighter payload, which extends endurance. The reduced communication hardware provides a chance to mount extra sensors on the UAV. Assuming a steady overall payload and lighter communication hardware, sophisticated sensors and peripherals can be installed. FANET designs for UAV platforms have space constraints as well. Mini UAVs face significant space constraints when it comes to communication hardware.

2.5.5. Bandwidth requirement:

In FANET applications, the main aim is to gather environmental data and transmit it to a ground base quickly. For example, in surveillance or rescue operations, real-time images or videos must be relayed swiftly with high bandwidth. Technological advancements allow for high-resolution data collection, increasing bandwidth needs. However, constraints such as communication channel capacity, UAV speed, and security concerns limit available bandwidth. FANET protocols must ensure sufficient bandwidth to transmit high-resolution data in real-time despite these constraints. [6]

2.6.FANET Communication:

There are three types of FANET COMMUNICATION:

2.6.1. UAV-TO-UAV (U2U) communication:

UAVs communicate with one other through regular data packet exchanges to meet mission requirements. However, limited transmission ranges need multi-hop connection

with other UAVs. This is vital for expanding coverage of a given area of interest. U2U communication often relies on line-of-sight (LoS) because to the absence of obstacles in the sky between UAVs. However, line-of-sight is not always assured, particularly when UAVs are near tall buildings or mountains.

2.6.2. UAV-TO-GROUND (U2G) communication:

To improve UAV control, ground-based systems (GBSs) are used to exchange essential control and command messages. GBSs connect several groups of UAVs together. UAVs may interact with GBSs, reducing network congestion and improving throughput and connection. When UAVs fly at high altitudes, the LoS dominates U2G networks. At low altitudes, UAVs cannot achieve LoS with GBSs because to ground impediments that cause reflections and diffraction.

2.6.3. Satellite communication (SATCOM):

UAVs are frequently used in tough situations like as the seas and mountains, where GBS installation is problematic. For FANETs with severe network partitioning, a centralized entity is necessary to provide continuous connectivity. Satellites can effectively manage UAVs and provide LoS coverage, enabling Satellite Communication (SATCOM). SATCOM is useful for both facilitating the interchange of important data between UAVs and transmitting gathered information to a radio station stationed far away on the ground. However, this is not a budget-friendly option.

TABLE 3 presents a quick comparison among the stated kinds of FANET communication. [7]

Table 3 FANET communication comparison [7]

	U2U	U2G	SATCOM
LoS	High	Medium	High
Cost	Cheaper	Expensive	Highly expensive
Coverage	Medium	Large	Huge
Exploitation	Short-term	Mid-term	Long-term

2.7.FANETS applications:

FANETs have three primary uses, which are as follows:

2.7.1. Multi-UAV cooperation:

Figure 13, box A displays applications that include many UAVs:

- Detection of Targets which Utilizing thermal and vision cameras, UAVs can detect objects and individuals.
- Disaster Situations Tracking and Monitoring which UAVs aid in assessing the movement of floods and identifying vulnerable buildings post-earthquake for prioritized rescue efforts.
- Response to Emergencies which UAVs monitor construction progress for safety and can provide temporary wireless coverage during network outages, among other emergency applications.

2.7.2. UAV-to-Ground tasks:

Figure 13, box B illustrates UAV-to-ground cooperation applications:

- Civilian and Public Applications which Small quadcopters, due to their cost-effectiveness and flexibility, are extensively utilized in various civilian and public applications, surpassing ground-based infrastructure advantages.
- Search and Rescue Operations which UAVs are indispensable in search and rescue missions (SAR), offering significant advantages in ensuring public safety, managing disasters, and assessing critical infrastructure safety during events like floods, earthquakes, and terrorist attacks. Providing communication coverage in such scenarios is crucial. In instances of public communication network disruptions, UAVs can swiftly issue disaster alerts and expedite rescue efforts. They are capable of transporting medical supplies to inaccessible areas and significantly accelerating SAR operations in scenarios such as avalanches, wildfires, and search missions for missing individuals.

2.7.3. UAV-to-VANET collaborations between UAVs and vehicles:

- Traffic Monitoring with FANETs which UAVs replace labor-intensive methods for monitoring road traffic incidents, providing rapid detection and reporting, along with real-time video capture for enhanced road safety.

- Data Packet Delivery which UAVs act as airborne relays, transporting data from ground devices to remote control centers using the Load-Carry-and-Delivery (LCAD) paradigm.
- Route Guidance which UAVs assist in VANETs by improving route guidance and routing efficiency through multi-hop relays between vehicles and UAVs. [8]

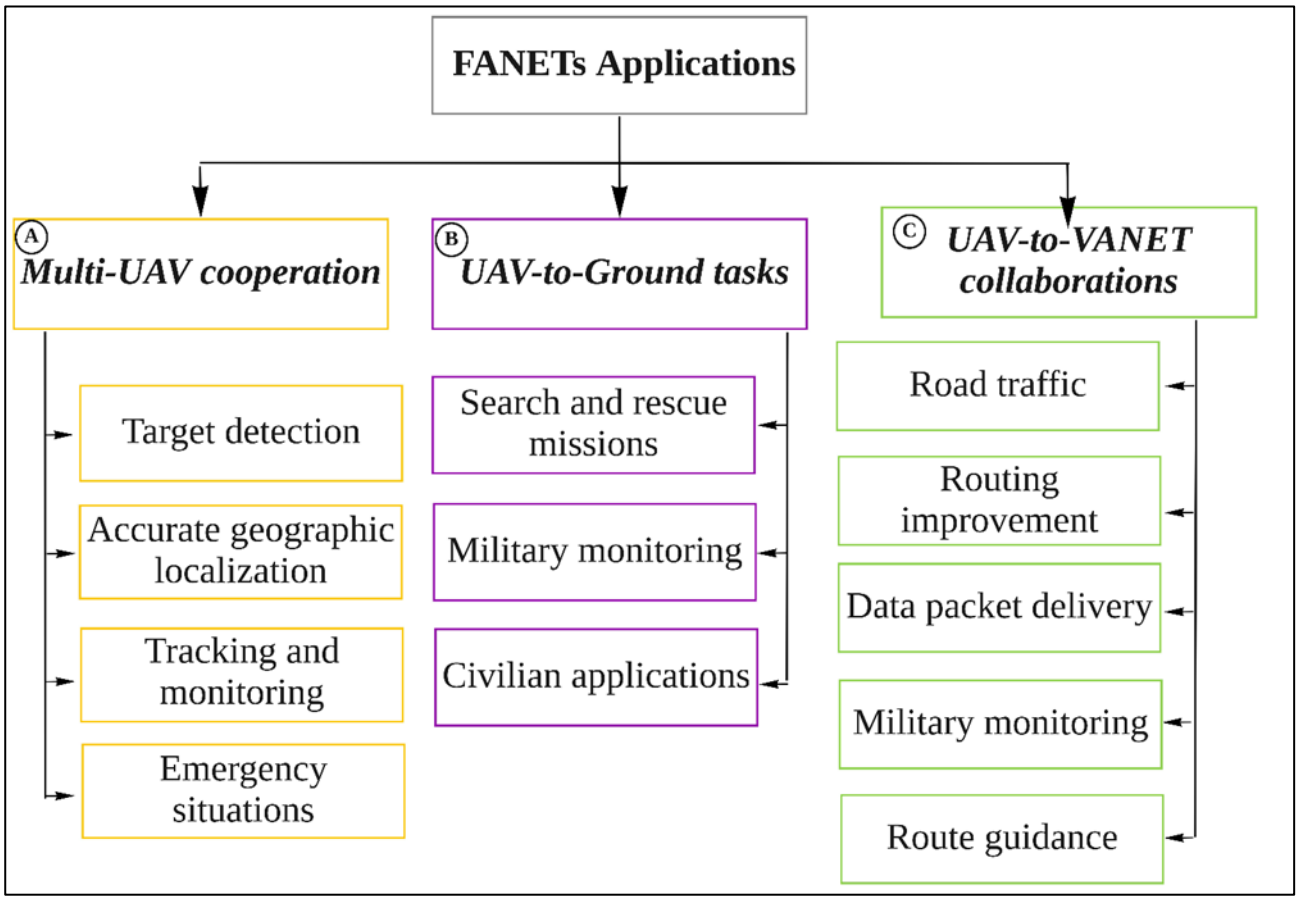

Figure 2-3 FANET applications: Multi-UAV cooperation, UAV-to-Ground cooperation, and UAV-to-VANET cooperation [8]

2.8.FANET Routing Protocols:

This section defines several routing protocols at the network layer for FANETs. Figure 2.4 depicts one possible classification of FANET routing protocols, which is addressed further below.

2.8.1. Position-based:

These protocols utilize GPS information to determine node positions, employing reactive, predictive, greedy, and hierarchical methods to establish sender-receiver positions in advance. Relay nodes are dynamically selected, with packets broadcast blindly until neighboring nodes receive them. Forwarding decisions are made based on dynamic forwarding delay values, with nodes closest to the destination becoming the next forwarder.

2.8.2. Topology-based:

These protocols forward packets along optimal paths determined by network topology and link-state information, including static, proactive, reactive, and hybrid approaches.

2.8.3. Delay-tolerant networks (DTNs):

DTN-based routing protocols address intermittently connected mobile nodes with high latency and low data rates, including deterministic, social network, and stochastic methods.

2.8.4. Heterogeneous:

FANETs interact with various ground networks, requiring heterogeneous routing protocols to exchange data between moving and fixed nodes, supporting network extension and sub-network assistance coverage.

2.8.5. Cluster-based:

Nodes with similar characteristics form clusters, each led by a cluster head for communication processing. Cluster-based routing protocols are classified as probabilistic and deterministic.

2.8.6. Swarm-based:

This routing technique mimics social behaviors of animals to find optimal paths and manage topology.

Selecting the best routing protocol depends on factors such as routing approach, mobility models, simulation tools, and performance metrics. Reactive and proactive approaches are preferred in dynamic FANETs, while hybrid protocols are suitable for large-scale monitoring applications. [8]

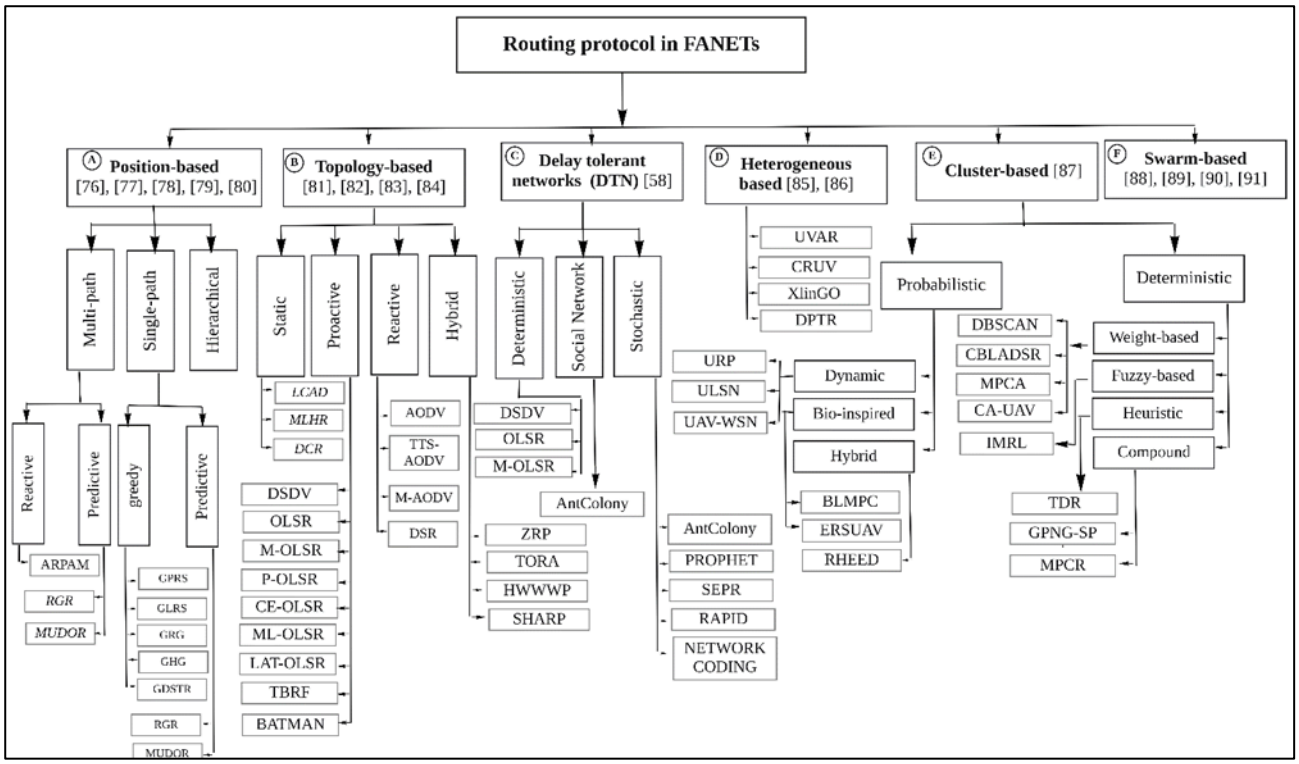

Figure 2-4 FANET Routing Protocols: position-based, topology-based, DTN, heterogeneous, cluster-based, and swarm-based [8]

2.9.Conclusion:

In conclusion, the exploration of FANETs unveils a landscape brimming with possibilities and challenges, poised to reshape the way we perceive and leverage aerial networks. From their inception as a response to the need for interconnected UAV systems to their evolution into versatile platforms for multi-UAV cooperation and beyond, FANETs have traversed a remarkable trajectory.

As we navigate the complexities of FANET design, communication protocols, and application domains, it becomes evident that unlocking the full potential of these networks demands a multidisciplinary approach and continuous innovation. Whether it's optimizing routing protocols for dynamic environments, addressing energy constraints in UAV platforms, or simulating FANET behaviors in virtual environments, each aspect contributes to the broader endeavor of advancing aerial networking capabilities.

Looking ahead, the future of FANETs holds promise for groundbreaking advancements in fields ranging from disaster response and surveillance to transportation

and infrastructure monitoring. By embracing the inherent challenges and embracing collaborative research efforts, we can propel FANET technology towards new frontiers, ushering in an era of unprecedented connectivity and efficiency in the skies.

Chapter 03

Simulation

3. Chapter 03: Simulation

3.1.Introduction:

The theoretical concepts about FANET networks are presented in the previous chapters. In this chapter, we will introduce the fundamental ideas behind simulating data transmission between drones in a FANET network using the Network Simulator-2 (NS-2) modeling tool.

In order to start the simulation, we installed Ubuntu 18.04 on our computer in order to install the open-source NS-2 logic.

3.2.Simulation:

Scientists, engineers, the military, and others utilize simulation as a tool to examine the effects of an action on an element without actually conducting the experiment on the element. It reduces risk and saves money by avoiding the expense of several real-world experiments [9]. Within the discipline of computer science, it denotes the process of executing a software on a computer or network to replicate the outcomes of an operation on an entity without carrying out the experiment on the actual entity. To accomplish the specified effects, this entails executing the intended application on the identified entities. However, due to the low adherence to reality, many experimental instances are impractical. Consequently, simulation is employed to examine an element's behaviors in a way that is comparable to the real research, enabling the acquisition of the anticipated outcomes.

3.3.Network simulation:

The modeling of network activity in a virtual environment is known as network simulation. Modifiable and repeatable files specify protocols and experiments. In performance studies of communication systems, simulation is quite prevalent. It enables the production of every element of the physical world in a virtual environment, including node placements, node motions, the surrounding region, and communication protocols. Simulations must be carried out in a setting that is similar to reality in order to obtain a better performance analysis of the systems. The virtual simulation environment consists of the attenuations, mobility model, and signal propagation model. Our research focuses on simulating ad hoc drone networks, or FANETs. [10]

Simulation is frequently used to gauge a computer network's and a protocol's performance. It is essential when theoretical research is extremely complicated and experimental is very expensive. By using a method where software (a simulator) controls a network's activity, an interactive virtual network may be constructed on a screen and its operation observed. [11]

3.4.Simulation of FANETs:

As will be outlined in more depth later in the chapter, in our project, we will use the network simulation tools NS-2.35 to simulate the transmission of a data in FANETs. Drones (unmanned aerial vehicles) are flying entities or vehicles that communicate with one another in a suitable environment to enable data transmission. we have simulated a whole data transmission environment made up of these drones. To guarantee 3D mobility, these UAVs go inside the area that the 3D PATCH provides.

3.5.FANET simulators:

There are several network simulators used for simulating FANETs (Flying Ad Hoc Networks). Some popular ones include:

3.5.1. ns-2 (Network Simulator 2):

A straightforward event-driven simulation program that has shown helpful in researching the dynamic nature of communication networks is Network Simulator (Version 2), or NS2. NS2 may be used to simulate both wired and wireless network functions and protocols, such as routing algorithms, TCP, and UDP. Generally speaking, NS2 gives users the ability to define these kinds of network protocols and mimic the behaviors that go along with them. Since its inception in 1989, NS2 has consistently garnered popularity in the networking research community due to its modular and flexible design. Since then, the tool has seen a number of revolutions and updates that have demonstrated its increasing maturity due to the significant contributions made by the industry participants. These include Cornell University and the University of California, who created the REAL network simulator1, the basis for NS. Via the Virtual Inter Network Test bed (VINT) project, the Defense Advanced Research Projects Agency (DARPA) has funded the development of NS from 1995.The National Science Foundation (NSF) has currently joined the development

ride. Finally, but just as importantly, NS2's community of researchers and developers is always working to maintain its strength and adaptability. [12]

3.5.2. OMNeT++:

OMNeT++ is an extensible, modular, component-based C++ simulation library and framework, primarily for building network simulators. “Network” is meant in a broader sense that includes wired and wireless communication networks, on-chip networks, queueing networks, and so on. Domain-specific functionality such as support for sensor networks, wireless ad-hoc networks, Internet protocols, performance modeling, photonic networks, etc., is provided by model frameworks, developed as independent projects. OMNeT++ offers an Eclipse-based IDE, a graphical runtime environment, and a host of other tools. There are extensions for real-time simulation, network emulation, database integration, SystemC integration, and several other functions. OMNeT++ is distributed under the Academic Public License QualNet: QualNet is a commercial network simulation software developed by Scalable Network Technologies. It offers a comprehensive set of features for simulating wireless networks, including FANETs. [13]

3.5.3. OPNET:

A tool for simulating any kind of network's behavior and performance is the OPNET Network Simulator. Opnet Network Simulator's strength and adaptability set it apart from other simulators. Pre-built models of devices and protocols are offered by IT Guru. It lets you model and build various network topologies. It is not possible to add new protocols or change the functionality of ones that already exist. The collection of devices and protocols is set. [14]

3.5.4. MATLAB/Simulink:

A block diagram environment for model-based design and multidomain simulation is called Simulink®. It facilitates continuous testing and verification of embedded systems, automated code creation, simulation, and system-level design. For modeling and simulating dynamic systems, Simulink offers a graphical editor, modifiable block libraries, and solver options. Because of its integration with MATLAB®, you may export simulation results to MATLAB for additional analysis and include MATLAB algorithms into models. [15]

These simulators vary in terms of features, complexity, and ease of use, so the choice depends on specific requirements and preferences for simulation projects involving FANETs.

3.6.Basic architecture of NS-2:

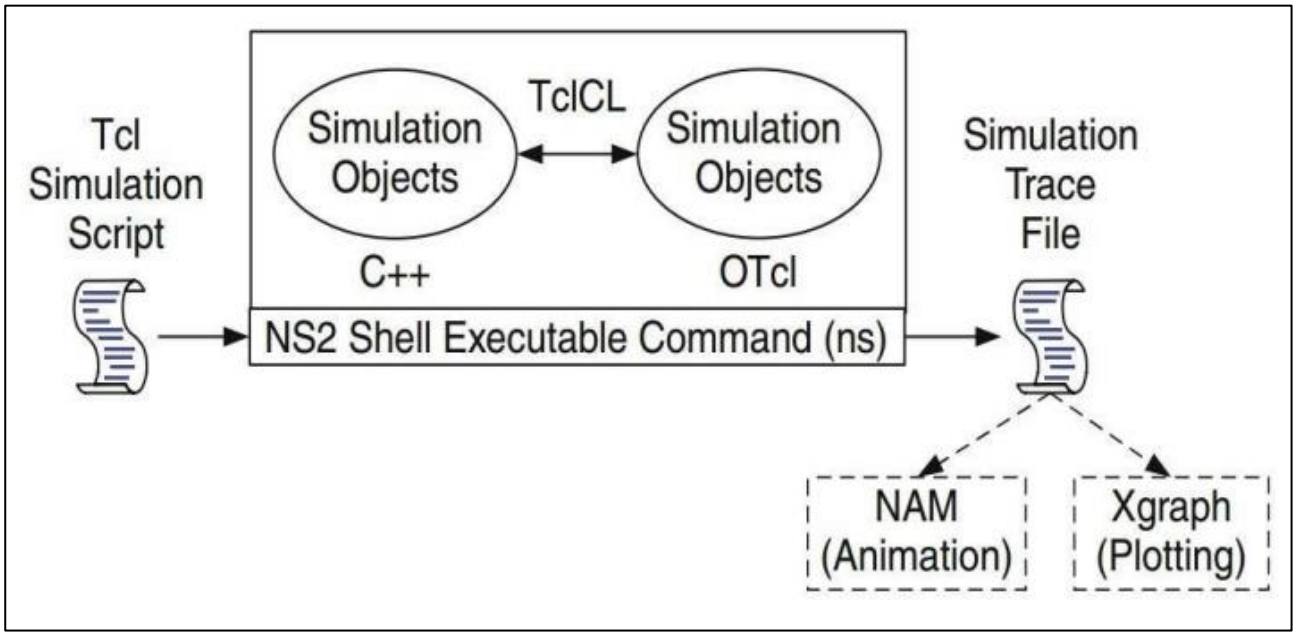

Figure 3-1 NS-2 architecture [12]

The NS2's fundamental architecture is seen in Figure 2.1. The executable command ns, which accepts as input the name of a Tcl simulation scripting file, is made available to users of NS2. The name of a Tcl simulation script that initiates a simulation is fed by users as an input parameter to the NS2 executable command ns.

Usually, a simulation trace file is created, which is then used to plot data and/or produce animation. The two main languages of NS2 are Object-oriented Tool Command Language (OTcl) and C++. The OTCl sets up simulation by assembling and configuring the objects as well as scheduling discrete events, whereas the C++ describes the internal mechanism (i.e., a backend) of the simulation objects (i.e., a front end). TclCL is used to connect the C++ with the OTcl. Variables in the OTcl domains that are mapped to a C++ object are sometimes called handles.

There are several built-in C++ objects available in NS2. It is recommended to utilize a Tcl simulation script to build up a simulation utilizing these C++ objects. However, these

items might not be adequate for advanced users. It is required that they create their own C++ objects and utilize an OTcl configuration interface to group these things together. Following simulation, NS2 produces simulation results that are either text- or animation-based. Graphing and interactive tools like XGraph and Network Animator (NAM) are utilized to understand these data. Users can take a relevant portion of text-based data and convert it into a more imaginable presentation in order to examine a certain behavior of the network. [12]

3.7.UBUNTU installation:

3.7.1. Create Ubuntu Installation Media:

- Download the Ubuntu ISO from the official Ubuntu website.
- Use Rufus software to create a bootable USB drive with the Ubuntu ISO.

3.7.2. Prepare Windows for Dual Boot:

- Open Disk Management in Windows.
- Shrink the volume of our main partition to create free space for Ubuntu (at least 20GB).

3.7.3. Disable Secure Boot:

- Restart my computer and enter the BIOS settings.
- Disable Secure Boot in the BIOS settings.

3.7.4. Install Ubuntu:

- Insert the Ubuntu USB drive and restart computer.
- Boot from the USB drive.
- Select "Install Ubuntu" from the GRUB menu.

  We follow the on-screen instructions. When asked about installation type, I choose "Install Ubuntu alongside Windows Boot Manager."
- Allocate space for Ubuntu by dragging the divider or entering the size manually.

3.7.5. Configure Installation:

- Set up my time zone, keyboard layout, and user details.
- Begin the installation process and wait for it to complete.
- Once done, remove the USB drive when prompted and restart computer.

3.7.6. Boot Options:

- Upon reboot, see the GRUB menu allowing to choose between Ubuntu and Windows 10.

3.7.7. Post-Installation:

- We update the package lists for upgrades and new package installations. It ensures that you are installing the latest versions available in the repositories.

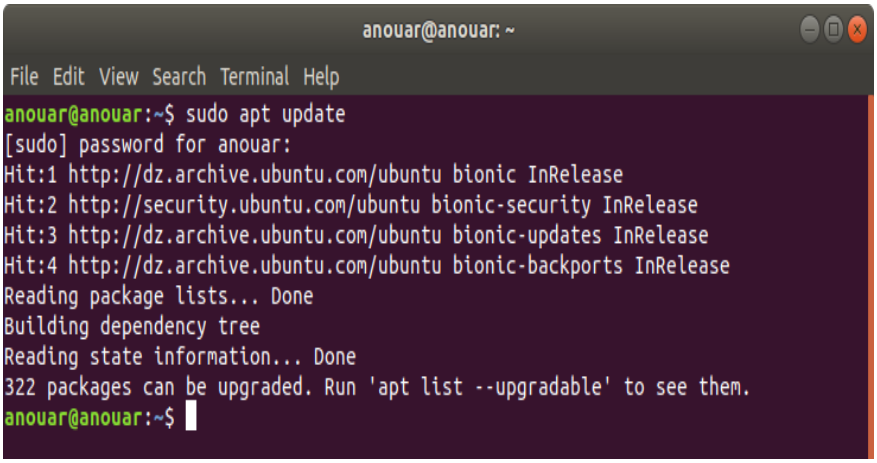

Figure 3-2 updating the package lists for upgrades and new packages installations

3.8.NS2 Installation on Ubuntu:

3.8.1. Install Required Dependencies:

- Install the necessary development tools and libraries by running.

```
→ sudo apt install build-essential autoconf automake libxmu-dev
```

- These packages are essential for compiling NS2 and its dependencies.

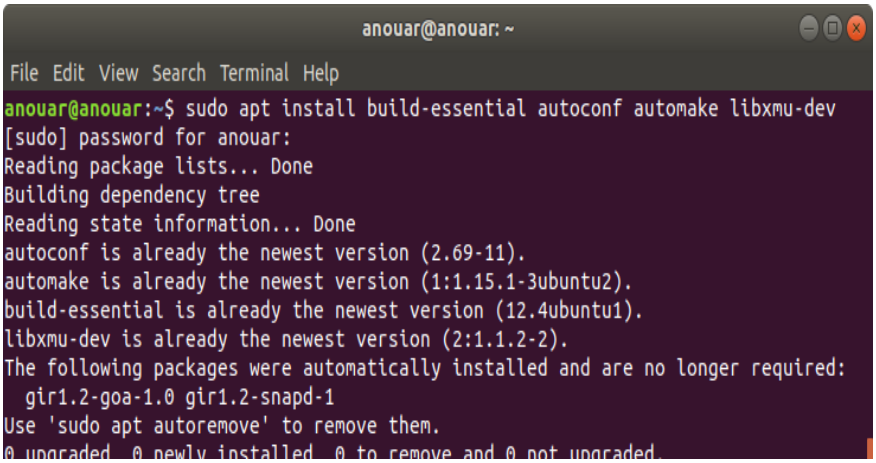

Figure 3-3 installing essential packages for compiling ns-2

3.8.2. Download NS2:

- Download the NS2 package from the official repository. Using the terminal run :

```
→ wget http://sourceforge.net/projects/nsnam/files/allinone/ns-allinone-2.35/ns-allinone-2.35.tar.gz
```

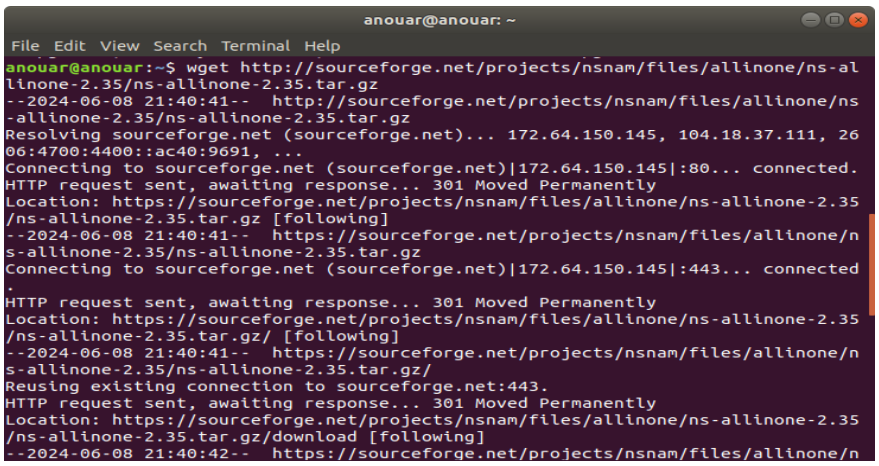

Figure 3-4 downloading ns-2

3.8.3. Extract the Downloaded Package:

- After downloading, we extracted the contents of the tarball using:

```
→ tar -xvzf ns-allinone-2.35.tar.gz
```

- This command unpacks the archive, creating a directory named **ns-allinone-2.35** with all the necessary files.

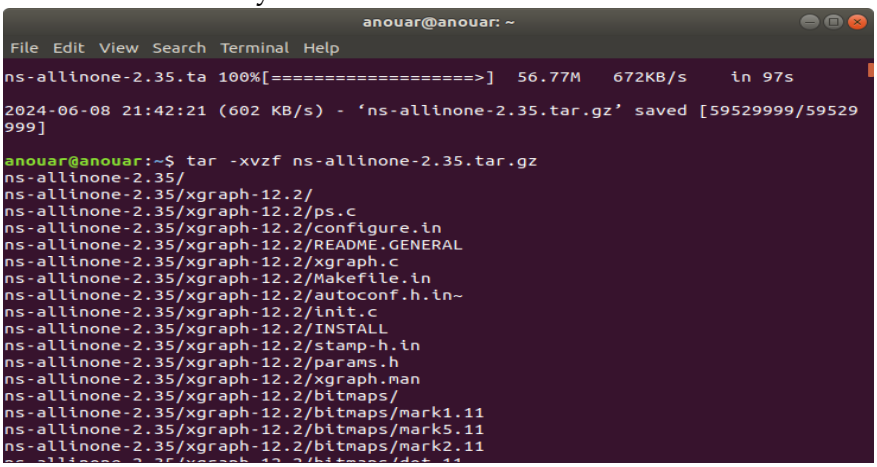

Figure 3-5 extracting the contents

3.8.4. Install GCC and G++ and edit them in Makefile:

- We installed the GCC and G++ compilers, which are necessary for building NS2. we ran:

```
→ sudo apt install gcc g++
```

to ensure these compilers were installed on my system.

- We changed the path to the ns-2.35 directory and I modified CC to gcc 4.8 and CPP to g++ 4.8.

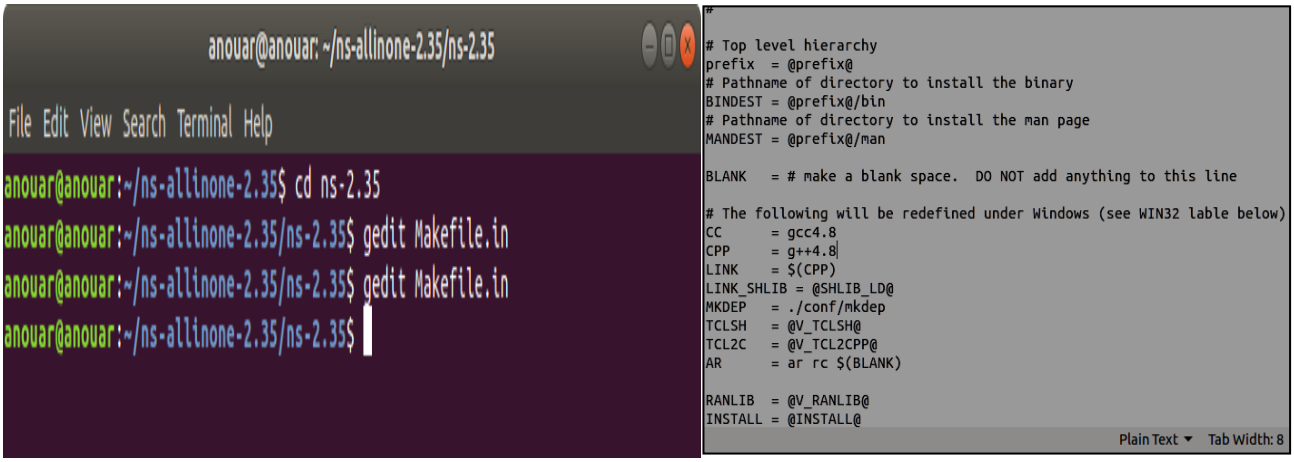

Figure 3-6 changing the path of ns-2.35

- In the next step, we modified the ls.h file located in the following path:
- In the ls.h file, we replaced the word 'erace' with 'this->erace' on line 137, then save the file as shown below:

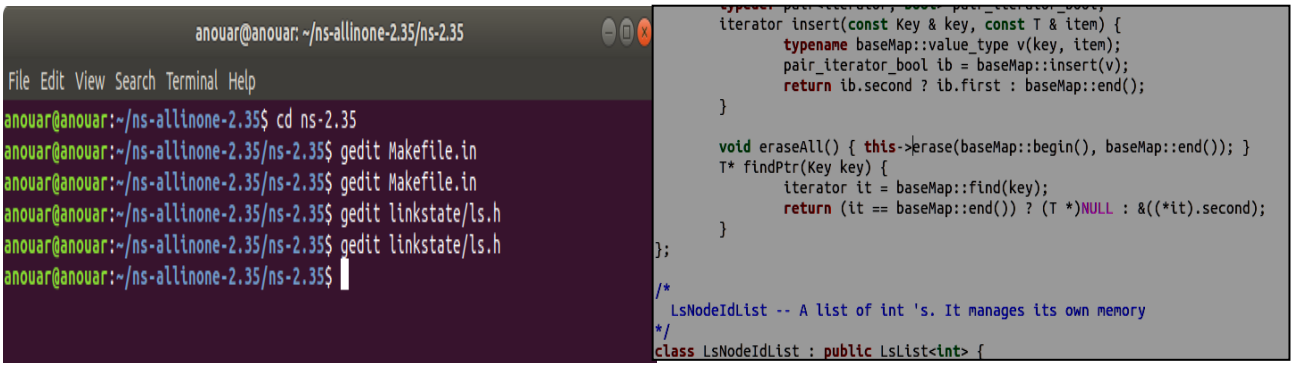

Figure 3-7 edit ls.h file

3.8.5. NS2 installation:

- we navigated to the extracted directory using:

```
→ cd ns-allinone-2.35
```

and started the installation process by running:

```
→ ./install
```

- This script compiles and installs NS2 along with its components.

3.8.6. Set Environment Variables:

- To ensure the system recognizes NS2 commands, we needed to set the environment variables. we edited the **.bashrc** file by running, Next, at the end of the (bashrc) file, add the code shown below, replacing the word "/path_to/" with the directory path of the ns-2 folder (/home/anouar/), then save the changes made:

```
      # LD_LIBRARY_PATH
```

```
1)  OTCL_LIB=/path_to/ns-allinone-2.35/otcl-1.14/NS2_LIB=/path_to/ns-allinone-
    2.35/lib/USR_LOCAL_LIB=/usr/local/lib/
2)  export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$OTCL_LIB:$NS2_LIB:$USR_LOCAL_LIB

# TCL_LIBRARY

3)  TCL_LIB=/path_to/ns-allinone-2.35/tcl8.5.10/library/USR_LIB=/usr/lib/
4)  export TCL_LIBRARY=$TCL_LIBRARY:$TCL_LIB:$USR_LIB

# PATH

5)  export  PATH=$PATH:/path_to/ns-allinone-2.35/xgraph-12.2/:/path_to/ns-allinone-
    2.35/bin/:/path_to/ns-allinone-2.35/tcl8.5.10/unix/:/path_to/ns-allinone-
    2.35/tk8.5.10/unix/:/path_to/ns-allinone-2.35/ns-2.35/:/path_to/ns-allinone-
    2.35/nam-1.15/
```

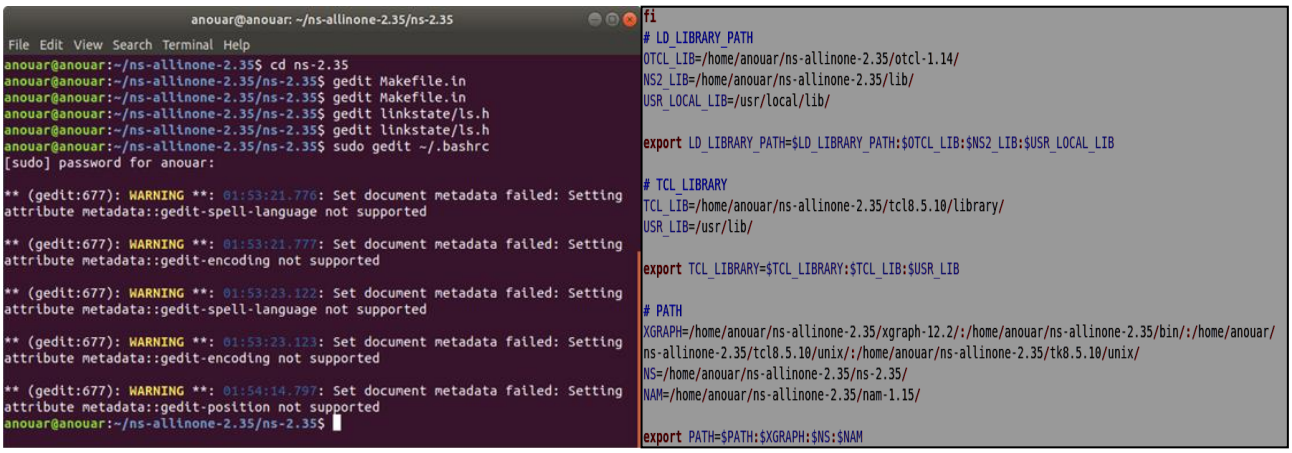

Figure 3-8 3. Set Environment Variables

- In this step, a system reboot is required, the final step is to verify the proper functioning of NS-2. To do this, in the terminal, navigate to the NS-2 directory and run the command './validate':

- To confirm success, type "ns" in the terminal. If the '%' sign is displayed, then NS-2 has been successfully installed.

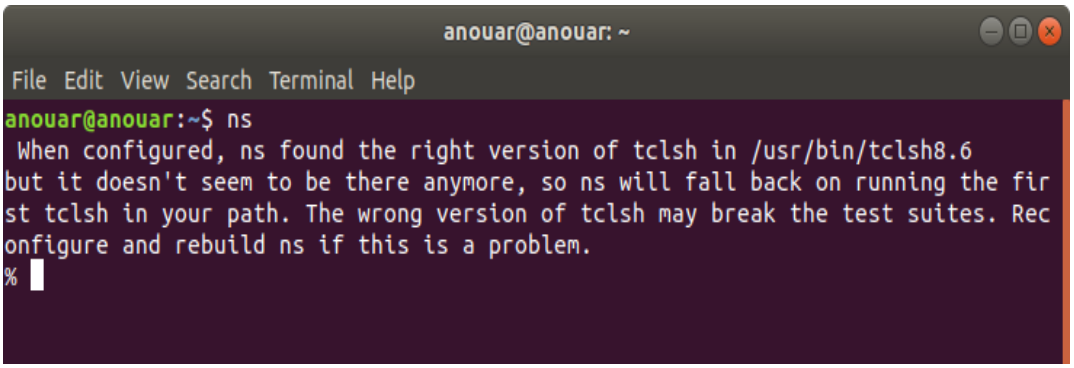

Figure 3-9 runing test ns

3.9.The 3D PATCH:

The open-source simulator ns2 has more than 300,000 lines of code spread over several libraries. The position of a mobile node is represented in the code, among other coding norms. Coordinates of mobile nodes are denoted by the variables X_, Y_, and occasionally Z_. Since Z_ is meant to be used only to represent 2D environments, it is set to zero whenever it is utilized [16]. So The 3D PATCH is necessary to install, as our work involves simulating a FANET network, where each drone is located in three dimensions (X, Y, and Z).

The installation steps for the 3D PATCH are as follows:

- Run validation tests to verify that the build was successful. I believe the installation concludes with instructions on how to proceed. It's important to note that to successfully use the allinone version of ns-2, it's necessary to configure certain environment variables.
- Obtain the ns2_3D.patch patch.
- Apply the patch. This can be done as follows:

```
→ Navigate to the ns-allinone-2.35/ns-2.35 directory
→ patch -p1 <where-the-patch-is/ns2_3D.patch
```

- Rebuild the simulator (now modified) as in step 2 above (i.e., rerun ./install). (You can also rerun the validation, as in step 3 above.)
- To generate 3D mobility scenarios, use the s etdest utility, for example:

```
→ ns-2.35/independants-utiles/cmu-scen-gen/setdest/setdest
  -n 3 -p 0.00 -s 10.00 -t 200.00 -x 1500.00 -y 1500.00 -z
  1500.00 > data.jpg
```

- The network animator name is not available in 3D.

3.10. Conclusion:

In this chapter, the simulation network has been clearly defined, its many tools have been briefly described, and the installation of NS-2 together. We also witnessed the installation of the PATCH 3D for the purpose of moving unmanned aerial vehicles (UAVs) in a three-dimensional space. We will see the design and execution of the FANET data transmission in the next chapter, all while building on the principles covered in this chapter.

Chapter 04

Design and Implementation of Data Transmission in FANETs

4. Chapter 04: Design and Implementation of Data Transmission in FANETs

4.1.Introduction:

This chapter focuses on the design and implementation of data transmission in Flying Ad-hoc Networks (FANETs). It provides a comprehensive overview of the architecture and protocols involved in enabling reliable data transmission in FANETs. The chapter includes a class diagram and a flowchart diagram to illustrate the communication flow between a UAV (Unmanned Aerial Vehicle. Additionally, it presents a Tcl script that simulates data transmission in FANETs, defining various simulation parameters, and it present also the visualization execution using NAM with Qr code to watch the execution.

4.2.Designing data transmission in FANETs:

4.2.1. Class diagram:

Our class diagram represents a conceptual model for a FANET (Flying Ad Hoc Network) composed of UAVs (Unmanned Aerial Vehicles) communicating through standardized wireless protocols to transmit data. The model includes three primary classes:

- UAVs: This class represents individual UAVs in the network. Each UAV is identified by a unique id and has attributes such as name, model, and a reference to the communicationProtocol it uses. Additionally, the UAVs handle dataPackets for communication purposes.
- CommunicationProtocols: This class defines the protocols that facilitate wireless communication between the UAVs. Each protocol is uniquely identified by an id, and includes attributes like name and standard, which refers to the IEEE 802.11 standard for wireless communication.
- DataPackets: This class encapsulates the data being transmitted over the network. Each data packet has a unique id and is associated with a communicationProtocol, indicating the method used for its transmission. It also includes attributes such as size and type to detail the packet's characteristics.

The relationships in the diagram indicate that each UAV can use one communication protocol and can handle multiple data packets, while each data packet is associated with a specific communication protocol. This structure allows for a clear representation of how data is transmitted in a FANET, providing a framework for implementation in NS-2 (Network Simulator 2).

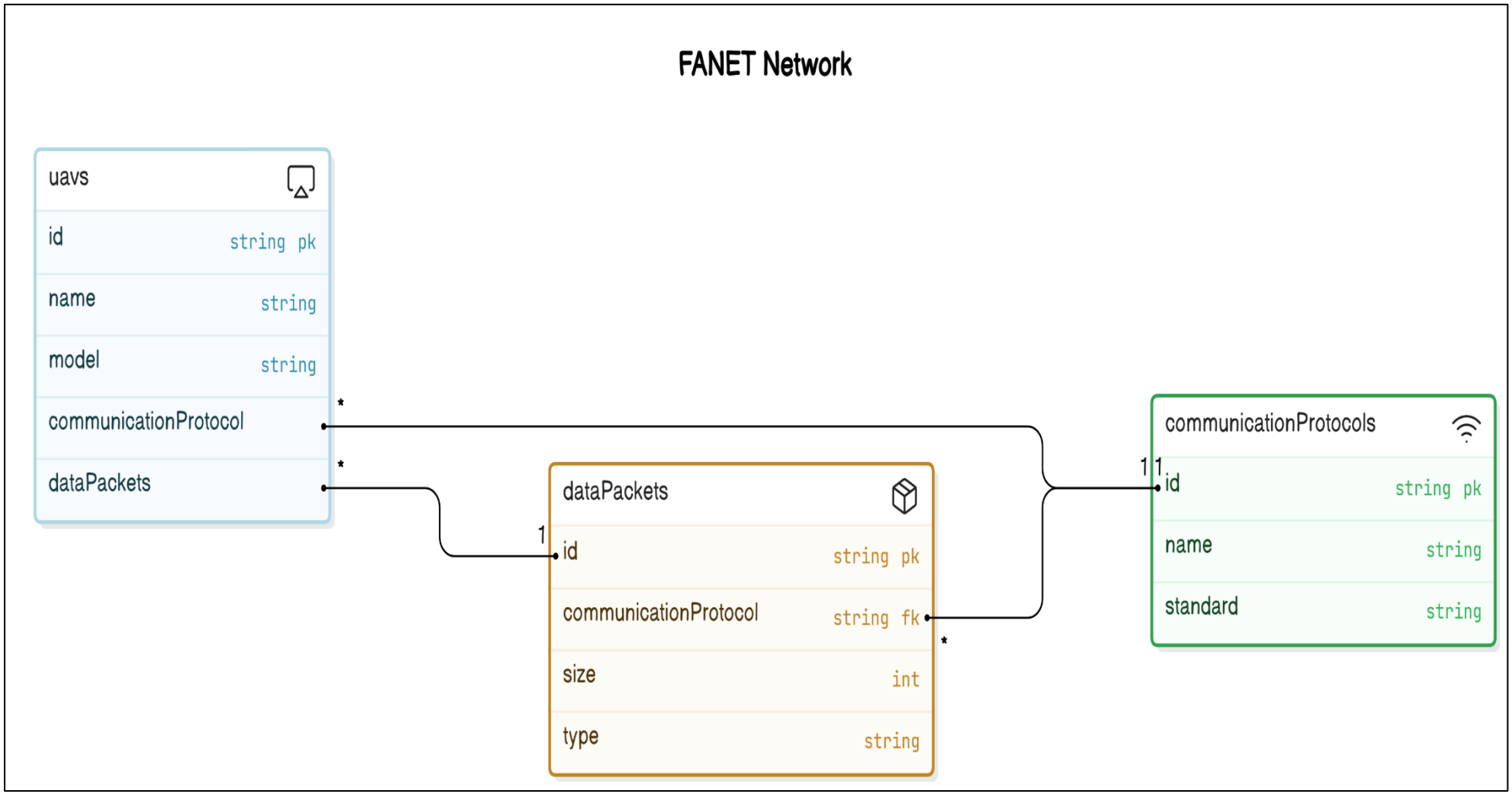

Figure 4-1 class diagram

4.2.2. Flowchart diagram:

Our flowchart provides a clear, step-by-step representation of the process for setting up and simulating a FANET with ten UAVs. It covers the initialization of the network, establishment of communication protocols, transmission of data packets, and the final simulation using NS-2. Each step is crucial for understanding the network's configuration, operation, and evaluation, offering a comprehensive view of how a FANET operates and is analyzed in a simulated environment. **(If the image does not display properly, you can scan the QR code shown below)**

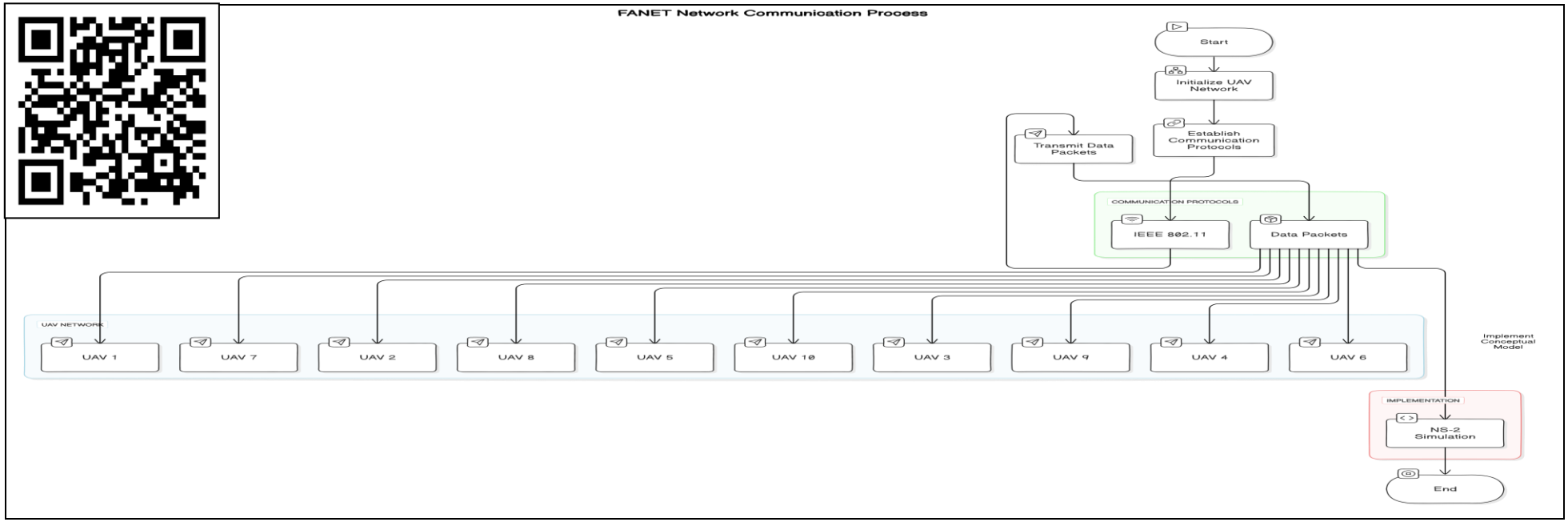

Figure 4-2 flowchart diagram

4.3.Simulation parameters values:

➢ **Simulator and Version:** NS2
➢ **NS2 Patches Applied:** 3D-patch (in chapter 3)
➢ **Channel Type:** Wireless Channel
➢ **Radio Propagation Model:** Propagation/FreeSpace
➢ **Grid Setup:** The topography is a cube with dimensions 1500x1500x1500 units.
➢ **Antenna:** Antenna/OmniAntenna
➢ **Routing Protocols Used:** AODV (Ad hoc On-Demand Distance Vector)
➢ **Queue Type:** Queue/DropTail/PriQueue
➢ **Number of Nodes:** 10
➢ **Type of Traffic:** CBR (Constant Bit Rate) over UDP
➢ **CBR Parameters:** Packet Size: 512 bytes (Interval: 0.1 seconds)
➢ **MAC Layer Protocol:** Mac/802_11
➢ **Simulation Duration:** 448.00 seconds
➢ **Mobile Node Speed:** Nodes are set to move to new positions with a speed of 10.0 units per second.
➢ **Initial Distance Between Nodes:** Positions are predefined, so the initial distance varies.
➢ **Altitude:** Nodes are positioned in a 3D space with specified Z coordinates.
➢ **Time Interval Between New Random Positions for Nodes:** Nodes are given specific times to set new destinations.

4.4.The Tcl script implemented for data transmission in FANETs:

```
# ======================================================================
# Define options
# ======================================================================
set val(chan)      Channel/WirelessChannel
set val(prop)      Propagation/FreeSpace
set val(netif)     Phy/WirelessPhy
set val(mac)       Mac/802_11
set val(ifq)       Queue/DropTail/PriQueue
set val(ll)        LL
set val(ant)       Antenna/OmniAntenna
set val(x)         1500
set val(y)         1500
```

```
set val(z)          1500
set val(ifqlen)     100
set val(seed)       0.0
set val(adhocRouting) AODV
set val(nn)         10
set val(stop)       448.00
# ============================================================================
# Main Program
# ============================================================================
# Initialize Global Variables
# Create simulator instance
set ns_ [new Simulator]
# Setup topography object
set topo [new Topography]
# Create trace object for ns
$ns_ use-newtrace
set tracef_3D [open 100.tr w]
$ns_ trace-all $tracef_3D
set namtrace [open NAM.nam w]
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)
# Define topology
$topo load_cube $val(x) $val(y) $val(z) 1
# Create God
set god_ [create-god $val(nn)]
# Define how node should be created
$ns_ node-config -adhocRouting $val(adhocRouting) \
                 -llType $val(ll) \
                 -macType $val(mac) \
                 -ifqType $val(ifq) \
                 -ifqLen $val(ifqlen) \
                 -antType $val(ant) \
                 -propType $val(prop) \
                 -phyType $val(netif) \
                 -channelType $val(chan) \
                 -topoInstance $topo \
                 -agentTrace ON \
                 -routerTrace ON \
                 -macTrace ON \
                 -movementTrace ON

# Create the specified number of nodes
```

```
for {set i 0} {$i < $val(nn)} {incr i} {
   set node_($i) [$ns_ node]
   $node_($i) random-motion 0
}
# Define node movement model
# Positions and movements of the nodes
$node_(0) set X_ 93.908255228986
$node_(0) set Y_ 1101.653449154440
$node_(0) set Z_ 1000.729306287061
$node_(1) set X_ 245.018595239117
$node_(1) set Y_ 95.635823531135
$node_(1) set Z_ 568.668820146265
$node_(2) set X_ 767.824777333893
$node_(2) set Y_ 1092.776793590182
$node_(2) set Z_ 1393.309670455850
$node_(3) set X_ 43.021823383993
$node_(3) set Y_ 1399.807036076641
$node_(3) set Z_ 490.807380198486
$node_(4) set X_ 1084.782605090826
$node_(4) set Y_ 358.355353828034
$node_(4) set Z_ 346.205739190521
$node_(5) set X_ 1478.645704700217
$node_(5) set Y_ 917.578970048676
$node_(5) set Z_ 197.335292479429
$node_(6) set X_ 1487.222048533658
$node_(6) set Y_ 130.345067617431
$node_(6) set Z_ 540.677531729668
$node_(7) set X_ 231.617125141519
$node_(7) set Y_ 434.575767929610
$node_(7) set Z_ 1135.575942625244
$node_(8) set X_ 1349.569900708864
$node_(8) set Y_ 714.750546419088
$node_(8) set Z_ 1100.579588655512
$node_(9) set X_ 783.587269714603
$node_(9) set Y_ 1401.486404591513
$node_(9) set Z_ 564.593006725271
$node_(1) color red
$node_(9) shape circle
$node_(9) color green
$ns_ at 0.000000000000 "$node_(0) setdest3d 728.541271420332 547.100973850350 1397.558991027128 10.000000000000"
$ns_ at 0.000000000000 "$node_(1) setdest3d 621.442085541774 1473.609318144326 1183.514598331190 10.000000000000"
$ns_ at 0.000000000000 "$node_(2) setdest3d 201.378724162182 1169.112693559248 1231.382836617443 10.000000000000"
$ns_ at 0.000000000000 "$node_(3) setdest3d 1436.530299670599 1253.991588561388 1260.299547259302 10.000000000000"
```

```
$ns_ at 0.000000000000 "$node_(4) setdest3d 918.761961884445 272.859401594558 228.887725181097 10.000000000000"
$ns_ at 0.000000000000 "$node_(5) setdest3d 884.429007992436 1303.094232930709 1429.071400837705 10.000000000000"
$ns_ at 0.000000000000 "$node_(6) setdest3d 20.225172561322 1005.340687723566 945.500347801501 10.000000000000"
$ns_ at 0.000000000000 "$node_(7) setdest3d 248.055910364638 202.765801030976 1449.984016617917 10.000000000000"
$ns_ at 0.000000000000 "$node_(8) setdest3d 655.278137139476 877.884471113787 733.819560761207 10.000000000000"
$ns_ at 0.000000000000 "$node_(9) setdest3d 1382.289730365450 1099.236846119460 502.382219768013 10.000000000000"
$ns_ at 22.053554867659 "$node_(4) setdest3d 1277.833811307660 1382.279347980885 32.565330917386 10.000000000000"
$ns_ at 39.097126672550 "$node_(7) setdest3d 1272.357759333778 1179.982213055785 1392.506333100916 10.000000000000"
$ns_ at 59.406110821600 "$node_(2) setdest3d 1328.697353221721 1136.190979119326 937.481778440115 10.000000000000"
$ns_ at 67.355000853669 "$node_(9) setdest3d 1085.323158941981 685.709059966608 166.737777921711 10.000000000000"
$ns_ at 80.197671313300 "$node_(8) setdest3d 1124.911520975083 229.695405060575 1442.913785955419 10.000000000000"
$ns_ at 93.153706947009 "$node_(0) setdest3d 179.670628013902 640.950585090956 775.328214459184 10.000000000000"
$ns_ at 128.334633995055 "$node_(9) setdest3d 1265.401110798919 482.882615164757 281.596025934437 10.000000000000"
$ns_ at 140.302770212941 "$node_(4) setdest3d 674.391619575549 205.979758837211 1150.457495892225 10.000000000000"
$ns_ at 142.087625822761 "$node_(5) setdest3d 217.299157101248 234.324115733480 485.255567158716 10.000000000000"
$ns_ at 145.252396351016 "$node_(7) setdest3d 285.636315299256 474.519278872222 196.018670086825 10.000000000000"
$ns_ at 180.101314431784 "$node_(5) setdest3d 1405.870392104871 990.529627950034 494.054387926592 10.000000000000"
$ns_ at 192.426447036346 "$node_(6) setdest3d 464.337385189004 646.865062108054 960.132045662702 10.000000000000"
$ns_ at 196.073399192729 "$node_(4) setdest3d 1366.540317599474 194.823091032715 409.818056640747 10.000000000000"
$ns_ at 225.524664494149 "$node_(7) setdest3d 1306.857791733314 1008.452792724769 812.723261636187 10.000000000000"
$ns_ at 232.720531056264 "$node_(6) setdest3d 1307.651141705437 925.097476060899 1427.460679119142 10.000000000000"
$ns_ at 262.995243743053 "$node_(2) setdest3d 68.623927666606 1202.425692914216 986.605857188003 10.000000000000"
$ns_ at 299.196294883680 "$node_(8) setdest3d 83.162598852210 1014.110579789290 1397.780574618620 10.000000000000"
$ns_ at 327.423715088779 "$node_(7) setdest3d 216.905313188083 707.350254898599 656.047681083923 10.000000000000"
$ns_ at 336.589078600591 "$node_(9) setdest3d 746.662758699229 1277.458646058924 292.678701834070 10.000000000000"
$ns_ at 354.124458701839 "$node_(0) setdest3d 62.019512809147 255.465329163694 451.519878868676 10.000000000000"
$ns_ at 370.194794037827 "$node_(8) setdest3d 226.385036216161 1138.726820672268 23.788451518419 10.000000000000"
$ns_ at 398.154718737730 "$node_(2) setdest3d 184.883745454749 40.193805105118 903.968309805239 10.000000000000"
$ns_ at 412.124947764821 "$node_(7) setdest3d 218.095818743372 722.209071923631 1285.332706695338 10.000000000000"
$ns_ at 426.081459896894 "$node_(0) setdest3d 1451.091749218198 1126.170424265248 983.831079127103 10.000000000000"
$ns_ at 430.684166398067 "$node_(8) setdest3d 1405.393836075520 1464.020258419458 930.346049942004 10.000000000000"
# Setup traffic
for {set i 0} {$i < 5} {incr i} {
   set udp($i) [new Agent/UDP]
   $ns_ attach-agent $node_($i) $udp($i)
}
for {set i 5} {$i < 10} {incr i} {
   set cbr($i) [new Application/Traffic/CBR]
   $cbr($i) set packetSize_ 512
   $cbr($i) set interval_ 0.1
   $cbr($i) attach-agent $udp([expr $i - 5])
}
# Set up connections
for {set i 5} {$i < 10} {incr i} {
   set null($i) [new Agent/Null]
   $ns_ attach-agent $node_($i) $null($i)
   $ns_ connect $udp([expr $i - 5]) $null($i)
}
# Start traffic
for {set i 5} {$i < 10} {incr i} {
   $ns_ at 0.5 "$cbr($i) start"
}
# ============================================================
# End of simulation
```

```
# ============================================================================
$ns_ at $val(stop) "$ns_ halt"
puts "Starting Simulation..."
$ns_ run
```

Our TCL code is designed for simulating data transmission in a Flying Ad-Hoc Network (FANET) using NS-2. It sets up a 3D wireless network environment with specific parameters such as propagation model, MAC type, and antenna type. The simulation initializes 10 nodes in a 1500x1500x1500 cubic area, configures them with AODV routing, and defines their initial positions and movements in 3D space. The script generates UDP agents on five nodes and CBR (Constant Bit Rate) traffic sources attached to these agents, which then transmit data packets at regular intervals to Null agents on the remaining five nodes. Output files are created for detailed event logging (`100.tr`) and visualization (`NAM.nam`). The simulation runs until a specified stop time, capturing the behavior and performance of the FANET under the given conditions.

4.5.Execution Result of the Script:

4.5.1. Graphical Data execution:

In this section we will present the graphical results of the execution of the script we have data packets of execution of the script we have implemented that simulates data transmission in a FANET network.

After the installation of NS-2 in Ubuntu and applying the ns2_3D.patch (chapter 3), we run our TCL code in the terminal using this commands below:

```
→ cd /path/to/file
→ ns namefile.tcl
```

After debugging the tcl file, we have the output file his name NAM.nam. Now, we open the nam generator and go to file -> open and we search in the directory and open the file.

After opening the file we have this first graphic interface:

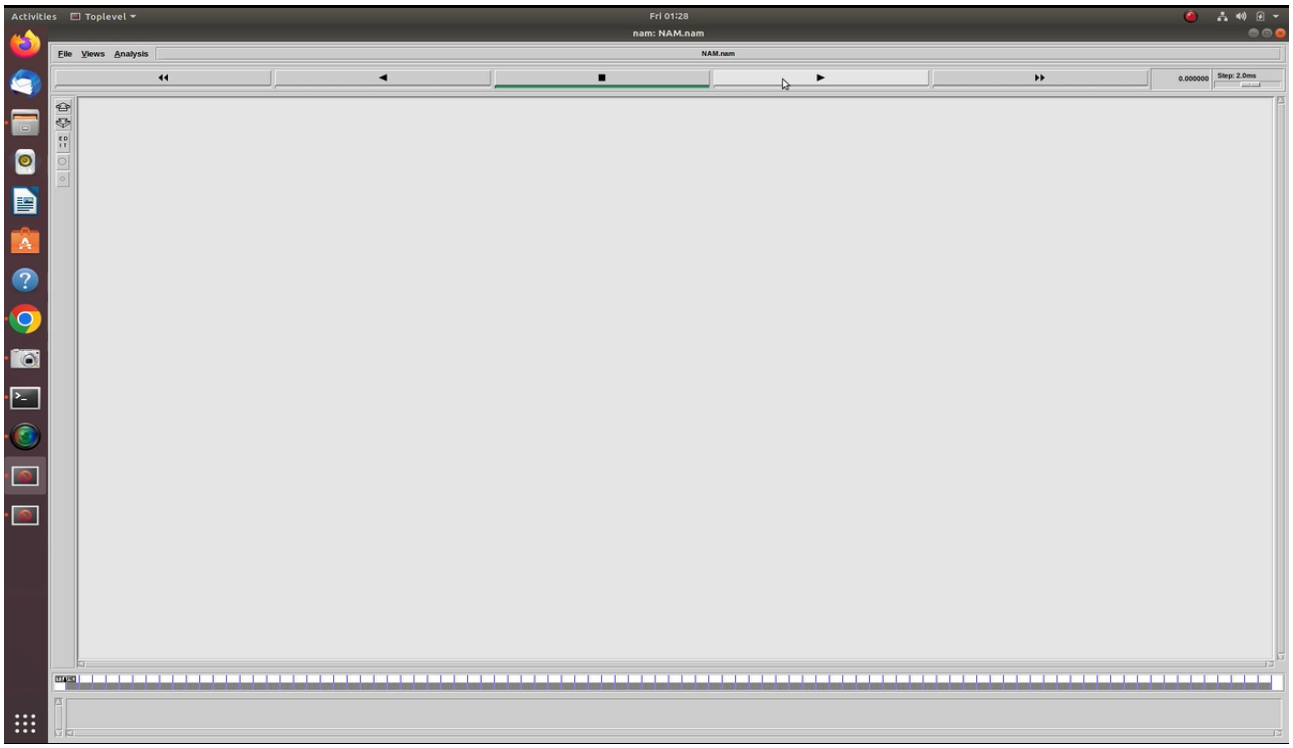

Figure 4-3 first interface when we open our file

The next interface refer to when we clicking in the play-button and starting the execution:

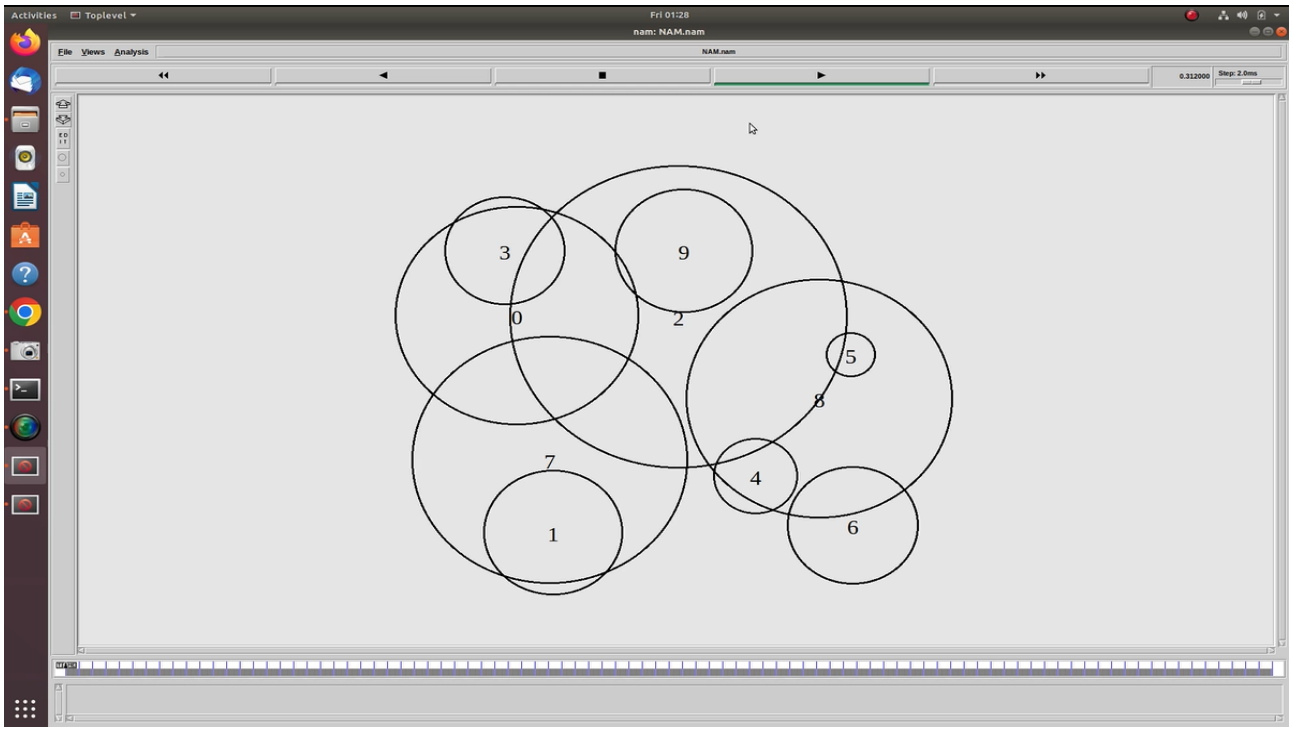

Figure 4-4 starting the excution

In the next interfaces, at a given moment, there will be data transmission.

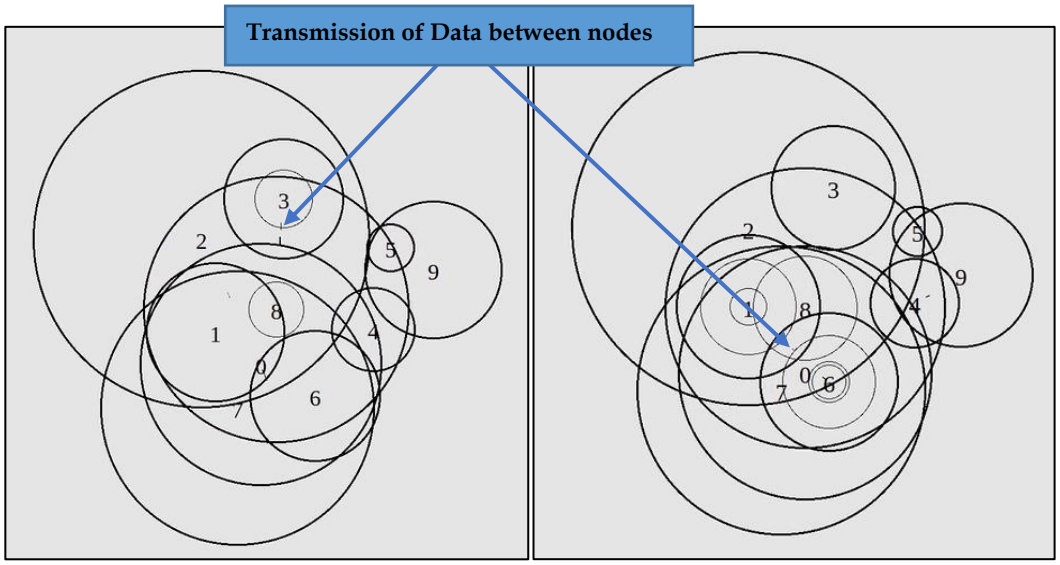

Figure 4-5 a given moment of data transsmission between nodes in fanet network

In the last interface and stopped time (448 ms), all nodes stop moving.

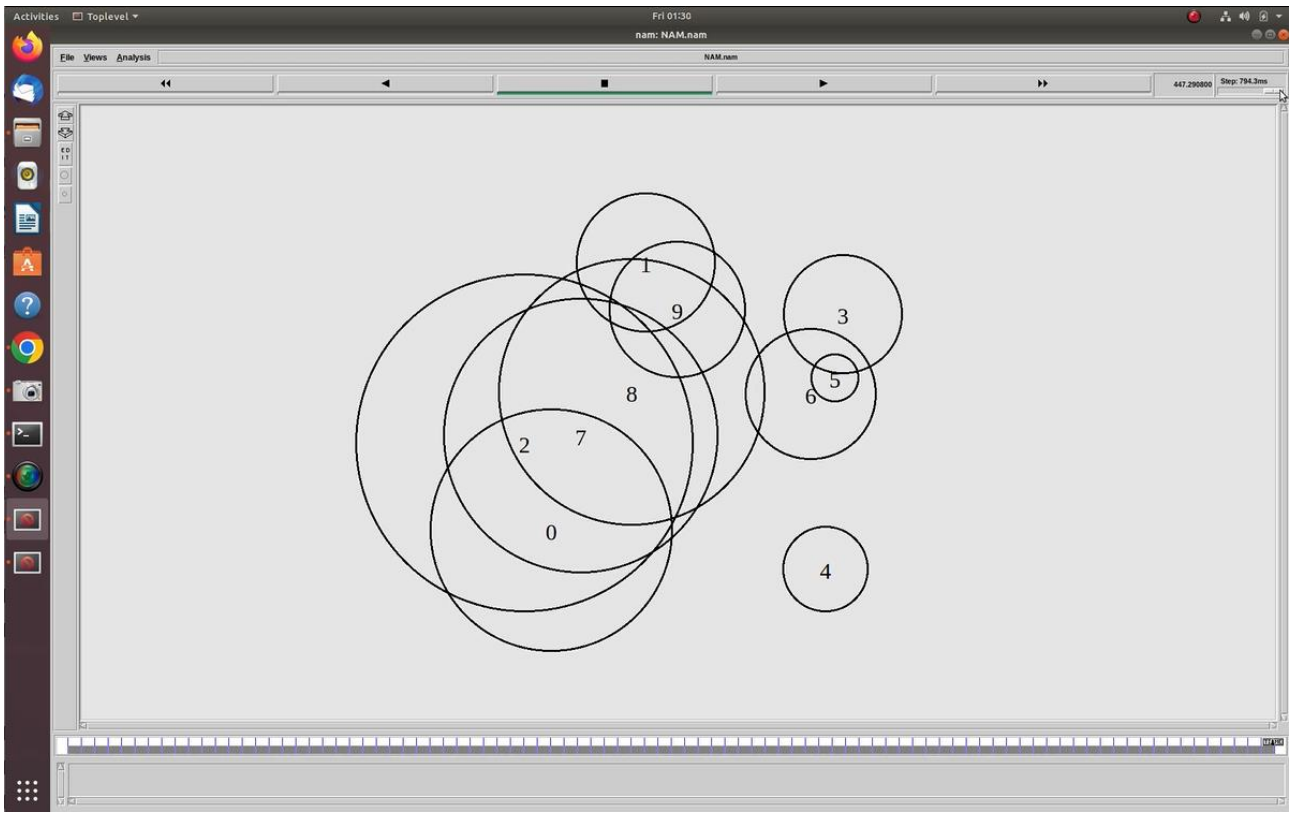

Figure 4-6 the end of the execution

note: you can see the video of execution by scanning this QR code (Google Drive):

4.6.Conclusion:

In this chapter, we simulate data transmission in the FANET network using the ns-2.35 simulation platform.

The drones move in three dimensions using the PATCH 3D and communicate with one another. The integration of these tools enabled us to simulate a FANET with drones (UAVs) sending and receiving data.

General conclusion

General conclusion

In conclusion, Flying Ad-Hoc Networks (FANETs) represent an innovative solution for enabling communication among Unmanned Aerial Vehicles (UAVs) without relying on a fixed infrastructure. FANETs offer a dynamic, self-organizing network architecture that facilitates real-time communication and coordination among drones and ground stations.

In our project, we simulated data transmission across FANET networks. To do this work, we went through several steps, starting with installing the ns-2.35 and PATCH 3D tools and following instructions on the Ubuntu 18.04 operating system.

Next, we created the UAVs' mobility and replicated data transmission over FANET. Finally, we generated the trace of the data packets sent and received throughout the simulation to determine whether there was a loss in the data transmission between the transmitter and the receiver.

With applications ranging from surveillance and monitoring to disaster response and delivery services, FANETs are poised to play a significant role in shaping the future of unmanned aerial operations. As technology continues to advance, FANETs are expected to become even more scalable, reliable, and efficient, opening up new possibilities for UAV-based services and applications.

Bibliography

Bibliographie

[1] M. Rouse, «Wireless Network,» Technology Expert, 16 june 2023. [En ligne]. Available: https://www.techopedia.com/definition/26186/wireless-network. [Accès le 28 05 2024].

[2] CDW Expert, «Wireless Networks: What are the Different Types and Which is Right for You?,» CDW, 06 september 2022. [En ligne]. Available: https://www.cdw.com/content/cdw/en/articles/datacenter/what-are-the-different-types-of-wireless-networks.html. [Accès le 28 05 2024].

[3] R. BABELHADJ et R. SEDRATI, *Routage dans les réseaux FANET, Etude comparative,* ouargla, Département d'informatique et technologie de l'information, 2022, pp. 4-6.

[4] D. M. MONDONGA, *Etude sur les protocoles de routage d'un réseau sans fil en mode Ad Hoc et leurs impacts. "cas de protocoles OLSR et AODV",* INSTITUT SUPERIEUR DE L'INFORMATIQUE, PROGRAMMATION ET ANALYSE, 2009.

[5] R. Agrawal, N. Faujdar, C. A. Tavera Romero, O. Sharma, G. M. Abdulsahib, O. I. Khalaf, R. . F. Mansoor et O. Ghoneim, «Classification and comparison of ad hoc networks: A review,» *ScienceDirect,* vol. 1, n° %124, pp. 7-10, 2023.

[6] İ. Bekmezci, O. K. Sahingoz et Ş. Temel, «Flying Ad-Hoc Networks (FANETs): A survey,» *ScienceDirect,* vol. 11, n° %13, pp. 1254-1270, 2013.

[7] O. S. Oubbati, A. Lakas, F. Zhou, M. Gunes et M. B. Yagoubi, «A Survey on Position-based Routing Protocols for Flying Ad hoc Networks,» *researchgate,* n° %110, pp. 6-10, 2017.

[8] S. Zaidi, M. Atiquzzaman et C. T Calafate, «Internet of Flying Things (IoFT): A Survey,» *HAL,* pp. 53-74, 2021.

[9] S. ZAIDI, S. BITAM et A. MELLOUK, «Enhanced adaptive sub-packet forward error correction mechanism,» chez *2016 IEEE Global Communications Conference (GLOBECOM)*, 2016.

[10] L. KHAMER, N. LABRAOUI et A. M. GUEROUI, «Road network layout based multi-hop broadcast protocols for Urban Vehicular Ad-hoc Networks,» *Wireless Networks,* vol. 27, pp. 1369-1388, 2021.

[11] S. ZAIDI, S. BITAM et A. MELLOUK, «Enhanced user datagram protocol for video streaming in VANET,» 2017.

[12] S. ZAIDI, S. BITAM et A. MELLOUK, «Hybrid error recovery protocol for video streaming in vehicle ad hoc,» *Vehicular communications,* vol. 12, pp. 110-126, 2018.

[13] Y. SAHRAOUI, A. GHANAM et S. ZAIDI, «Performance evaluation of TCP and UDP based video streaming in vehicular ad-hoc networks,» *2018 International Conference on Smart Communications in Network Technologies (SaCoNeT),* pp. 67-72, 2018.

[14] S. ZAIDI, M. OGAB et L. KHAMER, «Evaluation and comparison study of video streaming routing protocols in vehicular ad-hoc networks,» chez *Conference Proceedings ICCSA'2021*, 2020.

[15] F. Pasandideh, J. P. J. da Costa, R. Kunst, N. Islam, W. Hardjawana et E. P. de Freitas, «A Review of Flying Ad Hoc Networks: Key Characteristics, Applications, and Wireless Technologies,» *MDPI,* vol. 14, n° %14459, pp. 7-10, 2022.

[16] M. Asghar Khan et A. Safi, «Flying Ad-Hoc Networks (FANETs): A Review of architectures, and Routing protocols,» pakistan.

[17] S. Kr Maakar, Y. Singh et R. Singh, «An Enhanced Gauss-Markov Mobility Model for Simulation of FANET in 3-D Environment,» *Reaserch Gate,* vol. 10, 2018.

[18] M. B. Yassein et N. A. Mohamad Ali Damer, «Flying Ad-Hoc Networks: Routing Protocols, Mobility Models, Issues,» *researchgate,* vol. 7, n° %16, pp. 162-168, 2016.

[19] vemu, «STUDY OF NETWORK SIMULATOR(NS2),» [En ligne]. Available: https://vemu.org/uploads/lecture_notes/09_01_2024_933395485.pdf. [Accès le 21 05 2024].

[20] unknow, «omnetpp,» [En ligne]. Available: https://omnetpp.org/intro/. [Accès le 21 05 2024].

[21] unknow, «OPNET NETWORK SIMULATOR,» [En ligne]. Available: https://opnetprojects.com/opnet-network-simulator/. [Accès le 21 05 2024].

[22] unknow, «Simulation and Model-Based Design,» [En ligne]. Available: https://www.mathworks.com/help/simulink/index.html?s_tid=hc_panel. [Accès le 21 05 2024].

[23] N. Mahmood, J. DeDourek et P. Pochec, «M2ANET Simulation in 3D in NS2,» chez *SIMUL 2014 : The Sixth International Conference on Advances in System Simulation*, nice, 2014.